Lecture Notes in Computer Science 2276
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

Third International Conference, CICLing 2002
Mexico City, Mexico, February 17-23, 2002
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Alexander Gelbukh
CIC Centro de Investigacion en Computacion
IPN Instituto Politecnico Nacional
Col Zacateno, CP 07738, Mexico DF, Mexico
E-mail: gelbukh@cic.ipn.mx

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Computational linguistics and intelligent text processing : third international conference ; proceedings / CICLing 2002, Mexico City, Mexico, February 17 - 23, 2002. Alexander Gelbukh (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2276)
ISBN 3-540-43219-1

CR Subject Classification (1998): I.2.7, I.7, F.4.3, I.2, H.3

ISSN 0302-9743
ISBN 3-540-43219-1 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN 10846212 06/3142 5 4 3 2 1 0

Preface

CICLing 2002 was the third annual Conference on Intelligent text processing and Computational LINGuistics (hence the name CICLing); see www.CICLing.org. It was intended to provide a balanced view of the cutting edge developments in both theoretical foundations of computational linguistics and practice of natural language text processing with its numerous applications. A feature of CICLing conferences is their wide scope that covers nearly all areas of computational linguistics and all aspects of natural language processing applications. The conference is a forum for dialogue between the specialists working in these two areas.

This year we were honored by the presence of our invited speakers *Nicoletta Calzolari* (Inst. for Computational Linguistics, Italy), *Ruslan Mitkov* (U. of Wolverhampton, UK), *Ivan Sag* (Stanford U., USA), *Yorick Wilks* (U. of Sheffield), and *Antonio Zampolli* (Inst. for Computational Linguistics, Italy). They delivered excellent extended lectures and organized vivid discussions.

Of 67 submissions received, after careful reviewing 48 were selected for presentation; of them, 35 as full papers and 13 as short papers; by 98 authors from 19 countries: Spain (18 authors), Mexico (13), Japan, UK (8 each), Israel (7), Germany, Italy, USA (6 each), Switzerland (5), Taiwan (4), Ireland (3), Australia, China, Czech Rep., France, Russia (2 each), Bulgaria, Poland, Romania (1 each).

In addition to high scientific level, one of the success factors of the CICLing conferences is their excellent cultural program. CICLing 2002 was held in Mexico, a wonderful country very rich in culture, history, and nature. The participants of the conference – in their souls active researchers of the world – had a chance to see the solemn 2000-years-old pyramids of legendary Teotihuacanas, a Monarch butterfly wintering site where the old pines are covered with millions of butterflies as if they were leaves, a great cave with 85-meter halls and a river flowing from it, Aztec warriors dancing in the street in their colorful plumages, and the largest anthropological museum in the world; see photos at www.CICLing.org.

A conference is the result of the work of many people. First of all I would like to thank the members of the Program Committee for the time and effort they devoted to the reviewing of the submitted articles and to the selection process. Especially helpful were Ruslan Mitkov, Ted Pedersen, Grigori Sidorov, and many others – a complete list would be too long.

Obviously I thank the authors for their patience in the preparation of the papers, not to mention the very development of their scientific results that form this book. I also express my most cordial thanks to the members of the local Organizing Committee for their considerable contribution to making this conference a reality. Last but not least, I thank our sponsoring organization – the Center for Computing Research (CIC, www.cic.ipn.mx) of the National Polytechnic Institute (IPN), Mexico, for hosting the conference for the third time.

December 2001 Alexander Gelbukh

Organization

Program Committee

Barbu, Cătălina (U. Wolverhampton, UK),
Blekhman, Michael (Lingvistica'98 Inc., Canada),
Boitet, Christian (CLIPS-IMAG, France),
Bolshakov, Igor (CIC-IPN, Mexico),
Bontcheva, Kalina (U. Sheffield, UK),
Brusilovsky, Peter (U. Pittsburgh, USA),
Calzolari, Nicoletta (ILC-CNR, Italy),
Carroll, John (U. Sussex, UK),
Cassidy, Patrick (MICRA Inc., USA),
Cristea, Dan (U. Iasi, Romania),
Gelbukh, Alexander (**Chair**, CIC-IPN, Mexico)
Hasida, Kôiti (Electrotechnical Laboratory – AIST, Japan),
Harada, Yasunari (Waseda U., Japan),
Hirst, Graeme (U. Toronto, Canada),
Johnson, Frances (Manchester Metropolitan U., UK),
Kharrat, Alma (Microsoft Research, USA),
Kittredge, Richard (CoGenTex Inc., USA / Canada),
Knudsen, Line (U. Copenhagen, Denmark)
Koch, Gregers (U. Copenhagen, Denmark),
Kübler, Sandra (U. Tübingen, Germany),
Lappin, Shalom (King's College, UK),
Laufer, Natalia (Russian Institute of Artificial Intelligence, Russia),
López López, Aurelio (INAOE, Mexico),
Loukanova, Roussanka (Indiana U., USA / Bulgaria),
Lüdeling, Anke (U. Stuttgart, Germany),
Maegard, Bente (Centre for Language Technology, Denmark),
Martín-Vide, Carlos (U. Rovira i Virgili, Spain),
Mel'čuk, Igor (U. Montreal, Canada),
Metais, Elisabeth (U. Versailles, France),
Mikheev, Andrei (U. Edinburgh, UK),
Mitkov, Ruslan (U. Wolverhampton, UK),
Murata, Masaki (KARC-CRL, Japan),
Narin'yani, Alexander (Russian Institute of Artificial Intelligence, Russia),
Nevzorova, Olga (Kazan State U., Russia),
Nirenburg, Sergei (New Mexico U., USA),
Palomar, Manuel (U. Alicante, USA / Spain),
Pedersen, Ted (U. Minnesota Duluth, USA),
Pineda Cortes, Luis Alberto (UNAM, Mexico),
Piperidis, Stelios (Institute for Language and Speech Processing, Greece),
Ren, Fuji (U. Tokushima, Japan),
Sag, Ivan (Standford U., USA),
Sharoff, Serge (Russian Institute of Artificial Intelligence, Russia),

Sidorov, Grigori (CIC-IPN, Mexico),
Sun Maosong (Tsinghua U., China),
Tait, John (U. Sunderland, UK),
Trujillo, Arturo (Vocalis plc, UK),
T'sou Ka-yin, Benjamin (City U. Hong Kong, Hong Kong),
Van Guilder, Linda (MITRE Corp., USA),
Verspoor, Karin (Intelligenesis Corp., USA / The Netherlands),
Vilares Ferro, Manuel (U. La Coruña, Spain),
Wilks, Yorick (U. Sheffield, UK).

Additional Reviewers

Aljohar, Badr (King Faisal U., Saudi Arabia),
Alonso Ramos, Margarita (U. La Coruña, Spain),
Evans, Richard (U. Wolverhampton, UK),
Ferrández Rodríguez, Antonio (U. Alicante, Spain),
Kahane, Sylvain (Lattice, U. Paris 7, France),
Le An Ha (U. Wolverhampton, UK),
Martínez-Barco, Patricio (U. Alicante, Spain),
McCarthy, Diana (U. Sussex, UK),
Muñoz, Rafael (U. Alicante, Spain),
Orasan, Constantin (U. Wolverhampton, UK),
Sailer, Manfred (U. Tübingen, Germany),
Saiz Noeda, Maximiliano (U. Alicante, Spain),
Tsuge, Satoru (U. Tokushima, Japan).

Organizing Committee

Gelbukh, Alexander (**Chair**),
Salcedo Camarena, Teresa,
Ulloa Castillejos, Carlos,
Vargas Garcia, Soila,
Vizcaíno Sahagún, Carlos.

Organization, Website, and Contact

The conference was organized by the Natural Language Laboratory (www.cic.ipn.mx/Investigacion/ltexto.html) of the Center for Computing Research (CIC, Centro de Investigación en Computación, www.cic.ipn.mx) of the National Polytechnic Institute (IPN, Instituto Politécnico Nacional, www.ipn.mx), Mexico City, Mexico.

The website of the CICLing conferences is www.CICLing.org (mirrored at www.cic.ipn.mx/cicling). Contact: gelbukh@CICLing.org; also gelbukh@cic.ipn.mx, gelbukh@earthling.net, or gelbukh@dr.com; see also www.gelbukh.com.

Table of Contents

Computational Linguistics

Semantics

Invited Talk:

Word Sense Disambiguation

Invited Talk:

Anaphora

Invited Talk:

Syntax and Parsing

Part of Speech Tagging

Lexicon and Corpus

Text Generation

Morphology

Speech

Intelligent Text Processing

Spelling

Information Extraction and Information Retrieval

Summarization

Text Mining

Text Classification and Categorization

Document Processing

Demo Descriptions

Multiword Expressions: A Pain in the Neck for NLP*

Ivan A. Sag[1], Timothy Baldwin[1], Francis Bond[2],
Ann Copestake[3], and Dan Flickinger[1]

[1] CSLI, Ventura Hall, Stanford University
Stanford, CA 94305-4115, USA
{sag,tbaldwin,danf}@csli.stanford.edu
[2] NTT Communication Science Labs., 2-4 Hikaridai
Seika-cho, Soraku-gun, Kyoto, Japan 619-0237
bond@cslab.kecl.ntt.co.jp
[3] University of Cambridge, Computer Laboratory, William Gates Building
JJ Thomson Avenue, Cambridge CB3 OFD, UK
Ann.Copestake@cl.cam.ac.uk

Abstract. Multiword expressions are a key problem for the development of large-scale, linguistically sound natural language processing technology. This paper surveys the problem and some currently available analytic techniques. The various kinds of multiword expressions should be analyzed in distinct ways, including listing "words with spaces", hierarchically organized lexicons, restricted combinatoric rules, lexical selection, "idiomatic constructions" and simple statistical affinity. An adequate comprehensive analysis of multiword expressions must employ both symbolic and statistical techniques.

1 Introduction

The tension between symbolic and statistical methods has been apparent in natural language processing (NLP) for some time. Though some believe that the statistical methods have rendered linguistic analysis unnecessary, this is in fact not the case. Modern statistical NLP is crying out for better language models (Charniak 2001). At the same time, while 'deep' (linguistically precise) processing has now crossed the industrial threshold (Oepen et al. 2000) and serves as the basis for ongoing product development in a number of application areas (e.g. email autoresponse), it is widely recognized that deep analysis must come

* The research reported here was conducted in part under the auspices of the LinGO project, an international collaboration centered around the LKB system and related resources (see http://lingo.stanford.edu). This research was supported in part by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University. We would like to thank Emily Bender and Tom Wasow for their contributions to our thinking. However, we alone are responsible for any errors that remain.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 1–15, 2002.

to grips with two key problems, if linguistically precise NLP is to become a reality.

The first of these is **disambiguation**. Paradoxically, linguistic precision is inversely correlated with degree of sentence ambiguity. This is a fact of life encountered by every serious grammar development project. Knowledge representation, once thought to hold the key to the problem of disambiguation, has largely failed to provide completely satisfactory solutions. Most research communities we are aware of that are currently developing large scale, linguistically precise, computational grammars are now exploring the integration of stochastic methods for ambiguity resolution. The second key problem facing the deep processing program – the problem of **multiword expressions** – is underappreciated in the field at large. There is insufficient ongoing work investigating the nature of this problem or seeking computationally tractable techniques that will contribute to its solution.

We define multiword expressions (MWEs) very roughly as "idiosyncratic interpretations that cross word boundaries (or spaces)". As Jackendoff (1997: 156) notes, the magnitude of this problem is far greater than has traditionally been realized within linguistics. He estimates that the number of MWEs in a speaker's lexicon is of the same order of magnitude as the number of single words. In fact, it seems likely that this is an underestimate, even if we only include lexicalized phrases. In WordNet 1.7 (Fellbaum 1999), for example, 41% of the entries are multiword. For a wide coverage NLP system, this is almost certainly an underestimate. Specialized domain vocabulary, such as terminology, overwhelmingly consists of MWEs, and a system may have to handle arbitrarily many such domains. As each new domain adds more MWEs than simplex words, the proportion of MWEs will rise as the system adds vocabulary for new domains.

MWEs appear in all text genres and pose significant problems for every kind of NLP. If MWEs are treated by general, compositional methods of linguistic analysis, there is first an **overgeneration problem**. For example, a generation system that is uninformed about both the patterns of compounding and the particular collocational frequency of the relevant dialect would correctly generate *telephone booth* (American) or *telephone box* (British/Australian), but might also generate such perfectly compositional, but unacceptable examples as *telephone cabinet*, *telephone closet*, etc. A second problem for this approach is what we will call the **idiomaticity problem**: how to predict, for example, that an expression like *kick the bucket*, which appears to conform to the grammar of English VPs, has a meaning unrelated to the meanings of *kick*, *the*, and *bucket*. Syntactically-idiomatic MWEs can also lead to parsing problems, due to nonconformance with patterns of word combination as predicted by the grammar (e.g. the determinerless *in line*).

Many have treated MWEs simply as **words-with-spaces**, an approach with serious limitations of its own. First, this approach suffers from a **flexibility problem**. For example, a parser that lacks sufficient knowledge of verb-particle constructions might correctly assign *look up the tower* two interpretations ("glance up at the tower" vs. "consult a reference book about the tower"), but fail to treat the subtly different *look the tower up* as unambiguous ("consult a reference book ..." interpretation only). As we will show, MWEs vary considerably with respect to this and other kinds of flexibility. Finally, this simple approach to MWEs suffers from a **lexical proliferation problem**. For example, light verb constructions often come in families, e.g. *take a walk*, *take a hike*, *take a trip*, *take a flight*. Listing each such expression results in considerable loss of generality and lack of prediction. Many current approaches are able to get commonly-attested MWE usages right, but they use ad hoc methods to do so, e.g. preprocessing of various kinds and stipulated, inflexible correspondences. As a result, they handle variation badly, fail to generalize, and result in systems that are quite difficult to maintain and extend.

Though the theory of MWEs is underdeveloped and the importance of the problem is underappreciated in the field at large, there is ongoing work on MWEs within various projects that are developing large-scale, linguistically precise computational grammars, including the ParGram Project at Xerox PARC (`http://www.parc.xerox.com/istl/groups/nltt/pargram/`), the XTAG Project at the University of Pennsylvania (`http://www.cis.upenn.edu/~xtag/`), work on Combinatory Categorial Grammar at Edinburgh University, and the LinGO Project (a multi-site collaboration including CSLI's English Resource Grammar Project — `http://lingo.stanford.edu`), as well as by the FrameNet Project (`http://www.icsi.berkeley.edu/~framenet/`), which is primarily developing large-scale lexical resources. All of these projects are currently engaged (to varying degrees) in linguistically informed investigations of MWEs[1].

We believe the problem of MWEs is critical for NLP, but there is a need for better understanding of the diverse kinds of MWE and the techniques now readily available to deal with them. In Section 2, we provide a general outline of some common types of MWE in English and their properties. In Section 3, we survey a few available analytic techniques and comment on their utility, drawing from our own research using HPSG-style grammars and the LKB system. In the conclusion, we reflect on prospects for the future of MWE research.

2 Some Kinds of MWE

MWEs can be broadly classified into **lexicalized phrases** and **institutionalized phrases** (terminology adapted from Bauer (1983)). Lexicalized phrases have at least partially idiosyncratic syntax or semantics, or contain 'words' which do not occur in isolation; they can be further broken down into **fixed expressions**, **semi-fixed expressions** and **syntactically-flexible expressions**, in roughly decreasing order of lexical rigidity. Institutionalized phrases are syntactically and semantically compositional, but occur with markedly high frequency (in a given context). Below, we examine instances of each category and discuss some of the peculiarities that pose problems for both words-with-spaces and fully compositional analyses.

[1] We thank Chuck Fillmore, Aravind Joshi, Ron Kaplan, and Mark Steedman for discussions of this point.

2.1 Fixed Expressions

There is a large class of immutable expressions in English that defy conventions of grammar and compositional interpretation. This class includes *by and large*, *in short*, *kingdom come*, and *every which way*. Many other MWEs, though perhaps analyzable to scholars of the languages whence they were borrowed, belong in this class as well, at least for the majority of speakers: *ad hoc* (cf. *ad nauseum*, *ad libitum*, *ad hominem*,...), *Palo Alto* (cf. *Los Altos*, *Alta Vista*,...), etc.

Fixed expressions are fully lexicalized and undergo neither morphosyntactic variation (cf. **in shorter*) nor internal modification (cf. **in very short*)). As such, a simple words-with-spaces representation is sufficient. If we were to adopt a compositional account of fixed expressions, we would have to introduce a lexical entry for "words" such as *hoc*, resulting in overgeneration and the idiomaticity problem (see above).

2.2 Semi-fixed Expressions

Semi-fixed expressions adhere to strict constraints on word order and composition, but undergo some degree of lexical variation, e.g. in the form of inflection, variation in reflexive form, and determiner selection. This makes it possible to treat them as a word complex with a single part of speech, which is lexically variable at particular positions. They can take a range of forms including non-decomposable idioms, and certain compound nominals and proper names. Below, we discuss some problematic instances of each, for which neither a fully compositional account nor simple string-type listing in a lexicon is appropriate.

Non-decomposable Idioms. Nunberg et al. (1994) introduced the notion of 'semantic compositionality' in relation to idioms, as a means of describing how the overall sense of a given idiom is related to its parts. Idioms such as *spill the beans*, for example, can be analyzed as being made up of *spill* in a "reveal" sense and *the beans* in a "secret(s)" sense, resulting in the overall compositional reading of "reveal the secret(s)". With the oft-cited *kick the bucket*, on the other hand, no such analysis is possible.

Based on the observation that this process of semantic deconstruction starts off with the idiom and associates particular components of the overall meaning with its parts, it has been recast as **semantic decomposability**. We distinguish between **decomposable idioms** such as *spill the beans* and *let the cat out of the bag*, and **non-decomposable idioms** such as *kick the bucket*, *trip the light fantastic* and *shoot the breeze*. We return to discuss decomposable idioms in Section 2.3.

Due to their opaque semantics, non-decomposable idioms are not subject to syntactic variability, e.g. in the form of internal modification (#*kick the great bucket in the sky*[2]) or passivization (**the breeze was shot*). The only types of

[2] We make the claim that *proverbial* as in *kick the proverbial bucket* is a metalinguistic marker, and thus does not qualify as an internal modifier.

lexical variation observable in non-decomposable idioms are inflection (*kicked the bucket*) and variation in reflexive form (*wet oneself*).

Adopting a words-with-spaces description of non-decomposable idioms is unable to capture the effects of inflectional variation and variation in reflexive form, except at the risk of lexical proliferation in describing all possible lexical variants of each idiom (with well over 20 lexical entries for *wet/wets/wetted/wetting myself/yourself/herself/himself/themselves/oneself/itself*). On the other hand, a fully compositional account may have no trouble with lexical variation, but it has troubles with idiomaticity (e.g. deriving the "die" semantics from *kick*, *the*, and *bucket*) and overgeneration (e.g. in generating **the breeze was shot*).

Compound Nominals. Compound nominals such as *car park*, *attorney general* and *part of speech* are similar to non-decomposable idioms in that they are syntactically-unalterable units that inflect for number. For many right-headed compound nominals, a words-with-spaces handling can generally cope with number inflection by way of the simplex word mechanism of simply adding an *-s* to the end of the string, as in [*car park*]*s*. For left-headed compounds such as *attorney general*, *congressman at large* and *part of speech*, on the other hand, this would result in anomalies such as *[*congressman at large*]*s*. Admittedly, the lexical proliferation associated with listing the singular and plural forms of each compound nominal is less dramatic than with non-decomposable idioms, but still leaves a lot to be desired in terms of systematicity.

As for non-decomposable idioms, fully compositional approaches suffer from the idiomaticity and overgeneration problems.

Proper Names. Proper names are syntactically highly idiosyncratic. U.S. sports team names, for example, are canonically made up of a place or organization name (possibly a MWE in itself, such as *San Francisco*) and an appellation that locates the team uniquely within the sport (such as *49ers*). The first obstacle for a words-with-spaces representation for U.S. team names is that the place/organization name is optionally ellidable (e.g. *the* (*San Francisco*) *49ers*), a generalization which cannot be captured by a single string-based lexical entry.

Additionally, U.S. sports team names take a definite reading. This results in the determiner *the* being selected by default when the team name occurs as an NP, as in *the* (*San Francisco*) *49ers* and *the* (*Oakland*) *Raiders*. When the team name occurs as a modifier in a compound noun (as in *an/the* [[(*Oakland*) *Raiders*] *player*]), however, the determiner is associated with the compound noun, and the team name becomes determinerless. Coordination also produces interesting effects, as it is possible to have a single determiner for a coordinated team name complex, as in *the* [*Raiders and 49ers*].

Lexical proliferation once again becomes a problem with a words-with-spaces approach to U.S. sports team names. We would need to generate lexicalizations incorporating the determiners *the* or *those*, as well as alternative lexicalizations with no determiner. And all of these would have to allow the place/organization name to be optional (e.g. *the San Francisco 49ers*, *those San Francisco 49ers*, *San*

Francisco 49ers, *the 49ers*, *those 49ers* and *49ers*). In addition, the words-with-spaces approach seems inconsistent with the internal modifiers we find in such examples as *the league-leading* (*San Francisco*) *49ers.* Full compositionality, on the other hand, runs up against gross overgeneration, as any place/organization name is allowed to combine with any appellation, yielding such non-denoting names as *the Oakland 49ers.*

2.3 Syntactically-Flexible Expressions

Whereas semi-fixed expressions retain the same basic word order throughout, syntactically-flexible expressions exhibit a much wider range of syntactic variability. We illustrate the types of variation possible in the form of verb-particle constructions, decomposable idioms and light verbs.

Verb-Particle Constructions. Verb-particle constructions consist of a verb and one or more particles, such as *write up*, *look up* and *brush up on.* They can be either semantically idiosyncratic, such as *brush up on*, or compositional such as *break up* in *the meteorite broke up in the earth's atmosphere* (Bolinger 1972, Dixon 1982, Dehé et al. to appear)[3]. In compositional usages, the particle(s) act as a construction and modify the spatial, aspectual, etc properties of the head verb, such as *up* transforming *eat* from an activity into an accomplishment in *eat up.* That is, the particle(s) generally assume semantics idiosyncratic to verb-particle constructions, but are semi-productive (cf. *gobble up* in the case of *up*).

Transitive verb-particle constructions take an NP argument either between or following the verb and particle(s) (e.g. *call Kim up* and *fall off a truck*, respectively). Certain transitive verb-particle constructions are compatible with only particle-initial realizations (consider **fall a truck off*), while others are compatible with both forms (e.g. *call Kim up* vs. *call up Kim*). Even with intransitive verb-particle constructions, adverbs can often be inserted between the verb and particle (e.g. *fight bravely on*). As a result, it is impossible to capture the full range of lexical variants of transitive verb-particle constructions as words-with-spaces.

As with other MWE types, a fully compositional approach is troubled by the idiomaticity and overgeneration problems. Even for seemingly synonymous verbs combining compositionally with the same particle, idiosyncrasies are observed (e.g. *call/ring/phone/telephone* vs. *call/ring/phone/*telephone up*: McIntyre 2001) which would be beyond the descriptive powers of a purely compositional account.

Decomposable Idioms. Decomposable idioms, such as *let the cat out of the bag* and *sweep under the rug*, tend to be syntactically flexible to some degree.

[3] The combination *break up* also has semantically idiosyncratic senses including "adjourn" and "separate".

Exactly which types of syntactic variation a given idiom can undergo, however, is highly unpredictable (Riehemann 2001).

Because decomposable idioms are syntactically variable to varying degrees, it is hard to account for them using only syntactic selection. Instead, they act like they are composed of semantically linked parts, which thus suggests a semantic approach is appropriate (Nunberg et al. 1994). Because they are highly variable syntactically, decomposable idioms are incompatible with a words-with-spaces strategy; fully compositional techniques suffer from the idiomaticity problem.

Light Verbs. Light-verb constructions (e.g. *make a mistake*, *give a demo*, **do a mistake*, **make a demo*) are highly idiosyncratic – it is notoriously difficult to predict which light verb combines with a given noun (Abeillé 1988). Although such phrases are sometimes claimed to be idioms, this seems to be stretching the term too far: the noun is used in a normal sense, and the verb meaning appears to be bleached, rather than idiomatic.

Light-verb constructions are subject to full syntactic variability, including passivization (e.g. *a demo was given*), extraction (e.g. *How many demos did Kim give?*) and internal modification (e.g. *give a revealing demo*). They thus cannot be treated as words-with-spaces. A fully compositional account, on the other hand, would be unable to model the blocking of alternative light verb formations (e.g. *give a demo* vs. **make a demo*), and thus would suffer from gross overgeneration.

2.4 Institutionalized Phrases

Institutionalized phrases are semantically and syntactically compositional, but statistically idiosyncratic. Consider for example *traffic light*, in which both *traffic* and *light* retain simplex senses and combine constructionally to produce a compositional reading. Given this strict compositionality, we would expect the same basic concept to be expressible in other ways, e.g. as *traffic director* or *intersection regulator*. Clearly, however, no such alternate form exists, because the form *traffic light* has been conventionalized. The idiosyncrasy of *traffic light* is thus statistical rather than linguistic, in that it is observed with much higher relative frequency than any alternative lexicalization of the same concept. Other examples of institutionalized phrases are *telephone booth* (or *telephone box* in British/Australian English), *fresh air* and *kindle excitement*. We refer to potential lexical variants of a given institutionalized phrase which are observed with zero or markedly low frequency as **anti-collocations** (Pearce 2001).

One subtle effect observed with institutionalized phrases is that association with the concept denoted by that expression can become so strong as to diminish decomposability. *Traffic light*, for example, could conceivably be interpreted as a device for communicating intended actions to surrounding traffic. However, partly as a result of the existence of an institutionalized term for such a device (i.e. *turn*(*ing*) *signals*) and partly due to the conventionalization of *traffic light* to denote a stoplight, this reading is not readily available.

Note that we reserve the term **collocation** to refer to any statistically significant cooccurrence, including all forms of MWE as described above and compositional phrases which are predictably frequent (because of real world events or other nonlinguistic factors). For instance, *sell* and *house* cooccur in sentences more often than would be predicted on the basis of the frequency of the individual words, but there is no reason to think that this is due to anything other than real world facts.

As institutionalized phrases are fully compositional, they undergo full syntactic variability. Words-with-spaces approaches thus suffer from lexical proliferation, while fully compositional approaches encounter the idiomaticity and overgeneration problems.

3 Some Analytic Techniques

In this section we will introduce some analyses for MWEs using the constraint-based Head-driven Phrase Structure Grammar (HPSG) formalism (Pollard and Sag 1994, Sag and Wasow 1999). Most of these analyses have been implemented in grammars in the LKB grammar development environment (Copestake in press). Ultimately, we plan to include them all in the English Resource Grammar; at present some are being tested in smaller grammars.

The LKB grammar development environment is a general system for developing typed feature structure grammars which implements a particular typed feature structure logic. It is written in Common Lisp and currently runs under Linux, Solaris, Windows and MacOS. Grammar development is effectively a process of programming in a very high-level specialized language, and the system supports interactive grammar development as well as parsing and generation.

The LinGO English Resource Grammar (ERG) is a broad-coverage grammar of English described in a typed feature structure logic compatible with the LKB and several other systems. The grammar itself is written in HPSG, while the semantic representation used is Minimal Recursion Semantics (MRS hereafter – Copestake et al. 1999). An overview of the ERG (from a computational linguistic perspective) is given in Copestake and Flickinger (2000).

3.1 Analyzing Fixed Expressions

Truly fixed expressions, like *ad hoc* or *of course*, can simply be dealt with as words-with-spaces. In this case a list of words is given the same lexical type as a single word and associated with a single semantic relation. For example, in the current ERG, *ad hoc* is defined as having the type `intrans_adj_l` (intransitive adjective listeme[4], which is also the type for simplex adjectives such as *pretty*). However, simply listing MWEs as strings, as in (1), is adequate only for expressions which allow no variability at all. The expression can be externally modified: *very ad hoc*, but not internally modified: **ad very hoc*[5].

[4] A listeme is a lexically-listed entity.

[5] This and subsequent feature structures are intended for illustrative purposes and are not as they appear in the ERG.

```
(1) ad_hoc_1 := intr_adj_l &
      [ STEM < "ad", "hoc" >,
        SEMANTICS [KEY ad-hoc_rel ]].
```

In practice, there is often an unfortunate side effect to allowing these expressions in an implementation: developers exploit this class to add entries that can vary, but don't often, in order to quickly achieve greater coverage.

3.2 Analyzing Semi-fixed Expressions

When analyzing semi-fixed expressions, it is important to strike a balance between too weak a mechanism, which will not allow sufficient variability, and too strong a mechanism, which will allow too much. We make heavy use of existing features of our grammars, in particular multiple inheritance. We also introduce two new mechanisms: the ability to specify which words inflect in an otherwise fixed expression and the ability to treat a list of listemes as a single listeme.

Internal Inflection. Some semi-fixed MWEs, such as *kick the bucket*, *part of speech* and *pain in the neck* differ from fixed expressions in that one word in them inflects, as though it were the phrasal head. In this case, it is still possible to treat the whole entry (a list of words) as a single listeme that is associated with a single semantic relation. We add a pointer showing which word to inflect (INFL-POS = inflection position, i.e. inflect the *n*th word in the STEM list). An entry for *part of speech*, where only the first word *part* inflects, is given in (2).

```
(2) part_of_speech_1 := intr_noun_l &
      [ STEM < "part", "of", "speech" >,
        INFL-POS "1",
        SEMANTICS [KEY part_of_speech_rel ]].
```

The analysis can be extended to words with two inflecting parts, such as *wine and dine*, which we would like to treat as a single transitive verb, but with both *wine* and *dine* inflecting: *Kim wined and dined Sandy.*

In a deeper treatment of these expressions the list of words would be replaced with a list of listemes (LEX-SIGNS), so that the words can inherit their properties from existing listemes. In this case, the expression as a whole would, by default, inherit its lexical type from the designated inflecting word: thus *part of speech* would inherit from *part* and would be a count noun, while *fool's gold* would inherit from *gold* and would be a mass noun. This inheritance allows us to capture the generalization that a *performance artist* is a kind of *artist* though the use of *performance* is non-compositional.

Hierarchical Lexicon with Default Constraint Inheritance. Default inheritance allows us to simplify the structure of the lexical types used. For example, by default, proper names in English take no determiner. In our analysis, we handle this by requiring the specifier (SPR) list to be empty, as in (3a). However, some names, such as those of U.S. sports teams, normally take a definite determiner. Therefore, the constraint on Name is defeasible: it can be overridden in rules that inherit from it. The logic for defaults we assume follows Lascarides and Copestake (1999), where default values are indicated by '/'.

The type USTeamName overrides the default, in this case, by specifying that the specifier must be a definite determiner, and that the number defaults to plural, as shown in (3b):

(3) a Name: [SPR / ⟨ ⟩]
b USTeamName: [SPR ⟨ Det[definite] ⟩, NUM / plural]

The specifier is not given as the listeme *the*, but just as the specification definite. In the absence of other information this would normally be the definite article[6], but other definite determiners are also possible: *How about those Raiders?*

The listeme for *the Oakland Raiders*, would thus be of the type USTeamName and described as a list of listemes, inherited from *Oakland* and *Raiders*. This analysis captures the fact that the first word is the same as the place *Oakland*. The structure is shown in (4), where oakland_1 and raiders_1 are listeme identifiers for the place *Oakland* and the appellation *Raiders*[7]:

```
(4) oakland_raiders_1 := USTeamName &
      [  LEX-SIGNS / < oakland_1, raiders_1 >,
         SEMANTICS < oakland_raiders_rel > ].
```

Note further that there are exceptions to the subregularity of sports team names. Certain teams have names that are combinations of determiner plus mass noun, such as *the* (*Miami*) *Heat*, *the* (*Philadelphia*) *Charge*, and *the* (*Stanford*) *Cardinal*[8]. Since mass nouns are singular, the appropriate constraint on the subtype MassTeamName overrides the defeasible [NUM / plural] specification in (3b).

The USTeamName type, as it is presented here, still does not capture (i) the optionality of *Oakland* and (ii) the fact that the first word in team names is typically a place or organization. Two analyses suggest themselves. In the first of these, the lexical type USTeamName licenses an optional second specifier, in addition to the determiner. This specifier would be the appropriate place name or organization. In the second possible analysis, an extremely circumscribed construction, inheriting from the noun-noun compound phrase rule, would license combinations headed by listemes of the type USTeamName with a modifier that must be a place or organization. It remains to be seen whether either of these proposals is viable.

[6] Obtainable by setting *the* to be the default definite determiner.

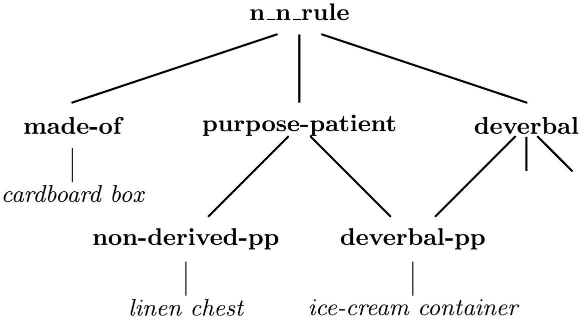
[7] Inheritance from identifiers diverges from standard HPSG practice, but see Copestake (1992) for formalization and motivation.

[8] This name refers to the color, not the bird.

3.3 Analyzing Syntactically-Flexible Expressions

Many of the syntactically-flexible MWEs can again be handled by existing mechanisms: the use of **circumscribed constructions** and lexical selection. We introduce a new mechanism to handle the most variable decomposable idioms, that allows us to check that all the idiomatic parts are there in the appropriate semantic relationships.

Circumscribed Constructions. Inheritance hierarchies of constructions for noun-noun compounds can be used to capture some of the semi-productivity of syntactically-flexible expressions (Copestake and Lascarides 1997). The idea is that compounds like *spring beginning* (cf. (*the*) *beginning of spring*) are not completely blocked, but they are prevented from having any conventional interpretation, and will be interpreted as incoherent unless licensed by a specific discourse context. The diagram below shows a fragment of the compound nominal construction hierarchy adapted from that paper, with example compounds corresponding to the various categories at each leaf node:



This hierarchy allows generalizations about productive and lexicalized forms to be represented: for productive forms, the construction is interpreted as a grammar rule, while lexicalized forms stipulate the construction as part of their entry. The use of defaults allows generalizations about stress, for instance, to be expressed.

Lexical Selection. Verb-particle constructions, conjunctions like *either... or...* and so on, where material intervenes between the elements of the phrase, can be accounted for by means of a lexical selection mechanism where a sign associated with one word of the phrase selects for the other word(s). For instance, in the existing ERG, there is an entry for *hand* which subcategorizes for *out*, as shown in (5):

```
(5) hand_out_v1 := mv_prep_particle_np_l &
      [ STEM < "hand" >,
        SEMANTICS [ KEY hand_out_rel,
                    --COMPKEY out_rel ] ].
```

The semantics of the whole expression is given in the `KEY` relation (`hand_out_rel`); the verb *hand* then selects for the preposition whose `KEY` relation is given by `COMPKEY` (`out_rel`). This allows:

(6) Kim handed out chocolate to the kids.

A lexical rule permutes the subcategorization list to allow:

(7) Kim handed the chocolate out to the kids.

Combinations with prepositions, such as *rely on*, *fond of* or *report on/about* can be handled in a similar manner, by selecting for the semantic relation encoded by the preposition. Early HPSG accounts of preposition selection used a `PFORM` (`PREPOSITION-FORM`) feature for this (Pollard and Sag 1994). The atomic values of `PFORM` simply encoded the phonetic form of the preposition. The ERG uses the basic semantic `KEY` relations. Either analysis allows prepositions to be grouped together into regularized types, which allows natural classes of prepositions to be selected.

Light Verbs. Light verbs, that is those verbs which cooccur with certain classes of nouns, can also be handled by selection. All nouns which can be used with a given light verb will have semantic types which inherit from the same type (for example `mistake_rel` inherits from `make_arg_rel`). The light verb *make* then has the selectional restriction that its direct object must be of the type `make_arg_rel`). Another light verb, such as *do*, does not select for `make_arg_rel`, and thus will not allow **do a mistake*. Nouns which can be used with more than one light verb multiply inherit from the relevant classes. The normal mechanisms of the grammar will allow for the selectional restrictions to be passed along through long distance dependencies such as in *the mistakes that he managed to make were incredible.*

Decomposable Idioms. Selection works if the syntactic relationship of the various parts of the phrase is fixed, as it indeed seems to be for verb particle constructions, but the mechanism runs into problems with some idioms, for instance, where the relationship between the words may be very flexible.

We start from the assumption that the relationship between words in decomposable idioms can be captured using a partially semantic mechanism, essentially following the approach described by Nunberg et al. (1994). The flat MRS representation adopted in the ERG is especially suited to this. Riehemann (2001) describes one approach that uses MRS; here we sketch another, which builds directly on ideas first presented in Copestake (1994).

Consider, for instance, the idiom *cat out of the bag* which can be described as a phrase containing the semantic relationships in (8), where `i_cat` and `i_bag` are the meanings corresponding to the idiomatic senses of *cat* “secret” and *bag* “hiding place”.

(8) $[\ \texttt{i\_cat}(x) \wedge \texttt{i\_bag}(y) \wedge \texttt{out}(x,y)\ ]$

This semantic representation is flexible enough to cover the most common forms of this idiom. The problem is that matching this specification to a conventional semantic representation is arbitrarily complex, because of the possible contributions of quantifiers and so on. In order to get this sort of idea to work, Pulman (1993) proposes an approach which relies on a form of quasi-inference operating on a compositionally derived logical form. However, his approach fails to allow for any syntactic idiosyncrasy among idioms.

Copestake (1994) develops a treatment of decomposable idioms that is semantically based, but which uses a notion of **idiomatic construction** to accommodate syntactic flexibility. Instead of locating interpretational idiosyncrasy in idiomatic listemes (e.g. *let*) that select for other such listemes (e.g. *the* and *cat*), this approach allows listemes to combine constructionally by ordinary syntactic means. However, idiomatic constructions provide an independent dimension of phrasal classification where idiomatic interpretations are assigned just in case the right pieces (e.g. *the*, *cat*, *out*, *of*, *the*, *bag*) are all present and in the right predicate-argument relations. Because the account is based on MRS, where the semantics is represented in terms of bags of predications, rather than representations with complex embeddings, it becomes natural to state a constraint requring that a given set of predications be present and appropriately related (e.g. the argument of *cat*'s predication must also be the first argument of the *out* predication). In this way, quantification and modification of pieces of idioms are allowed, as is reordering of idiomatic elements from their canonical position. This constructional approach thus differs from earlier lexical approaches, but retains the notion that there is a dependency among the lexical parts of decomposable idioms.

3.4 Information about Frequency

The treatment of frequency is different in type from the analyses described above. The grammatical rules constrain the space of possible sentences and interpretations, while frequency-based probabilities allow us to predict which of these is the preferred interpretation or string. In order to use probabilities in both analysis (from strings to meanings) and generation (from meanings to strings), we need frequency information about both semantic relations and construction rules, in so far as they contribute to semantic interpretation. The necessity of semantic frequency information has been somewhat neglected in current NLP research, no doubt largely because it is difficult to collect.

Johnson et al. (1999) describe a potentially viable approach to developing probabilistic grammars based on feature structures; Hektoen (1997) suggests an alternative model of semantic probabilities. Both of these are possible approaches to institutionalized phrases because of the fine granularity we assume for relations in MRS. For instance, `fine_rel` and `good_rel` are distinct, so the relative frequency of *fine weather* versus *good weather* could be considered in terms of their semantic relations.

The question of determining the preferred interpretation is sometimes regarded as outside the scope of a formal linguistic account, but we believe that frequency information should be regarded as part of a speaker's knowledge of language. In any case, its utility in natural language processing is beyond question.

4 Conclusion

In this paper we hope to have shown that MWEs, which we have classified in terms of lexicalized phrases (made up of fixed, semi-fixed and syntactically flexible expressions) and institutionalized phrases, are far more diverse and interesting than is standardly appreciated. Like the issue of disambiguation, MWEs constitute a key problem that must be resolved in order for linguistically precise NLP to succeed. Our goal here has been primarily to illustrate the diversity of the problem, but we have also examined known techniques — listing words with spaces, hierarchically organized lexicons, restricted combinatoric rules, lexical selection, idiomatic constructions, and simple statistical affinity. Although these techniques take us further than one might think, there is much descriptive and analytic work on MWEs that has yet to be done. Scaling grammars up to deal with MWEs will necessitate finding the right balance among the various analytic techniques. Of special importance will be finding the right balance between symbolic and statistical techniques.

References

Abeillé, Anne: 1988, 'Light verb constructions and extraction out of NP in a tree adjoining grammar', in *Papers of the 24th Regional Meeting of the Chicago Linguistics Society*.

Bauer, Laurie: 1983, *English Word-formation*, Cambridge: Cambridge University Press.

Bolinger, Dwight, ed.: 1972, *Degree Words*, the Hague: Mouton.

Charniak, Eugene: 2001, 'Immediate-head parsing for language models', in *Proc. of the 39th Annual Meeting of the ACL and 10th Conference of the EACL (ACL-EACL 2001)*, Toulouse.

Copestake, Ann: 1992, 'The representation of lexical semantic information', Ph.D. thesis, University of Sussex.

Copestake, Ann: 1994, 'Representing idioms', Presentation at the HPSG Conference, Copenhagen.

Copestake, Ann: in press, *Implementing Typed Feature Structure Grammars*, Stanford: CSLI Publications.

Copestake, Ann & Dan Flickinger: 2000, 'An open-source grammar development environment and broad-coverage English grammar using HPSG', in *Proc. of the Second conference on Language Resources and Evaluation (LREC-2000)*, Athens.

Copestake, Ann, Dan Flickinger, Ivan Sag & Carl Pollard: 1999, 'Minimal recursion semantics: An introduction', (`http://www-csli.stanford.edu/~aac/papers/newmrs.ps`), (draft).

Copestake, Ann & Alex Lascarides: 1997, 'Integrating symbolic and statistical representations: The lexicon pragmatics interface', in *Proc. of the 35th Annual Meeting of the ACL and 8th Conference of the EACL (ACL-EACL'97)*, Madrid, pp. 136–43.

Dehé, Nicole, Ray Jackendoff, Andrew McIntyre & Silke Urban, eds.: to appear, *Verb-particle explorations*, Mouton de Gruyter.

Dixon, Robert: 1982, 'The grammar of English phrasal verbs', *Australian Journal of Linguistics*, **2**: 149–247.

Fellbaum, Christine, ed.: 1998, *WordNet: An Electronic Lexical Database*, Cambridge, MA: MIT Press.

Hektoen, Eirik: 1997, 'Probabilistic parse selection based on semantic cooccurrences', in *Proc. of the 5th International Workshop on Parsing Technologies (IWPT-97)*, MIT, pp. 113–122.

Jackendoff, Ray: 1997, *The Architecture of the Language Faculty*, Cambridge, MA: MIT Press.

Johnson, Mark, Stuart Geman, Stephan Canon, Zhiyi Chi & Stefan Riezler: 1999, 'Estimators for stochastic "unification-based" grammars', in *Proc. of the 37th Annual Meeting of the ACL*, University of Maryland, pp. 535–541.

Lascarides, Alex & Ann Copestake: 1999, 'Default representation in constraint-based frameworks', *Computational Linguistics*, **25**(1): 55–106.

McIntyre, Andrew: 2001, 'Introduction to the verb-particle experience', Ms, Leipzig.

Nunberg, Geoffery, Ivan A. Sag & Thomas Wasow: 1994, 'Idioms', *Language*, **70**: 491–538.

Oepen, Stephan, Dan Flickinger, Hans Uszkoreit & Jun-ichi Tsujii: 2000, 'Introduction to the special issue on efficient processing with HPSG: methods, systems, evaluation', *Natural Language Engineering*, **6**(1): 1–14.

Pearce, Darren: 2001, 'Synonymy in collocation extraction', in *Proc. of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, CMU.

Pollard, Carl & Ivan A. Sag: 1994, *Head Driven Phrase Structure Grammar*, Chicago: University of Chicago Press.

Pulman, Stephen G.: 1993, 'The recognition and interpretation of idioms', in Cristina Cacciari & Patrizia Tabossi, eds., *Idioms: Processing, Structure and Interpretation*, Hillsdale, NJ: Lawrence Erlbaum Associates, chap. 11.

Riehemann, Susanne: 2001, 'A constructional approach to idioms and word formation', Ph.D. thesis, Stanford.

Sag, Ivan A. & Tom Wasow: 1999, *Syntactic Theory: A Formal Introduction*, Stanford: CSLI Publications.

A Hypothesis on the Origin of the Sign Types

Roland Hausser

Universität Erlangen-Nürnberg
Abteilung Computerlinguistik (CLUE)
rrh@linguistik.uni-erlangen.de

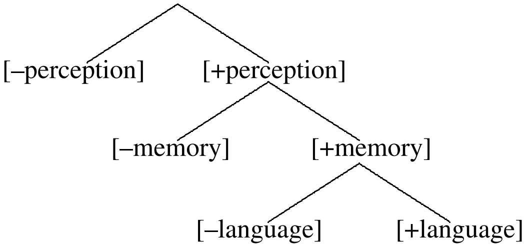
Abstract. The functioning of natural language communication depends crucially on the different kinds of signs, i.e., symbol, indexical, and name, and their characteristic mechanisms of reference. In this paper, the meanings of these sign types are traced to cognitive agents with memory but without language.
The argument is based on the task of relating the view-dependent format of a content stemming from an agent's current situation to a view-independent format for storage in memory. The view-dependent analysis of a content, called a task analysis, is built from cognitive structures suitable to serve as the literal meanings of the different sign types. Furthermore, the transfer between the view-dependent format relating to the current situation and the view-independent format of content stored in memory in [–language] agents is shown to be closely related to the alternation between the hearer and the speaker mode in [+language] agents.

1 Introduction

According to Aristotle (Metaphysics, I (A), 980b), all living beings have perception[1], but only some of them can store what they recognize in memory. The possibility of storing analyzed perception in memory is the precondition for having experience, for learning, and for deliberate action.

Furthermore, living beings with memory may be divided into those with language and those without. By representing these basic distinctions in terms of the binary features [± perception], [± memory], and [± language], we arrive at the following tree structure:

1.1 Basic Distinctions Characterizing Cognitive Agents



[1] While perception is a necessary condition for life, it is not sufficient. For example, we may grant that robots have perception, yet deny that they are alive.

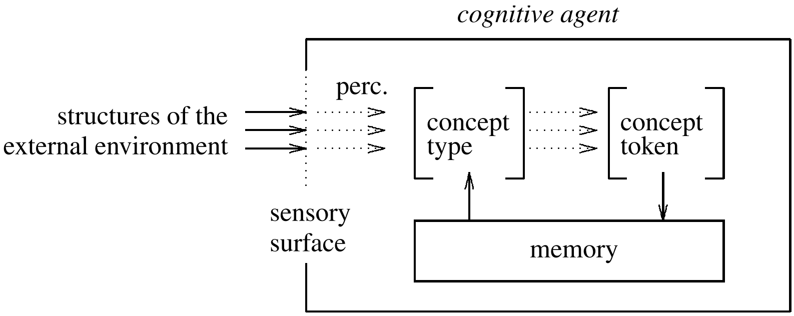
A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 16–31, 2002.

[–memory] agents must use perception to initiate action directly. From a computational point of view, this kind of reflex-connection between a certain type of perception and an associated action does not require the development of concepts.

[+memory] agents, in contrast, can store what they perceive without performing an external action and they can act without the stimulus of an associated perception. This requires first that perceptions are analyzed in terms of recognitions[2] and stored properly in memory. Second, it requires that intentions are derived and realized as actions.

The transfer of content into the data structure of a [+memory] system during recognition and out of the data structure during action is based on concept types and concept tokens. This is illustrated by the following schematic analysis of recognition:

1.2 Concept Types and Concept Tokens in Recognition



Perceptions of the cognitive agent's sensory surface are classified by a suitable concept type provided by memory. Concept types define the necessary properties of a concept by means of constants and the accidental properties by means of variables. For example, the concept type of the geometric object *square* is defined as follows:

1.3 Definition of the Concept Type of *Square*

$$\begin{bmatrix} \text{edge 1: } \alpha \text{ cm} \\ \text{angle 1/2: } 90^0 \\ \text{edge 2: } \alpha \text{ cm} \\ \text{angle 2/3: } 90^0 \\ \text{edge 3: } \alpha \text{ cm} \\ \text{angle 3/4: } 90^0 \\ \text{edge 4: } \alpha \text{ cm} \\ \text{angle 4/1: } 90^0 \end{bmatrix}$$

The necessary properties of this concept type are four angles of 90 degrees and four edges of equal length. The accidental property is the edge length, represented by the variable α. The variable makes the concept type applicable to squares of any size.

[2] What we call a recognition is sometimes called a percept. The latter term does not indicate, however, whether a percept is a raw sensation or a sensation classified in terms of a preexisting concept. Only the latter is called here a recognition.

When the concept type 1.3 is matched onto the incoming parameter values of 1.2, the variable is bound to a particular edge length, for example 2cm, resulting in the following concept token (which instantiates the concept type 1.3):

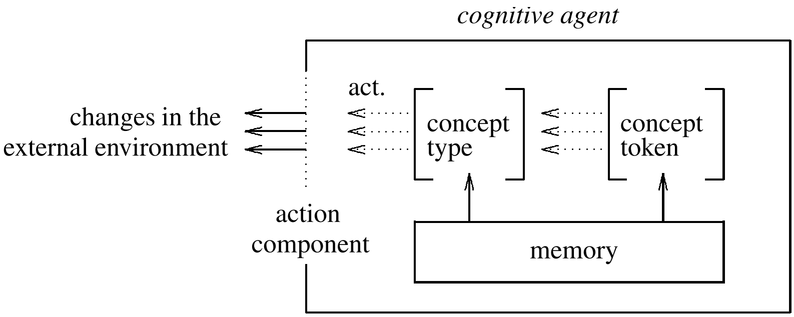
1.4 Definition of a Concept Token of *Square*

$$\begin{bmatrix} \text{edge 1: 2cm} \\ \text{angle 1/2: } 90^0 \\ \text{edge 2: 2cm} \\ \text{angle 2/3: } 90^0 \\ \text{edge 3: 2cm} \\ \text{angle 3/4: } 90^0 \\ \text{edge 4: 2cm} \\ \text{angle 4/1: } 90^0 \end{bmatrix}$$

This token may be stored in (the episodic part of) the agent's memory.

The inverse of perception and recognition are intention and action. Intention is the process of developing an action cognitively, while action is the mechanism of realizing an intention by changing the external environment.

1.5 Concept Types and Concept Tokens in Action



The formation of intentions is based on the agent's control structure, current situation, and inferences over content stored in memory. Intentions are represented as constellations of concept tokens and realized as actions by means of corresponding types.

In addition to the definition of concept types and concept tokens, the analysis of contextual (or non-language-based) cognition requires a data structure for indexing and retrieval, an activation of content by means of a time-linear navigation through the database (motor algorithm), a control structure, inferences, etc. (cf. Hausser 2001b).

The computational implementation of perception, recognition, intention, and action is a precondition not only for the construction of [+memory,–language], but also of [+memory,+language] agents. This is because non-language-based recognition and action are an important part of the *context* relative to which language is interpreted.

2 The Nonlinguistic Nature of the Internal Context

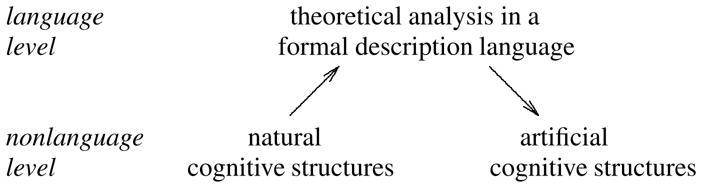
The different aspects of the external world, e.g., the program of a washing machine, the number of planets in the solar system, the atomic structure of the elements, the colors, etc., are inherently nonlinguistic in nature. To call these structures a ‘language’ is inappropriate because it would stretch the notion of a language beyond recognition.

The essentially nonlinguistic nature of the external originals holds also for their internal representations based on a cognitive agent’s recognition. Higher nontalking animals like a dog may well be able to develop something like concept types, to derive concept tokens, to combine them into elementary context propositions, to concatenate these into subcontexts, to draw inferences, and to derive view-dependent analyses of their current situation. These cognitive structures and procedures do not constitute a language, however, because they do not have external surfaces conventionalized in a community and do not serve in inter-agent communication. Instead, they evolve solely as internal, physiologically grown structures.

The contextual structures of a natural [–language] agent, e.g., a dog, acquire a language aspect only if and when they are being *described* by means of a language. Corresponding artificial systems, on the other hand, usually begin with a language-based definition which is then realized in terms of the hardware and software of the implementation. However, even in artificial systems the language aspect may be completely ignored once a system is up and running: on the level of its machine operations (electronic switching) the cognitive procedures of an artificial [–language] agent (non-talking robot) are just as nonlinguistic as those of a natural [–language] agent (dog).

The correlation between the nonlanguage and the language levels in the description of a natural [–language] agent and its artificial model may be described as follows:

2.1 Artificial Modeling of Natural Cognition


language level
theoretical analysis in a formal description language
nonlanguage level
natural cognitive structures
artificial cognitive structures

The point is that modeling the representation of context within a robot in terms of, for example, propositions based on feature structures and defining the procedures operating on these context structures in terms of a formal grammar are not in conflict with the essentially nonverbal character of these phenomena. Instead feature structures and grammatical algorithms are general abstract formalisms which may be used as much for the description of nonverbal structures as for the description of natural or artificial languages. Furthermore, once the nonverbal structures and the associated inferences have been implemented as electronic procedures they function without any recourse to the language that was used in their construction[3].

[3] The opposite position is taken by Fodor, who argues for (a) an internal language called ‘Mentalese,’ but (b) against ‘recognitional concepts’ like *red*. See Hausser 1999/2000, pp. 64, 65, for an analysis of Fodor’s fallacy.

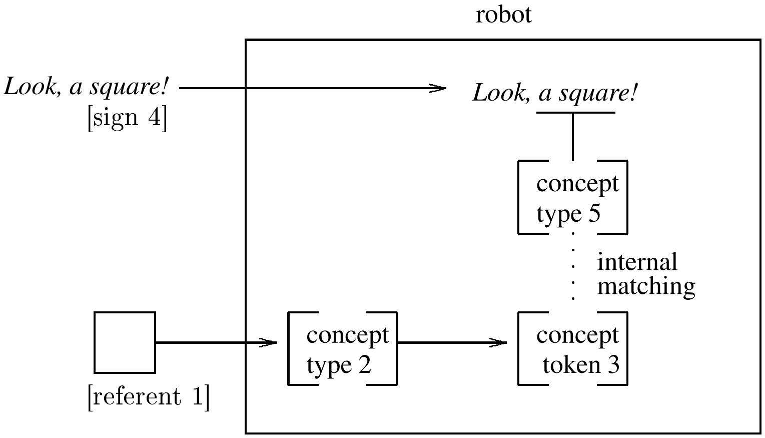
3 Adding Language

The primary capability of transferring content into and out of an agent's data structure by means of contextual recognition and action may be complemented by a secondary capability based on language. Language production (speaker mode, export of content) and interpretation (hearer mode, import of content) raise two related questions.

One is a question of coding and decoding: the speaker must code contextual content into natural language, and the hearer must decode natural language into a format resembling contextual content. The other is a question of indexing and retrieval: the speaker must specify in the language sign where in his database the content is coming from (indexing), and the hearer must infer the corresponding position in his database (retrieval) for storing the content correctly in order for communication to be successful.

The SLIM[4] theory of language approaches these questions by first reconstructing immediate reference. Its functioning is illustrated with a sign (square), a referent (geometric object), and an artificial [+language] agent (talking robot in hearer mode):

3.1 Immediate Reference Based on Internal Matching



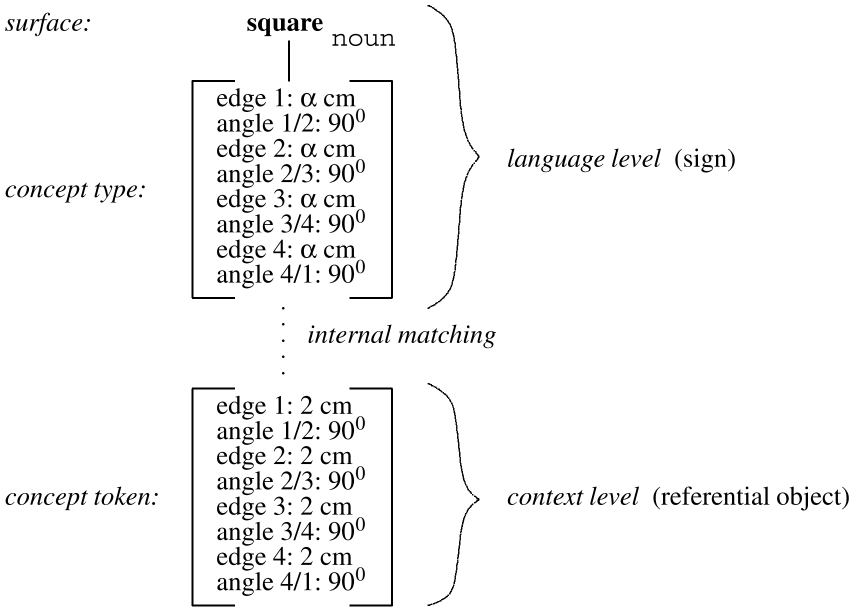
Instead of the usual analysis of reference as a relation between the external sign (4) and the external object of reference (1), reference is reconstructed by SLIM as a purely cognitive process within the cognitive agent. This cognitive process is based on the agent's recognition of the external language sign and of the external referent.

The semantic interpretation of the recognized language sign consists in lexically assigning a literal meaning, defined as a concept type (5) which is identical to the concept type (2). According to this analysis, the evolution of language is based in part on literal meanings which have evolved earlier as the concept types needed for the contextual recognition and action of [–language] agents.

The relation of reference between a sign at the level of language and a recognized object at the level of context is functionally established by the principle of internal matching between a concept type and a concept token:

[4] SLIM (Hausser 1999/2001) stands for the methodological, empirical, ontological, and functional principles of **S**urface compositionality, time-**L**inearity, and **I**nternal **M**atching.

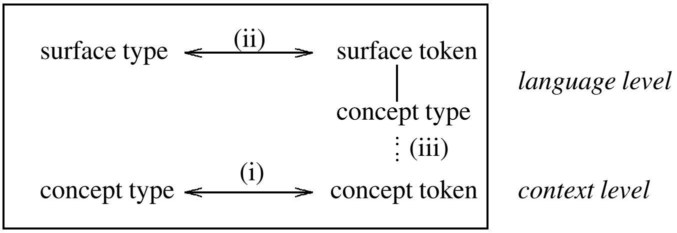
3.2 Internal Matching Based on the Type-Token Correlation



The sign is a fixed lexical structure consisting of (i) a surface (square), (ii) a syntactic category (noun), and (iii) a semantic interpretation defined as a concept type. The referential object is a concept token resulting from recognition. The pragmatics, i.e., the use of the sign in communication, consists in matching the sign's concept type (literal meaning) with a concept token (referential object) provided by the current context of use. SLIM's principle of internal matching (IM) models the flexibility which distinguishes the natural languages from the logical and programming languages.

The evolution of language requires two type-token relations which function in addition to the one characteristic of [–language] agents:

3.3 Three Type-Token Relations in [+language] Agents



The type-token relations arise (i) between concept types and concept tokens during contextual recognition (→) and action (←), (ii) between surface types and surface tokens during surface recognition (→) and synthesis (←), and (iii) between concept types and concept tokens during language interpretation (↓) and language production (↑).

4 Pragmatics

A central concern of semiotics as represented by C.S. Peirce (1839–1914) and C.W. Morris (1903–1979) is pragmatics. This concern is shared by the SLIM theory of language, though with the additional goal of arriving at a computational theory of natural language use in communication.

SLIM formulates pragmatics in terms of seven principles, of which the first relates the literal meaning[5] of language signs, called meaning$_1$, to the speaker meaning of utterances, called meaning$_2$:

4.1 First Principle of Pragmatics (PoP-1)

> The speaker's utterance meaning$_2$ is the use of the sign's literal meaning$_1$ relative to an internal context.

The crucial notion of use is implemented as an internal matching between literal meanings$_1$ at the level of language and referential objects at the level of context, as illustrated in 3.1 and 3.2 above with the sign type 'symbol'.

The second principle introduces the STAR relative to which a content is coded by the speaker (indexing) and decoded by the hearer (retrieval). The acronym STAR stands for the parameters of **S**pace, **T**ime, **A**uthor, and intended **R**ecipient of the sign[6].

4.2 Second Principle of Pragmatics (PoP-2)

> A sign's STAR determines the *entry context* of production and interpretation in the contextual database in terms of parameter values.

The STAR is crucial in the case of mediated reference, i.e., when the speaker refers to a context which is removed, for example, spatio-temporally, from the circumstances of utterance or when the hearer refers to a context which is removed from the circumstances of interpretation.

The third principle describes how complex signs (sequences of word forms in sentences or texts) are related to corresponding items at the level of context.

4.3 Third Principle of Pragmatics (PoP-3)

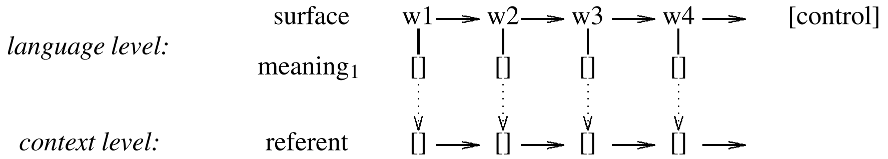
> The matching of the signs' meaning$_1$ with corresponding items at the level of context is incremental, whereby in production the elementary signs follow the time-linear order of the underlying thought path, while in interpretation the thought path follows the time-linear order of the incoming elementary signs.

In language interpretation, the navigation through the context is controlled by the language signs: the hearer follows the surfaces of the signs, looks up their meanings$_1$ in the lexicon, and matches them with suitable referents at the level of context:

[5] SLIM's use of literal meanings is in contrast to traditional semiotics, especially Morris. The aims of SLIM in comparison to other theories of language are discussed in Section 6 below.

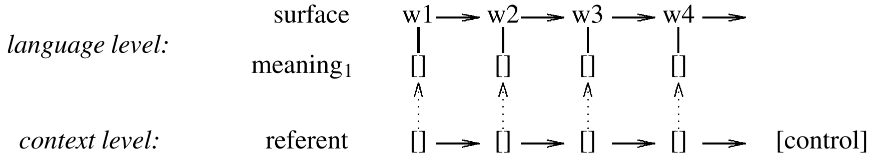
[6] The STAR is an extension of the 'point of speech' by Reichenbach 1947, p. 288, which refers to the temporal parameter T only and is terminologically restricted to spoken language. The STAR was first described in Hausser 1989, pp. 274 f. See also Hausser 1999/2001, pp. 93 f.

4.4 Schema of Language Interpretation (Hearer Mode)



In language production, the navigation control is located in the context, i.e., the data structure of the speaker's memory. Each unit traversed at the level of context is matched with the corresponding meaning$_1$ of a suitable word form:

4.5 Schema of Language Production (Speaker Mode)



Utterance of the word form surfaces allows the hearer to reconstruct the speaker's time-linear navigation path.

The schemata 4.4 and 4.5 agree with the natural view that interpretation (↓) and production (↑) are inverse[7] vertical procedures. Nevertheless, interpretation and production have their main direction in common, namely a horizontal time-linear structure (→) – in line with de Saussure's second law. The time-linear syntactic-semantic interpretation and contextual navigation are based on the algorithm of LA-grammar (Hausser 1992).

5 Different Kinds of Signs

The basic setup of pragmatic interpretation has been illustrated in 3.2 with a specific sign type, called *symbol*. Other sign types of natural language are *indexicals* like now, here, or this, and *proper names* like John or R2D2. In addition, there is the sign type of *icons*, which is marginal for synchronic natural language communication[8], but important for explaining the evolution of symbols.

In modern times, the theory of signs was founded by Peirce, who analyzes the sign types symbol, indexical, and icon, but omits names. Symbols are defined as follows:

> A symbol is a sign which would lose the character which renders it a sign if there were no interpretant. Such is any utterance of speech which signifies what it does only by virtue of its being understood to have signification.
>
> Peirce 1940, p. 104.

[7] This view is expressed, for example, by Mel'čuk 1988, p. 50.

[8] As pointed out by de Saussure 1967, pp. 81, 82. See Hausser 1999/2001, pp. 114 f.

Similarly, an index is defined as a sign which would lose the character which renders it a sign as soon as the object it is pointing at is removed; an icon is defined as a sign which retains its character as a sign even if there is no object to refer to, and no interpretant.

The disadvantage of Peirce's definitions is that they are unsuitable for computational implementation. Alternatively, SLIM explains the functioning of the different sign types, i.e., their respective mechanisms of reference, in terms of their cognitive structure. This analysis is formalized as the principles of pragmatics PoP-4 to PoP-6.

5.1 Fourth Principle of Pragmatics (PoP-4)

> The reference mechanism of the sign type **symbol** is based on a meaning$_1$ which is defined as a concept type. Symbols refer from their place in a positioned sentence by matching their meaning$_1$ with suitable contextual referents.

The reference mechanism of symbols is called iconic, because the functioning of symbols and icons is similar: both can be used to refer spontaneously to new objects of a known kind. The difference resides in the relation between the meaning$_1$ and the surface, which is arbitrary in the case of symbols, but motivated in the case of icons.

5.2 Fifth Principle of Pragmatics (PoP-5)

> The reference mechanism of the sign type **indexical** is based on a meaning$_1$ which is defined as a pointer. An indexical refers by pointing from its place in the positioned sentence to a value in an appropriate parameter.

Indexical reference is illustrated by the adverbs here and now, which point to values in the spatial and temporal parameters of an utterance, respectively, and the pronouns I and you, which point to the author and the intended recipient, respectively.

5.3 Sixth Principle of Pragmatics (PoP-6)

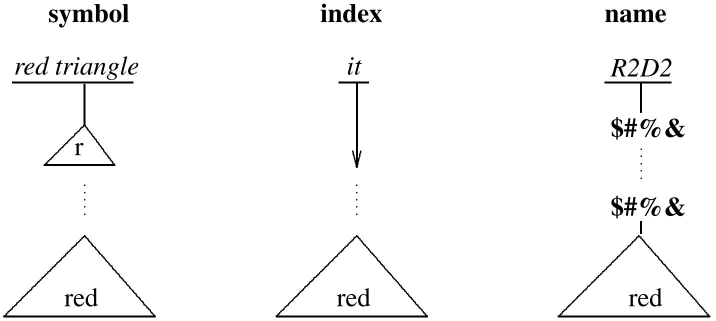
> The reference mechanism of the sign type **name** is based on a meaning$_1$ defined as a private marker which corresponds to a private marker contained in the cognitive representation of the corresponding referential object. Reference with a name consists in matching the two private markers[9].

As an example, consider agent A observing a dog. For easier reference, the cognitive structure representing the dog in agent A's context is abbreviated by the private marker $#%&. Later, another agent calls the dog by the name Fido. Agent A adopts this name by attaching the private marker $#%& to the public surface Fido. Henceforth, the name Fido refers for A to the dog in question by matching the private marker attached to the name with the corresponding marker attached to the referent.

The respective structural basis of iconic, indexical, and name-based reference is illustrated in the following schematic comparison in which the three sign types are used to refer to the same contextual object, i.e., a red triangle.

[9] In analytic philosophy, names have long been a puzzle. Attempts to explain their functioning range from *causal chains* to *rigid designators* (cf. S. Kripke 1972). The present analysis is more general than the one in Hausser 1999/2001.

5.4 Comparing Iconic, Indexical, and Name-Based Reference



All three sign types have a meaning$_1$ which is firmly attached to a surface. The symbolic expression red triangle refers on the basis of the type-token relation: the meaning$_1$ is a concept type which matches a corresponding concept token in a limited range of contextual candidates. The indexical it refers on the basis of pointing: the meaning$_1$ is a characteristic pointer which points at the referential object within an associated parameter (here, third person, i.e., everything that is neither author nor addressee). The name R2D2 refers on the basis of matching private markers: the meaning$_1$ is $#%&, which is matched with an identical marker in the contextual referent.

All three mechanisms of reference must be analyzed as internal, cognitive procedures because it would be ontologically unjustifiable to locate the fixed connections between their signs' surface and meaning$_1$ in external reality. Instead, meaning$_1$ is assigned cognitively by means of a lexicon which all speakers-hearers have to learn.

For explaining the phylo- and ontogenetic development of natural language it is of interest that the basic mechanisms of iconic and indexical reference[10] constitute the foundation of nonverbal and preverbal communication as well. Thereby

1. nonverbal iconic reference consists in spontaneously imitating the referent, and
2. nonverbal indexical reference consists in pointing at the referent.

While essentially limited to face-to-face communication, these nonverbal mechanisms of reference may be quite effective. By avoiding the use of conventionally established surfaces, nonverbal reference allows spontaneous communication in situations in which no common language is available.

It is important to note that the distinction between the different *sign types*, i.e., symbol, indexical, and name, is orthogonal to the distinction between the main *parts of speech*, i.e., noun, verb, and adjective, as well as to the corresponding distinction between the basic *elements of propositions*, i.e., argument, functor, and modifier.

5.5 Seventh Principle of Pragmatics (PoP-7)

> The sign type *symbol* occurs as noun, verb, and adjective. The sign type *indexical* occurs as noun and adjective. The sign type *name* occurs only as noun.

[10] The early form of name-based reference is not included here because it constitutes the transition to language-based communication. See Hausser 1999/2001 for comparison.

The orthogonal correlation between sign types and parts of speech described in PoP-7 may be illustrated graphically as follows:

5.6 Relation between Sign Types and Parts of Speech

	noun	adjective	verb
name	*Peter*		
indexical	*this*	*now*	
symbol	*triangle*	*red*	*contain*

The sign type which is the most general with respect to different parts of speech is the symbol, while the name is the most restricted. Conversely, the part of speech (and, correspondingly, the propositional element) which is the most general with respect to different sign types is the noun (object), while the verb (relation) is the most restricted.

6 The Role of Context in Communication

The SLIM-theoretical analysis of natural language aims at a functional model of communication. For functioning in the real world, the model has to be procedural rather than metalanguage-based.

A procedural model must have a declarative specification defining its necessary properties. In contradistinction to a meta-language based approach, however, it is not dependent on set theory to provide immediately obvious basic meanings.

Instead, basic meanings are programmed as concept types for classifying incoming parameter values, such as colors or geometric shapes, and for realizing outgoing intention tokens, such as certain actions of locomotion or gripping. Furthermore, the agent's saying something true is characterized in terms of contextual recognition and action, as well as language interpretation and production, working properly.

SLIM differs from previous theories of language because it provides an objectively testable method of verification. This method consists in building artificial cognitive agents which can recognize new objects of a known kind in their real world environment, talk about which objects they found in the past or which objects they hope to find in the future, understand questions and commands regarding their experiences and actions, etc.[11]

For this, the different sign types must be analyzed in terms of their cognitive structure and associated functioning – in contradistinction to the classificational approach of Peirce. As we have seen, Peirce distinguishes between symbols, indices, and icons in terms of whether or not there has to be an interpretant, and whether or not there has to be a referent, in order for the respective types of signs 'to retain their character.'

[11] For a simple 'fragment' comprising language understanding, conceptualization, language production, inferencing, and querying, see Hausser 2001b.

The realization of SLIM in terms of a functioning artificial agent cannot prove that this theory is the only one correct. However, given alternative theories, the one functioning best is to be preferred – provided the theoretical goal is a computational model of how communication works and the practical goal comprises unrestricted human-computer communication. Note that previous theories of language, such as pragmatism, structuralism, behaviorism, model theory, speech act theory, or nativism, have not been designed for these goals. Not surprisingly, they are unsuitable for reaching them.

In addition to functional performance, SLIM may be supported by other desiderata of scientific research, such as compatibility with results from psychological experiments, findings of neurology, or a plausible explanation of how [+language] agents have evolved from [–language] agents[12]. Regarding the latter, SLIM is special in comparison to previous theories of language in that it begins with an explicit definition of the cognitive agent's internal context. This is motivated as follows:

First, modeling the agent's context as a database is functionally necessary for realizing certain communication types[13], such as language-controlled action (telling the robot what to do) and commented recognition (the robot describing what it sees). It is also necessary in mediated reference, when content independent of the current task environment (for example, regarding past events) is being read into and out of the contextual database by means of natural language.

Second, starting with modeling the context in [–language] agents and then building the language level on top of it is in concord with evolution. Thereby the development of concept types needed for contextual recognition and action in [–language] agents (cf. Section 1) provides cognitive structures suitable to serve as the meaning$_1$ of symbols in [+language] agents: the evolution of this sign type merely requires *reusing* already available concept types by attaching them to categorized surfaces (cf. Section 3).

The question now is whether the evolution of the other sign types can be described in a similar manner: do [–language] agents have a need to develop the indexical pointers and the private name markers, such that these cognitive structures may be simply reused in the evolution of the corresponding sign types by attaching them to categorized surfaces?

7 Relating Stored Content to the Current Situation

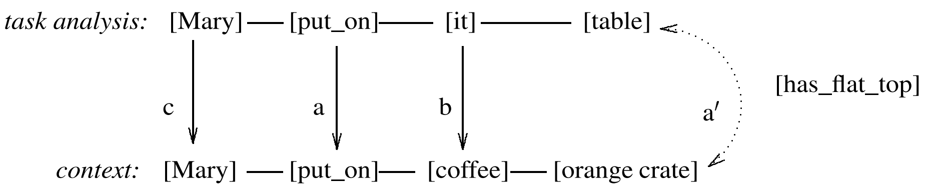
From the viewpoint of evolution it seems plausible that the meaning$_1$ of the different sign types develops already in [–language] agents. Our argument for this, however, is functional in nature. It is based on the computationally motivated hypothesis that [–language] agents need a simplified, purpose-oriented view which is superimposed on the many details of their current recognitions and intentions.

Such a view-dependent representation is called a *task analysis*. This momentary construction exists in addition to and simultaneously with the corresponding context. While the context constitutes the content's literal representation, the task analysis *refers* to the content by using indexical pointers and metaphorically used concepts in addition to literally used concepts and private markers. Consider the following example:

[12] A major flaw of Grice's 1965 theory of sentence meaning and utterer's meaning is that it cannot explain how types initially evolved – as pointed out, for example, by Searle 1969, p. 44f.

[13] They are classified as SLIM-1 – SLIM-10 in Hausser 1999/2001, pp. 469–473.

7.1 Primary Task Analysis of an Immediate Context



The above context is assumed to be a complex situation consisting of many facts. Most of them are omitted in the representation, however, for the sake of simplicity, for example, the shape of the room, the color of the walls, the position of the windows, etc. From these, a small subset, represented as the proposition *Mary put_on coffee orange-crate*, is selected by the task analysis, thus providing a simplified, purpose-oriented extract[14].

The task analysis is a secondary representation which selects corresponding referents at the level of context by means of concept types (a), indexical pointers (b), and private markers (c). The selection is guided by the cognitive agent's habits, knowns, unknowns, likes, dislikes, needs, and purposes. The latter may even lead to viewing, for example, an orange crate metaphorically as a table (a′), based on the property *has flat top*. The simultaneous two-level representation has the following functions:

First, for recognition and action, the task analysis complements the current immediate context with a simplified, purpose-oriented representation which highlights relevant and/or familiar patterns[15]. This simplified representation helps to keep track of referents when interacting with the task environment in a sequence of recognitions and actions.

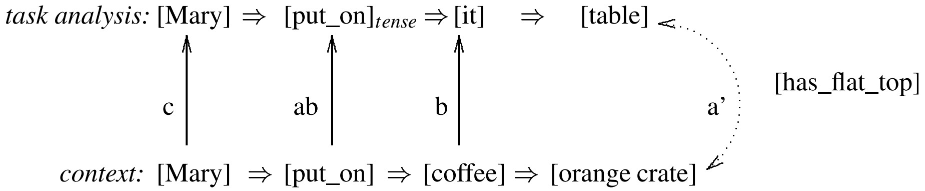
Second, for storage in memory, the task analysis selects relevant aspects from the current immediate context to avoid overflow. When the selected content is stored in memory, however, the task analysis' view-dependent indexical and metaphorical aspects must be eliminated. This requires a mapping from the view-dependent task analysis to a view-independent representation suitable for long-term storage.

Third, for relating stored view-independent content to the current situation, a *secondary* task analysis is constructed from the viewpoint of the cognitive agent's current tasks and purposes. This requires a mapping from the view-independent representation of long-term storage to a representation which takes the agent's current viewpoint into account. Consider the following example:

[14] For better readability, the private name marker is represented as 'Mary' in 7.1 rather than $#%&, the concepts as 'put_on' and 'table' rather than explicit concept types at the level of the task analysis and concept tokens at the level of context, and the indexical pointer as 'it' rather than an explicit vector pointing at a certain value of a certain parameter.

[15] The simultaneous evolution and separation of the levels of context and task environment seems to have occurred quite gradually. In a frog, for example, the level of context impinges on a primitive task analysis realized as a prewired array of reflexes, such as *small moving spot–jump for it* or *larger shadow–go for cover*.

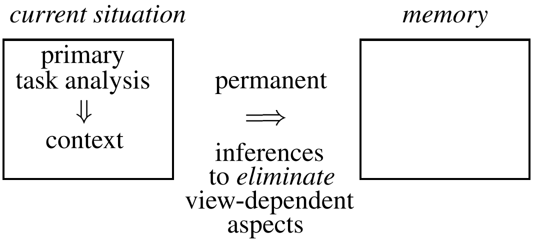
7.2 Secondary Task Analysis of a Stored Content


task analysis: [Mary] ⇒ [put_on]tense ⇒ [it] ⇒ [table]
c
ab
b
a’
[has_flat_top]
context: [Mary] ⇒ [put_on] ⇒ [coffee] ⇒ [orange crate]

The proposition *Mary put_on coffee orange-crate* has been stored in memory for some time. When the proposition is activated by means of navigating through it (cf. context), a secondary task analysis is constructed. It expresses the proposition's temporal relation to the current situation indexically by assigning tense to the verb (ab). Furthermore, assuming that the referent *coffee* has already been activated, the secondary task analysis represents it indexically (b). Like a primary task analysis, a secondary task analysis may also refer metaphorically, such as viewing the orange-crate as a table (a′).

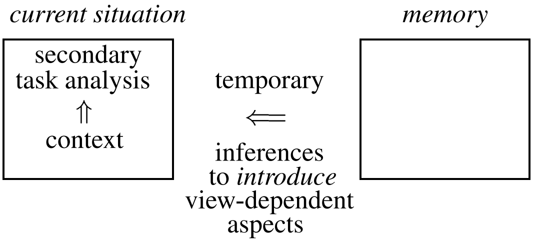
When the view-dependent selection from a current content provided by a primary task analysis is read into memory, it must be modified into a view-independent format by way of inferences. The result is permanent insofar as the view-independent content stored in memory may remain unchanged over time.

7.3 Storage in Memory



Conversely, when a content stored in memory is adapted to the current situation, its view-independent format is complemented by a view-dependent secondary task analysis, based on inferences. The result is temporary insofar as (i) the current situation may change quickly and (ii) the same content may be adapted to different current situations.

7.4 Retrieval from Memory



Formalized examples of the inferences eliminating and introducing view-dependent aspects are presented in Hausser 1999/2001[16].

8 Conclusion

The derivation of task analyses for (i) interacting with the current situation, (ii) storing current interactions in memory, and (iii) adapting stored content to the current viewpoint in [–language] agents prepares the evolution of natural language as follows:

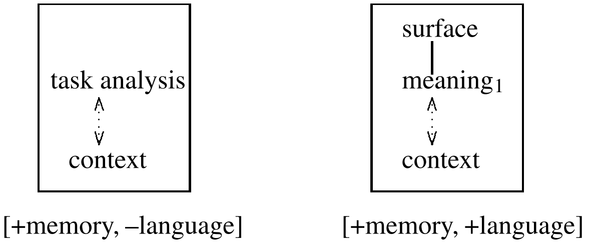
1. The elements from which task analyses are built are suitable to be reused as the meaning$_1$ of symbols, indexicals, and names such that the emergence of these sign types requires no more than attaching the preexisting meanings$_1$ to public surfaces.
2. The inferences eliminating the view-dependent aspects of a primary task analysis for the purpose of long-term storage are essential also for the hearer mode of natural language interpretation.
3. The inferences guiding the choice of elements in the construction of a view-dependent secondary task analysis, e.g., a nonliteral use of a concept type, a pointer, or a private marker, are needed also in the speaker mode of natural language production.

More generally, the [–language] agents' need to (i) transform view-dependent content into a view-independent format suitable for storage and (ii) adapting view-independent content to a view of the current situation has a direct counterpart in [+language] agents, namely the switching between the hearer mode and the speaker mode.

Consider the following example: A [–language] agent stores at some point in time the sequence of propositions *I am visiting the Munchkins. The Munchkins serve coffee. Mary puts coffee on orange crate.* When this content is activated one year later in memory, it may be adapted to the present situation by the following task analysis: *I visited the Munchkins last year. They served coffee. Mary put_on it table.*

The point is that a [–language] agent's cognitive adaption of the stored content to the present situation requires the same modifications as a [+language] agent's corresponding coding (speaker mode) into natural language. The only difference is the absence vs. presence of language surfaces, as shown below:

8.1 Comparing [–language] and [+language] Agents



[16] For inferences eliminating view-dependent aspects see p. 499, 24.4.4 and 24.4.6, for inferences introducing view-dependent aspects see p. 495, 24.4.3, and p. 499, 24.4.5.

The analogous situation holds for the opposite direction of eliminating view-dependent aspects, which corresponds to the operations necessary for the hearer's interpretation[17].

In summary, the crucial qualitative contribution of language over and above the cognitive functions of [–language] agents consists in the added ability of *communicating* content from one agent to another by using the signs' public surfaces. It goes without saying that this added qualitative ability has enormous quantitative consequences with regard to the amount of knowledge potentially available to [+language] agents.

References

Davidson, D. & G. Harman (eds.) (1972) *Semantics of Natural Language*, D.Reidel, Dordrecht.

Fodor, J. (2000) *In Critical Condition, Polemical Essays on Cognitive Science and the Philosophy of Mind*, Bradford Books.

Grice, P. (1965) "Utterer's Meaning, Sentence Meaning, and Word Meaning," *Foundations of Language*, 4:1–18.

Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283-308, Elsevier.

Hausser, R. (2001a) "Spatio-Temporal Indexing in Database Semantics," in A. Gelbukh (ed.) *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science Vol. 2004. Springer-Verlag, Berlin, New York.

Hausser, R. (2001b) "Database Semantics for Natural Language," *Artificial Intelligence*, Vol. 130.1:27–74, Elsevier.

Hausser, R. (1999/2001) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language*, Springer-Verlag, Berlin, New York, 2nd Ed.

Kripke, S. (1972) "Naming and Necessity," in Davidson & Harmann (eds.), 253–355.

Mel'čuk, I. A. (1988) *Dependency Syntax: Theory and Practice*, State University of New York Press, New York.

Morris, C.W. (1946) *Signs, Language, and Behavior*, New York.

Peirce, C.S. (1940) *The Philosophy of Peirce: Selected Writings*. J. Buchler (ed.), London.

Reichenbach, H. (1947) *Elements of Symbolic Logic*, The Free Press, New York.

Saussure, F. de (1972) *Cours de linguistique générale*, Édition critique préparée par Tullio de Mauro, Éditions Payot, Paris.

Searle, J.R. (1969) *Speech Acts*, CUP, Cambridge, England.

[17] An important special case requiring inferences beyond those needed for introducing and eliminating task analyses is the interpretation of permanent natural language signs, e.g., a letter. For a detailed analysis see Hausser 2001a, Section 7, or Hausser 1999/2001, Section 22.5.

Quantification and Intensionality in Situation Semantics

Roussanka Loukanova

Illinois Wesleyan University,
Department of Mathematics and Computer Science,
P.O. Box 2900, Bloomington, IL 61702-2900, USA
rloukano@iwu.edu

Abstract. In this paper we present a type-driven interpretation approach to semantic theory. We will introduce a formalization of the notion of the speaker's references in a context, and a semantical storage mechanism for resolving the quantificational and *de re/de dicto* scope ambiguities. In Montague Grammar (PTQ) all verbs are initially treated as intensional, and then the extensional translations, whenever they exist, are derived from the intensional ones by the use of meaning postulates. Such a treatment of the extensional verbs, besides of being counterintuitive, increases dramatically the complexity of the computations of the semantical representations of the sentences, which exponentially propagates to the text processing. The paper introduces a lexicalized situation semantics treatment of the quantifiers and the intensional verbs. The approach is illustrated by a grammar, which uses a semantical storage. Primarily, the semantic storage targets the anaphoric and quantificational relations and related scope ambiguities, but it also facilitates the represention of the *de re/de dicto* "reading" capacities of the intensional verbs. The grammar accounts for the difference between extensional and intensional verbs by the means of *appropriateness conditions* for filling the semantical argument roles of the relations denoted by the verbs. The semantical rules for the verb phrases result in type-driven calculations triggered by discourse information.

In Montague's PTQ (see Dowty at al. (1981)) all verbs are originally treated as intensional. Meaning postulates permit the translation of an expression having as a constituent either an extensional verb, or an intensional one in its extensional reading, to be calculated out of the corresponding initially intensional translation. Such an approach leads inevitably to an exponential growth of the complexity of the computations of the semantical representations. The current approach to situation semantics does not use any meaning postulates for deriving extensional semantical representations out of the intensional ones. Rather, the difference between extensional and intensional verbs is governed by constrains about what kinds of objects are appropriate to fill up the argument roles of the relations denoted by the verbs.

The lexicon component of the grammar considered in this paper is responsible for the relevant semantical constrains and classification of the verbs into exten-

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 32–45, 2002.

sional and intensional. The grammar, including the lexicon, can be organized either as an HPSG, or LFG. The lexical entries of the verbs can be represented by complex structural descriptions of syntactical and semantical information, for example, in the form of feature structures. The current semantical approach and the semantical rules given are neutral with respect to any particular grammar system. What is important for the current treatment is that the semantical information is represented by situation theoretical objects and that the semantical component of each verb, and common noun, gives the semantical argument structure of the relation (or property) it denotes. The argument roles of the relations are associated with semantical restrictions[1], i.e. with *appropriateness conditions* for filling them. For introducing the appropriateness conditions in the semantical representations (which are situation theoretical objects), the lexicon of the suggested grammar uses the notion of a restricted parameter.

When a given verb permits either extensional or intensional interpretations, which of the readings is appropriate for it, is up to the specific context of use. Generally, the speakers know (consciously or not) the possible options for the argument roles of a given verb — the circumstances of the discourse, and the speaker's conversational intentions trigger the appropriate selections. The grammar represented in this paper uses a semantical storage to keep the available semantical ambiguities unresolved in absence of enough information. A semantical operator uses the storage to resolve ambiguities, when there is appropriate linguistic or contextual information for that, and by respecting the appropriateness conditions of the argument roles.

The semantics of the natural language expressions is concerned with two interrelated sides: the linguistic meaning of an expression in abstraction of any particular context, and its interpretation in one or other context of use. In order to enable the calculations of the interpretations of the utterances by using the linguistics meanings and the specific context information, the current grammar defines the *linguistic meaning* of an expression as a pair of a *semantical storage* and a *semantical basis.* The storage is a set of semantical representations of quantifiers, the scopes of which are pending unresolved. A generalized quantification AL operator is defined over the storage and the basis. It moves some, or all of the quantifiers from the storage to the basis in a particular order. Some quantifiers can be moved at the level of the calculation of the pure linguistic meaning depending on linguistic factors, others — at the level of the interpretation in a particular context. A particular scope reading of a sentence is obtained when the storage has been emptied by moving all quantifiers into the basis.

The quantificational operator, which moves quantifier representations from the storage to the basis, is a subject of structural restrictions, introduced in

[1] These are, typically, semantical restrictions that do not have explicit syntactical manifestation. For example, although the agreement properties such as number, person and gender, are semantical in nature, they have regular syntactical representation and the agreement information can be part of the syntactical components of the grammar descriptions.

Loukanova (2002, this volume), that do not permit free parameters to fall out of the scope of the quantificational binding.

1 Introduction and Some Notations

For an introduction to situation theory and very detailed information on the existing literature on the topic, see Seligman and Moss (1997). Brief introduction into the situational notions and notations can be found in Cooper (1992) and Loukanova (2000). For a modern approach to situation semantics and in particular, a treatment of interogatives, see Ginzburg and Sag (2001). The following propositions and types shall be used in the paper:

- $pu(u, l, x, y, \alpha) = (u \models \ll \mathit{tells\_to}, x, y, \alpha, l; 1 \gg)$, this is the proposition expressing who is the speaker x, who is the listener y, what is the space-time location, and which is the expression α uttered in an utterance situation u, i.e. a minimum of context information;
- $ru(l, x, y, \alpha) = [u/pu(u, l, x, y, \alpha)]$, the abstract type of an utterance situation;
- $rsp(u, l, y, \alpha) = [x/pu(u, l, x, y, \alpha)]$, the type of an individual to be the speaker in the utterance situation u;
- $rlst(u, l, x, \alpha) = [y/pu(u, l, x, y, \alpha)]$, the type of an individual to be the listener in the utterance situation u;
- $rdl(u, x, y, \alpha) = [l/pu(u, l, x, y, \alpha)]$, the type of an object to be the utterance (or discourse) location;
- $r_\varphi(u, l, x, y, s_{res}) = [z/q(u, l, x, y, z, \alpha)]$, the type of an object to be referred to by the expression α, where
- $q(u, l, x, y, z, \alpha) = (u^{ru(l,x,y,\alpha)} \models$
  $\ll \mathit{refers\_to\_by\_in}, x^{rsp(u,l,y,\alpha)}, z, \alpha, l^{rdl(u,x,y,\alpha)}; 1 \gg)$,
  this is the proposition asserting that the speaker x^{rsp} refers to z by using the expression α. More elaborate representation of the names can be expressed by the following aversion of the proposition $q(u, l, x, z, \alpha)$[2]:

- $$\begin{aligned} q(u, l, x, y, z, \alpha, s_{res}) = (u^{ru(l,x,y,\alpha)} \models & \\ & \ll \mathit{refers\_to\_by}, x^{rsp(u,l,y,\alpha)}, z, \alpha, l^{rdl(u,x,y,\alpha)}; 1 \gg \wedge \\ & \ll \mathit{believes},\ x^{rsp(u,l,y,\alpha)}, \\ & \qquad (s_{res} \models \ll \mathit{named\_}\alpha, z; 1 \gg), \\ & \qquad l^{rdl(u,x,y,\alpha)}; 1 \gg) \end{aligned}$$

The last proposition asserts that the speaker x^{rsp} refers to z by using the name α and believing that z is named α. In what follows, all the abode restrictions shall be written without explicitly specifying the parameter arguments.

[2] For an excellent discussion of the above types and propositions, see Barwise and Perry (1983).

2 Lexicon

The approach taken in the present grammar permits two alternatives for representation of the APS that are proper names and pronouns, and called *individual terms.* If α is an individual term, then:

(Alt.1) $\mathcal{B}(\alpha) = x_i$, and $\mathcal{M}(\alpha) = \{\langle \lambda s\,[T/(T : x_i^{restr})], x_i\rangle\}$,
(Alt.2) $\mathcal{B}(\alpha) = x_i^{restr}$, and $\mathcal{M}(\alpha) = \emptyset$,

where *restr* represents the semantical information carried by the individual term α. In a particular context of use, the NP α gets its referent by assigning (anchoring) the restricted parameter x_i^{restr} to a particular individual a that has to satisfy the restriction *restr*. The first alternative is more in Montagovian style, while the second one is close to the treatment of the singular NPs, proposed in Barwise and Perry (1983). We shall see that second alternative, along with being more simple, is consistent with the quantificational restrictions defined later in this grammar.

Proper Names. If α is a proper name, then α_i is a NP. For simplifying the statement of the rules in this grammar, all names are indexed with natural numbers.

(Alt.1) $\mathcal{B}(\alpha) = x_i$ and $\mathcal{M}(\alpha) = \{\langle \lambda s\,[T/(T : x_i^{r_\alpha})], x_i\rangle\}$.
(Alt.2) $\mathcal{B}(\alpha) = x_i^{r_\alpha}$, and $\mathcal{M}(\alpha) = \emptyset$.

Pronouns. I, YOU, HE$_i$, SHE$_i$, IT$_i$ are NPs.

(Alt.1) $\mathcal{B}(\mathrm{I}) = x_i$ and $\mathcal{M}(\mathrm{I}) = \{\langle \lambda s\,[T/(T : x_i^{rsp})], x_i\rangle\}$;
$\mathcal{B}(\textsc{you}) = x_i$ and $\mathcal{M}(\textsc{you}) = \{\langle \lambda s\,[T/(T : x_i^{rlst})], x_i\rangle\}$;
$\mathcal{B}(\textsc{he}_i) = x_i$ and $\mathcal{M}(\textsc{he}_i) = \{\langle \lambda s\,[T/(T : x_i^{masc})], x_i\rangle\}$,
$\mathcal{B}(\textsc{she}_i) = x_i$ and $\mathcal{M}(\textsc{she}_i) = \{\langle \lambda s\,[T/(T : x_i^{feminine})], x_i\rangle\}$, ... where

masc and *feminine* are the types of objects being of masculine or feminine, respectively, personification.

(Alt.2) $\mathcal{B}(\mathrm{I}) = x_i^{rsp}$ and $\mathcal{M}(\mathrm{I}) = \emptyset$;
$\mathcal{B}(\textsc{you}) = x_i^{rlst}$ and $\mathcal{M}(\textsc{you}) = \emptyset$;
$\mathcal{B}(\textsc{he}_i) = x_i^{masc}$ and $\mathcal{M}(\textsc{he}_i) = \emptyset$,
$\mathcal{B}(\textsc{she}_i) = x_i^{feminine}$ and $\mathcal{M}(\textsc{she}_i) = \emptyset$, ...

Common Nouns (N). If α is a lexical common noun, then $\mathcal{F}(\alpha)$ is a primitive relation with at least two argument roles[3], *Arg* and *Loc*. The *Arg*-role is for the object having the property denoted by the noun, the *Loc*-role — for the space-time location where the object has that property. The argument roles are associated with some minimal appropriateness conditions over the objects filling them up, and are represented by the set: $\mathcal{ARG}(\mathcal{F}(\alpha)) = \{<Arg, Appr>, <Loc, \mathrm{S}PT>\}$,

[3] In this paper, we do not consider the relational nouns.

where *Appr* is a type representing the appropriateness conditions of the *Arg* role of the noun, and *SPT* is the primitive type of the objects being space-time locations. Generally, in this sample grammar, we shall not be careful about giving the appropriateness restrictions, with the important exception for distinguishing between the intensional and extensional verbs. The storage and the basis of α are defined in the following way:

$\mathcal{M}(\alpha) = \emptyset$,
$\mathcal{B}(\alpha) = \lambda s, l\,[x/(s \models \ll \mathcal{F}(\alpha), x^{Appr}, l; 1 \gg)]$.
For example:
$\mathcal{B}$(STUDENT)$= \lambda s, l\,[x/(s \models \ll \textit{student}, x^{r_1}, l; 1 \gg)]$, where

(4.1) $r_1 = [x/(s \models \ll \textit{human\_being}, x, l; 1 \gg)]$.

Determiners (DET). Only determiners that can be treated as generalized quantifiers shall be considered[4]. If $\delta \in \{$A,THE,EVERY,NO, MOST, $\ldots\}$, then δ is a DET, and its semantical representative is the primitive quantificational relation $\mathcal{F}(\delta)$. It is associated with two argument roles: $\mathcal{ARG}(\delta) = \{< \textit{QDomain}, \textit{TI} >, < \textit{QRange}, \textit{TI} >\}$, where TI is the primitive type of the objects that are types (i.e. properties) of individuals, and

$\mathcal{M}(\delta) = \emptyset$,
$\mathcal{B}(\delta) = \lambda s\,\lambda T_1\,\lambda T_2 (s \models \ll \mathcal{F}(\delta), T_1, T_2; 1 \gg)$, where

T_1 and T_2 are type parameters that fill up correspondingly the *QDomain* and the *QRange* roles of the determiner $\mathcal{F}(\delta)$. In this paper, we shall follow the traditional "bracketed" notation of λ-abstraction over propositions:

$$\mathcal{B}(\delta) = \lambda s\,\lambda T_1\,[T_2/(s \models \ll \mathcal{F}(\delta), T_1, T_2; 1 \gg)].$$

Verbs. If α is a verb, then $\mathcal{F}(\alpha)$ is a primitive relation with some argument roles, among them—a space-time location role:

$$\mathcal{ARG}(\mathcal{F}(\alpha)) = \{< \textit{Arg}_1, \textit{Appr}_1 >, \ldots, < \textit{Arg}_k, \textit{Appr}_k >, < \textit{Loc}, \mathrm{S}PT >\},$$

$\mathcal{M}(\alpha) = \emptyset$.

The basis $\mathcal{B}(\alpha)$ is defined in the following way depending on whether α is an intransitive or transitive verb:

Intransitive Verbs (IV):

$\mathcal{B}(\alpha) = \lambda s, l, t\,[x/(s \models \ll \mathcal{F}(\alpha), x^{Appr}, l^{SPT}; t \gg)]$,

[4] A rule for treating A, THE as singular indefinite and definite determiners, respectively, can be added to the grammar.

Transitive Verbs (TV):

$$\mathcal{B}(\alpha) = \lambda s, l, t\, \lambda y\, [x/(s \models \ll \mathcal{F}(\alpha), x^{Appr_1}, y^{Appr_2}, l^{SPT}; t \gg)].$$

In both above cases, t is a parameter for a polarity value of 0 or 1. For example,

$\mathcal{ARG}(read) = \{< Subj, r >, < Obj, r' >, < Loc, SPT >\}$, where

(5.1) $r = [x/(s \models \ll living\_being, x; 1 \gg)]$,
(5.2) $r' = [x/(s \models \ll readable, x; 1 \gg)]$, and

$\mathcal{M}(\textsc{read}) = \emptyset$,
$\mathcal{B}(\textsc{read}) = \lambda s, l, t\lambda y[x/(s \models \ll read, x^r, y^{r'}, l; t \gg)]$,

Tense Markers. We shall consider only two tense markers, -PR.C. and -PST that represent Present Continuous and Simple Past. Let $\circ$ and $\prec$ be, correspondingly, the relations of space-time overlapping and time precedence:
$[\![\text{-PR.C.}]\!] = [l/l \circ l^{rdl}]$,
$[\![\text{-PST}]\!] = [l/l \prec l^{rdl}]$.

3 Tensed Verbs (IV$_t$) and (TV$_t$)

If α is either an IV, or a TV, and τ is a Tense Marker, then α_τ and $\sim \alpha_\tau$ are, correspondingly IV$_t$, or TV$_t$ (for simplicity, only two-valence verbs are considered). Here, $\sim \alpha_\tau$ stands for the negative form of the verb. The storage and the basis are:
$\mathcal{M}(\alpha_\tau) = \mathcal{M}(\sim \alpha_\tau) = \emptyset$,
$\mathcal{B}(\alpha_\tau) = \lambda s, l\, (\mathcal{B}(\alpha)(s, l^{[\![\tau]\!]}, 1))$ and
$\mathcal{B}(\sim \alpha_\tau) = \lambda s, l\, (\mathcal{B}(\alpha)(s, l^{[\![\tau]\!]}, 0))$.

Example 1. $\mathcal{B}(\textsc{run-pr.c.}) =$

$$\lambda s, l\, (\lambda s, l, t\, [x/(s \models \ll run, x, l; t \gg)](s, l^{[l/l \circ l^{rdl}]}, 1)) =$$
$$\lambda s, l\, [x/(s \models \ll run, x, l^{[l/l \circ l^{rdl}]}; 1 \gg)].$$

Example 2. $\mathcal{B}(\textsc{read-pr.c.}) =$

$$\lambda s, l\, (\lambda s, l, t\, \lambda y\, [x/(s \models \ll read, x^r, y^{r'}, l; t \gg)](s, l^{[l/l \circ l^{rdl}]}, 1)) =$$
$$\lambda s, l\, \lambda y\, [x/(s \models \ll read, x^r, y^{r'}, l^{[l/l \circ l^{rdl}]}; 1 \gg)].$$

Here, it is not claimed that α_τ is the appropriately inflected verb form, which would require more elaborated syntax for agreement with the subject. The rule only stipulates that α_τ is a verb with a tense marker attached to it.

4 Noun Phrases ($\mathbf{NP_1}$)

This rule represents the simplest case of quantificational NPs. This rule is generalized in Loukanova (2002, this volume). If δ is a Det, β is a N, and β does not contain any NPs, then $(\delta(\beta)^j)_i$ is a NP.

The superindex j stands for the *resource* situation[5] s_j for "evaluating" the meaning of the noun β, i.e. s_j is the situation of the domain of the quantification: the specified by $\mathcal{F}(\delta)$ quantity of objects having the property denoted by the noun β in the situation s_j. By the grammar rules, it follows that:

$\mathcal{B}(\beta) = \lambda s, l\, [x/p(x, s, l)]$ for some proposition $p(x, s, l)$ and $\mathcal{M}(\beta) = \emptyset$.

The basis and the storage of $(\delta(\beta)^j)_i$ are defined as follows:

$\mathcal{B}((\delta(\beta)^j)_i) = x_i$, and

$\mathcal{M}((\delta(\beta)^j)_i) = \{\langle \sigma, x_i \rangle\}$, where

$\sigma = \lambda s\, ((\mathcal{B}(\delta)(s))\, (\mathcal{B}(\beta)(s_j, l_j)))$.

After λ-application operation is performed:

$$\sigma = \lambda s\, (\lambda T_1 [T_2/(s \models\ll \mathcal{F}(\delta), T_1, T_2; 1 \gg)]([x/p(x, s_j, l_j)])) = \\ \lambda s\, [T_2/(s \models\ll \mathcal{F}(\delta), [x/p(x, s_j, l_j)], T_2; 1 \gg)].$$

We shall call the type σ the *basic type meaning* of the noun phrase $(\delta(\beta)^j)_i$. In the above calculation, the application $\mathcal{B}(\delta)(s)$ "inserts" the situation parameter s, i.e. the parameter for a situation which supports the quantificational infon. Then the result is applied to $\mathcal{B}(\beta)(s_j, l_j) = [x/p(x, s_j, l_j)]$, which fills up the *QDomain*-role of the relation $\mathcal{F}(\delta)$. The index i represents the antecedent-anaphora relation between $(\delta(\beta)^j)_i$ and some other NPs, which might occur in a broader phrase. The intuition and the need of the second component x_i in the type meaning $\langle \sigma, x_i \rangle$ saved in the storage together with the quantifier σ will become more transparent later. The quantifier type σ has a binding force over the parameter x_i, that occurs free in the basis. The index i also plays important role in the restrictions over the scoping order.

Example 3. $\mathcal{B}((\text{A UNICORN}^j)_i) = x_i$,

$\mathcal{M}((\text{A UNICORN}^j)_i) = \{\langle \sigma, x_i \rangle\}$, where

$\sigma = \lambda s\, [T/(s \models\ll a,\ [x/(s_j \models\ll \mathit{unicorn}, x, l_j; 1 \gg)], T; 1 \gg)]$.

Note that s_j and l_j are (free) parameters in σ. Some appropriate values, though again parametric, can be assigned to them by the speaker's references in a particular context.

[5] For a discussion of the notion of a resource situation, see Barwise and Perry (1983).

5 Simple Verb Phrases ($\mathbf{VP_0}$)

If α is an intransitive tensed verb IV_t, then it is a verb phrase VP. The storage and the basis are the same. (In a more elaborated grammar this rule can be more complicate.)

6 Extensioanlity vs. Intensionality of the Verbs

If α_τ is a TV_t and β is a NP, then $\alpha_\tau\beta$ is a VP (β might be indexed or not).

The expression $\alpha_\tau\beta$ might be semantically inconsistent, although generally accepted as syntactically well-formed structure, as for example, $[$READS A DRESS$]_{NP}$. By careful selection of the appropriateness conditions of the argument roles such semantical inconsistency can be escaped. Generally, We are suggesting a type-driven approach to linguistic meanings and interpretations. The notion of "type-driven interpretation" was first introduced by Klein and Sag (1985). In this grammar, we shall not go into more details about specifying the appropriateness conditions, with the important exception of the suggestion about using them for distinguishing the extensional and intensional verbs. The basis $\mathcal{B}(\alpha_\tau\beta)$ and the storage $\mathcal{M}(\alpha_\tau\beta)$ shall be defined by considering two cases depending on the class to which the verb α belongs, i.e. depending on the appropriateness conditions over the argument roles. The *Obj*-role of an extensional verb has to be filled by an individual (or an individual parameter), while that of an intensional verb, can be alternatively filled either by a type, or by an individual. The appropriateness conditions of the argument roles in $\mathcal{ARG}(\alpha)$ will explicitly specify which is the case. They may include some more restrictions to constrain semantically the combinations of lexical units. Thus, for an intensional verb α, like SEEK, the appropriateness conditions of the *Obj*-role permit two options. The type of the *Obj*-role filler might be either the individual type *IND*, or the type of types of individuals *TTI*:

(6.0) $\mathcal{ARG}(seek) = \{< Subj, r >, < Obj, (IND \vee TTI) >, < Loc, SPT >\}$,

Before considering further the verb SEEK and its meanings, we need to define $\mathcal{B}(\alpha_\tau\beta)$ and $\mathcal{M}(\alpha_\tau\beta)$ for the two kinds of verbs — intensional and extensional.

6.1 Option1 for the Transitive Verbs ($\mathbf{VP_1}$)

Let α_τ be a TV_t such that $Appr(Obj) \sqsubseteq IND$[6], i.e. the *Obj*-role can be filled up by an individual (or an individual parameter). Let β be a NP such that $\mathcal{B}(\beta)$ is either an individual, or an individual parameter, which respects all *Obj* conditions. Then $\alpha_\tau\beta$ is a VP and:

$$\mathcal{B}(\alpha_\tau\beta) = \lambda s, l\,((\mathcal{B}(\alpha_\tau)(s,l))\,\mathcal{B}(\beta)),$$

$$\mathcal{M}(\alpha_\tau\beta) = \mathcal{M}(\beta).$$

[6] The relation $\sqsubseteq$ represents the notion of informational subsumption: $F_1 \sqsubseteq F_2$ iff the information available in F_1 is also available in F_2.

Example 4. $\mathcal{ARG}(meet) = \{< Subj, IND >, < Obj, IND >\}$, and let alternative (Alt.1) be taken for the pronoun $\textsc{you}$, then

$$\mathcal{B}(\textsc{met you}){=}\ \lambda s, l\, (\lambda y[x/(s \models\ll meet, x, y, l^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)](x_i)) =$$

$$\lambda s, l\, [x/(s \models\ll meet, x, x_i, l^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)], \text{ and}$$

$$\mathcal{M}(\textsc{met you}) = \{\langle \lambda s\, [T/(T : x_i^{rlst})], x_i\rangle\}.$$

If we use alternative (Alt.2) for $\textsc{you}$, then

$$\mathcal{B}\ (\textsc{met you}){=}\ \lambda s, l\, [x/(s \models\ll meet, x, x_i^{rlst}, l^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)], \text{ and}$$

$$\mathcal{M}(\textsc{met you}) = \emptyset.$$

6.2 Option2 for the Transitive Verbs (VP$_2$)

This rule "inserts" types into the object role of the transitive verbs.

Let α_τ be a TV$_t$, such that $Appr(Obj) \sqsubseteq TTI$ i.e. the appropriateness conditions of its *Obj*-role permit it to be filled by an object that is a type of types of individuals. Let β_i be an indexed[7] NP, then $\alpha_\tau\beta_i$ is a VP. Here we are formulating the rule for generating the basis and storage of $\alpha_\tau\beta_i$, for a special case of simple VPs the head, of which is an intensional verb. (Two generalized versions of this rule shall be given in Loukanova (2002, this volume). Let

$$\mathcal{M}(\beta_i) = \{\langle\sigma, x_i\rangle\} \text{ and } \mathcal{B}(\beta_i) = x_i. \text{ Then}$$

$$\mathcal{B}(\alpha_\tau\beta_i) = \lambda s, l\, (\mathcal{B}(\alpha_\tau)(s, l)\sigma(s_q)), \text{ and } \mathcal{M}(\alpha_\tau\beta) = \emptyset.$$

If σ is a quantitificational type (which is the case for all indexed NPs in this grammar), the situation s_q is the quantificational situation supporting the quantificational infon. By this rule, σ, which is the type meaning of the quantifier β_i, is taken out of the storage and inserted into the basis to fill up the *Obj*-role of the relation denoted by α.

Example 5. Let us consider the verb phrase $\textsc{is seeking}\ (\textsc{a unicorn}^j)_i$, where the base and the storage of $(\textsc{a unicorn}^j)_i$ are as in Example 4.1. The appropriateness conditions given in (6.0) and the rule TV$_t$ yield:

$$\mathcal{B}(\textsc{is seeking}){=}\ \lambda s, l\, \lambda Y[x/(s \models\ll seek, x^r, Y^{(IND \vee TTI)}, l^{[l/l\, \circ\, l^{rdl}]}; 1 \gg)],$$

$$\mathcal{M}(\textsc{is seeking}){=}\ \emptyset.$$

1. *De re* (specific) reading of $\textsc{is seeking a}\ (\textsc{unicorn}^j)_i$ is a result of applying VP$_1$:

   $\mathcal{B}(\textsc{is seeking}\ (\textsc{a unicorn}^j)_i) =$

[7] We do not consider this case for nonindexed NPs as it would be trivial for both alternatives of the pronouns $\textsc{i}$ and $\textsc{you}$, which are the only nonindexed NPs in this paper.

$\lambda s, l\, (\mathcal{B}(\textsc{is seeking})(s,l)(\mathcal{B}((\textsc{a unicorn}^j)_i))) =$

$\lambda s, l\, (\lambda Y[x/(s \models\ll \mathit{seek}, x^r, Y^{(IND \vee TTI)}, l^{[l/l \circ l^{rdl}]}; 1 \gg)](x_i)) =$

$\lambda s, l\, [x/(s \models\ll \mathit{seek}, x^r, x_i^{IND}, l^{[l/l \circ l^{rdl}]}; 1 \gg)].$

$\mathcal{M}(\textsc{is seeking}\ (\textsc{a unicorn}^j)_i) = \{\langle \sigma, x_i \rangle\}.$

The linguistic meaning of $\textsc{is seeking}\ (\textsc{a unicorn}^j)_i$ is the pair of the storage and the basis:

$[\![\textsc{is seeking}\ (\textsc{a unicorn}^j)_i]\!] =$

$\langle \{\langle \lambda s\, [T/(s \models\ll a,\ [x/(s_j \models\ll \mathit{unicorn}, x, l_j; 1 \gg)], T; 1 \gg)], x_i \rangle\},$
$\lambda s, l\, [x/(s \models\ll \mathit{seek}, x^r, x_i^{IND}, l^{[l/l \circ l^{rdl}]}; 1 \gg)] \rangle.$

The quantifier type meaning σ saved in the storage as the pair $\langle \sigma, x_i \rangle$ has a binding force over the parameter x_i, that occurs free in the basis. The above meaning pair has a nice intuitive interpretation: the type meaning σ from the storage expresses that an indeterminate representative (instance) of the domain of the determiner a is an object of type $[x/(s_j \models\ll \mathit{unicorn}, x, l_j; 1 \gg)]$. By the meaning constraint (C_a) for a introduced later in the paper, there is at least one such representative. The range (still not complete) is expressed by the basis. The index i in the quantificational pair $\langle \sigma, x_i \rangle$ is the link between the storage (domain of the quantification) and the basis (range of the quantification), i.e. i indexes the argument role of an abstraction over x_i in the basis. The basis is still incomplete predication as the expression considered is a verb phrase which needs a subject to be applied to:

2. *De dicto* (non-specific) reading of $\textsc{is seeking a}\ (\textsc{unicorn}^j)_i$ is a result of applying Option2:

$\mathcal{B}(\textsc{is seeking}\ (\textsc{a}\ (\textsc{unicorn}^j)_i)) =$

$\lambda s, l\, ((\lambda Y[x/(s \models\ll \mathit{seek}, x^r, Y^{(IND \vee TTI)}, l^{[l/l \circ l^{rdl}]}; 1 \gg)])\sigma(s_q)) =$

$\lambda s, l\, [x/(s \models\ll \mathit{seek},\ x^r,$
$[T/(s_q \models\ll a, [x/(s_j \models\ll \mathit{unicorn}, x, l_j; 1 \gg)], T; 1 \gg)],$
$l^{[l/l \circ l^{rdl}]}; 1 \gg)]$, and

$\mathcal{M}(\textsc{is seeking}\ (\textsc{a unicorn}^j)_i) = \emptyset.$

7 Sentences ($\mathbf{S_1}$)

If α_i is a NP and β is a VP, then $\alpha_i\beta$ is a sentence. (The cases where the subject NP is not indexed are trivial cases of everything that follows.)

By the rules of the grammar, as stated in this paper, and in Loukanova (2002, this volume), if the subject NP α_i is a quantificational NP, then its type meaning is in the storage, and $\mathcal{B}(\alpha_i) = x_i$. If (Alt.2) is taken for the individual terms, then $\mathcal{B}(\alpha_i) = x_i^r$ for some restriction r and the storage is empty. There are various other reasons (some of which become clear in the extended grammar in Loukanova (2002, this volume) besides reducing complexity of the calculations, for taking the second alternative. In any of these cases, if β is a VP, then $\mathcal{B}(\beta) = \lambda s, l\,[x/p(x, s, l)]$ for some proposition $p(x, s, l)$. Generally, the starages of β and α_i may contain more than one pair of quantificational type meaning and its corresponding indexed parameter. These are basic type meanings of NPs that occur in α_i and β. Some might already have been quantified into the $\mathcal{B}(\beta)$, or into $\mathcal{BMT}(\alpha_i)$. As before let

$$\begin{aligned}\mathcal{M}_0(\alpha_i\beta) = {} & \mathcal{M}'(\alpha_i) \bigcup \mathcal{M}'(\beta) \bigcup \\ & \{\langle \lambda s \mathcal{R}(\sigma(s), \tau(s)), x_j\rangle / \text{ where } j,\ \sigma \text{ and } \tau \text{ are such} \\ & \text{that } \sigma \neq \tau,\ \langle \sigma, x_j\rangle \in \mathcal{M}(\alpha_i) \text{ and } \langle \tau, x_j\rangle \in \mathcal{M}(\beta)\},\end{aligned}$$

where $\mathcal{M}'(\alpha_i)$ and $\mathcal{M}'(\beta)$ are the largest sets such that $\mathcal{M}'(\alpha_i) \subseteq \mathcal{M}_0(\alpha_i)$, $\mathcal{M}'(\beta) \subseteq \mathcal{M}(\beta)$, and for which there are no σ, τ, and j such that $\sigma \neq \tau$, $\langle \sigma, x_j\rangle \in \mathcal{M}'(\alpha_i)$ and $\langle \tau, x_j\rangle \in \mathcal{M}'(\beta)$. The basis and the storage of $\alpha_i\beta$ are defined as follows:

$$\mathcal{B}(\alpha_i\beta) = \mathcal{P}_c(\mathcal{B}(\alpha_i), \mathcal{B}(\beta), \mathcal{M}_0(\alpha_i\beta)) =$$

$$\begin{aligned}&[s_d, l_d/\ (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ &\qquad (\sigma_k(s_{q_k}) : [x_{i_k}/(\mathcal{B}(\beta)(s_d, l_d) : \mathcal{B}(\alpha_i))])\ldots])], \text{ where}\end{aligned}$$

$k \geq 0$ and $\{\langle \sigma_1, x_{i_1}\rangle, \ldots, \langle \sigma_k, x_{i_k}\rangle, \} \subseteq \mathcal{M}(\alpha)$, and

$$\mathcal{M}(\alpha_i\beta) = \mathcal{M}_0(\alpha_i\beta) - \{\langle \sigma_1, x_{i_1}\rangle, \ldots, \langle \sigma_k, x_{i_k}\rangle\}.$$

After appropriate substitution we get:

$$\begin{aligned}\mathcal{B}(\alpha_i\beta) = [s_d, l_d/\ & (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ & (\sigma_k(s_{q_k}) : [x_{i_k}/p(x_i, s_d, l_d)])\ldots])].\end{aligned}$$

It should be remembered that the substitutions in the proposition $p(x_i, s_d, l_d)$ must "inherit" all available restrictions over the parameters. The index i might be equal to one of the indices $i_1, \ldots, i_k$, in which case it gets bound. Otherwise its corresponding quantifier is still in the storage $\mathcal{M}(\alpha_i\beta)$.

The operator $\mathcal{P}_c$ is context dependent and must respect the quantificational restriction (QR) formulated in Loukanova (2002, this volume). In the above definition we have different situation parameters for the situations that support the quantifications and for the situation of the main verb phrase predication. In many cases, these might be same as the described situation. The above sentence rule preserves the basic sentence predication to be $(\mathcal{B}(\beta)(s_d, l_d) : \mathcal{B}(\alpha_i))$, i.e. the VP property is predicated for the individual representatives of the NP.

8 Sentences ($\mathbf{S_2}$)

This rule does not correspond to any syntactical changes in a sentence α, rather it changes its semantical representation. If the linguistic meaning $[\![\alpha]\!] = \langle \mathcal{M}(\alpha), \mathcal{B}(\alpha)\rangle$

of a sentence α is such that $\mathcal{M}(\alpha) \neq \emptyset$, it is possible some of the type meanings in the storage to be moved into the basis after getting additional information which resolves the corresponding quantificational ambiguity. More likely this rule belongs to the level of getting an interpretation of the sentence in a particular context of use.

The new basis and the storage are defined as follows.

$$\begin{aligned}\mathcal{B}'(\alpha) = \mathcal{P}_c(\mathcal{B}(\alpha), \mathcal{M}(\alpha)) = [s_d, l_d/\ (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ (\sigma_k(s_{q_k}) : [x_{i_k}/\mathcal{B}(\alpha)])(s_d, l_d) \ldots])], \text{ where}\end{aligned}$$

$k > 0$, and $\{\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle, \} \subseteq \mathcal{M}(\alpha)$.

$\mathcal{M}'(\alpha) = \mathcal{M}(\alpha) - \{\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle\}$.

As in the previous rule, the operator $\mathcal{P}_c$ must respect the quantificational restriction (QR).

Example 6. $\gamma =$ (EVERY LOGICIAN^3)$_5$ MET (A PHILOSOPHER^1)$_2$

Without using VP$_3$ we have as before:

$\mathcal{B}(\text{MET (A PHILOSOPHER}^1)_2) = \lambda s, l\, [x/(s \models\ll \mathit{meet}, x, x_2; l^{[l/l \prec l^{rdl}]} \gg)]$

$\mathcal{M}(\text{MET (A PHILOSOPHER}^1)_2) = \mathcal{M}((\text{A PHILOSOPHER}^1)_2) = \{\langle \sigma, x_2 \rangle\}$, where

$\sigma = \lambda s\, [T/(s \models\ll a, [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], T; 1 \gg)]$.

$\mathcal{B}((\text{EVERY LOGICIAN}^3)_5) = x_5$,

$\mathcal{M}((\text{EVERY LOGICIAN}^3)_5) = \{\langle \sigma_5, x_5 \rangle\}$, where

$\sigma_5 = \lambda s\, [T/(s \models\ll \mathit{every}, [x/(s_3 \models\ll \mathit{logician}, x, l_3; 1 \gg)], T; 1 \gg)]$.

Now the rule S$_1$ can be applied in five different ways. Two of them, with $k = 2$, result in an empty storage. One of these two gives a specific reading, while the other one is for a nonspecific reading. Let, for example consider the nonspecific one.

$$\begin{aligned}\mathcal{B}(\gamma) = [s_d, l_d/(\sigma_5(s_{q_5}) : [x_5/(\sigma(s_{q_2}) : [x_2/ \\ (s_d \models\ll \mathit{meet}, x_5, x_2, l^{[l/l \prec l^{rdl}]}; 1 \gg)])])] =\end{aligned}$$

$$\begin{aligned}[s_d, l_d/(s_{q_5} \models\ll \mathit{every},\ [x/(s_3 \models\ll \mathit{logician}, x, l_3; 1 \gg)], \\ [x_5/(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\ [x_2/(s_d \models\ll \mathit{meet}, x_5, x_2, l^{[l/l \prec l^{rdl}]}; \\ 1 \gg)]; \\ 1 \gg)]; 1 \gg)].\end{aligned}$$

9 Conclusion

The inherent ambiguity and vagueness of NLs is a tough obstacle for any theory of linguistic meaning. There are various kinds of ambiguities neither of which is easy for effective algorithmization. A theory, which choses to generate all alternative "readings" is inevitably of exponential complexity. As pointed by Hobbs and Shieber (1987), the number of the scope readings of a sentence with 5 NPs is between 16 and 120.

The current approach for semantics assumes that there is only one syntactical representation of a sentence with scope ambiguity. The linguistic meaning of an ambiguous quantificational expression represents the alternatives without generating the various readings with resolved scopes. The particular readings are calculated when disambiguating information is supplied either by some linguistic constraints (lexical, syntactical, semantical), or at the moment of semantical interpretation in a particular context of use, by the speaker's references and intentions. Non-linguistic information, such as knowledge of natural lows also contributes to reducing the ambiguity, and can be taken into account by an elaborated implementation of the semantical rules of the grammar.

Acknowledgments

I would like to thank Alexander Gelbukh for the crucial assistance and support in writing and organizing the grammar presented in this paper. I would like to include a special thanks to Arnold Insel for spending many hours with me on discussions of the problems covered by the grammar, patiently reading my writings, and for the Latex lessons.

References

1. Barwise, Jon and John Perry. 1983. *Situations and Attitudes.* Cambridge, MA:MIT press. Republished in 1999 by The David Hume Series of Philosophy and Cognitive Science Reissues.
2. Barwise, Jon. 1987. *The situation in Logic.* CSLI Lecture Notes Number 17. Stanford: CSLI Publications.
3. Cooper, Robin. 1992. *A Working Person's Guide to Situation Theory.* in Topics in Semantic Interpretation, ed. by Steffen Leo Hansen and Finn Soerensen, Samfundslitteratur, Frederiksberg, Denmark
4. Dowty, David, Robert Wall and Stanley Peters. 1981. *Introduction to Montague Semantics.* Studies in Linguistics and Philosophy. Volume 11. Kluwer Academic Publishers. Dordrecht/Boston/London.
5. Ginzburg Jonathan and Ivan A. Sag. 2001. *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives.* Stanford: CSLI Publications. Distributed by the University of Chicago Press.
6. Hobbs, Jerry and Stuart Shieber. An Algorithm for Generating Quantifier Scopings. Computational Linguistics, volume 13, numbers 1-2, pages 47-63, 1987.
7. Klein, Ewan and Ivan Sag *Type-Driven Translation.* Linguistics and Philosophy, 8 (1985). p. 163-201.

8. Loukanova, R. 2000. *Nominal Scope in Situation Semantics.* CICLing-2000 Proceedings: Computational linguistics and Intelligent Text Processing by IPN Publishing House. Mexico. [A shorter version is reprinted in the Proceedings of 14th Pacific Asia Conference on Language, Information and Computation (PACLIC14), 2000 Waseda University International Conference Center, Tokyo, Japan. Proceedings, 243-252 p. Ikeya A. and M. Kawamori (eds.) Logico-Linguistic Society of Japan, The Institute of Language Teaching, Waseda University Media Network Center.]
9. Loukanova, R. 2002. *Generalized Quantification in Situation Semantics.* in this volume.
10. Seligman, Jerry and Lawrence S. Moss. 1997. *Situation Theory.* in Van Benthem, J. and A. ter Meulen, eds. *Handbook of Logic and Language.* Cambridge, MA: MIT Press.

Generalized Quantification in Situation Semantics

Roussanka Loukanova

Illinois Wesleyan University,
Department of Mathematics and Computer Science,
P.O. Box 2900, Bloomington, IL 61702-2900, USA
rloukano@iwu.edu

Abstract. This paper extends the grammar presented in Loukanova (2002, this volume) to cover NL expressions that contain multiple quantificational NPs, restrictive relative clauses, and intensional verbs. The grammar rules use a quantificational operator, which moves quantifier representations from the storage to the basis. This operator is highly context dependent and is a subject of structural restrictions, introduced in this paper. These restrictions do not permit free parameters to fall out of the scope of the quantificational binding. The grammar permits more than one NPs, different from pronouns, to be in antecedent-anaphora relations. The relevant quantificational rules use a two argument operator $\mathcal{R}$, defined over types of individuals, which combines the meaning types corresponding to the determiners A, SOME, THE.

1 Introduction

Before generalizing the rule for the transitive verbs given in the grammar by Loukanova, (2002, this volume), let us see some data behind it. A quantificational NP β_i may contain other quantificational NPs as constituents, for example, let $\beta_i =$
(A PHILOSOPHER WHO WANTS (TWO BOOKS WRITTEN BY (A LOGICIAN)$_5$)$_3$)$_i$.

In absence of enough scope resolving information, all possibilities should be left open. Rather than generating all scope readings, which should be avoided in order to achieve computational effectiveness, the grammar rules shall be defined in such way that the basis and the storage of $\beta_i =$ are generated as given below. A sentence, in which $\beta_i =$ occurs as a component NP inherits the storage $\mathcal{M}(\beta_i)$. In the prcess of generating the sentence, if some scope resolving linguistic, or extra-linguistic information becomes available, some or all of the type meanings in the storage can be quantified into the basis.

$\mathcal{B}(\beta_i) = x_i$

$\mathcal{M}(\beta_i) = \{\langle \sigma_3, x_3 \rangle, \langle \sigma_5, x_5 \rangle, \langle \sigma_i, x_i \rangle\}$, where

$$\sigma_5 = \lambda s[T/(s \models \ll a, \quad [x/(s_5 \models \ll \mathit{logician}, x, l_5; 1 \gg)], \\ T; 1 \gg)]$$

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 46–57, 2002.

$$\begin{aligned}\sigma_3 = \lambda s[T/(s \models\ll \mathit{two},\ & [x/(s_3 \models\ll \mathit{book\_written\_by}, x, x_5, l_3; 1 \gg)],\\ & T; 1 \gg)]\\ \sigma_i = \lambda s[T/(s \models\ll a,\ & [x/(s_3 \models\ll \mathit{philosopher\_who\_wants}, x, x_3, l_3; 1 \gg)],\\ & T; 1 \gg)]\end{aligned}$$

The *type meanings* σ_3 and σ_5 can be kept in the storage $\mathcal{M}(\beta_i)$ outside of the basic type meaning σ_i. This means that up to this moment the quantificational scopes, i.e. the domains and the ranges of the component NPs have not been resolved. In this case, the type meaning σ_i has as a constituent the (free) parameter x_3 filling the relevant argument role instead of σ_3, and σ_3 has as a constituent the (free) parameter x_5 instead of σ_5. The NP β_i = can be used as an object argument of a VP having an intensional head verb:

$[$IS SEEKING (A PHILOSOPHER WHO WANTS (TWO BOOKS WRITTEN BY (A LOGICIAN$)_5)_3)_i]_{VP}$

A generalized version of Option2 of the (VP$_2$) rule of the grammar in Loukanova (2002 this volume) need to be defined in such way that σ_i can be moved to the basis to fill the *Obj*-role of the transitive intensional verb α_τ. Also it has to permit some of σ_3 and σ_5 to be quantified into, so that σ_i to be in their ranges[1]. Let now state the general quantificational rule for calculating the basis and the storage of a VP $\alpha_\tau\beta_i$.

Quantificational VP Rule (VP$_2$):

Let α_τ be a TV$_t$, such that $\mathit{Appr}(\mathit{Obj}) \sqsubseteq \mathit{TTI}$ i.e. the appropriateness conditions of its *Obj*-role permit it to be filled by an object that is a type of types of individuals. Let β_i be an indexed NP, then $\alpha_\tau\beta_i$ is a VP.

Let $\mathcal{B}(\beta_i) = x_i$, and the types $\sigma_1, \ldots, \sigma_k, \sigma$ and the indices $i_1, \ldots, i_k$ be such that $\{\langle\sigma_1, x_{i_1}\rangle, \ldots, \langle\sigma_k, x_{i_k}\rangle, \langle\sigma, x_i\rangle\} \subseteq \mathcal{M}(\beta_i)$, where $k \geq 0$. The type σ which is the first element of that pair in $\mathcal{M}(\beta_i)$, the second element of which is x_i, is called *basic type meaning* of β_i and is denoted by $\mathcal{BMT}(\beta_i)$. The types $\sigma_1, \ldots, \sigma_k$ are the basic type meanings of some of the constituent NPs in β_i that have been kept in the storage $\mathcal{M}(\beta_i)$. The general quantificational (NP) rule given later in this paper will state how the basic meaning types of the NPs are generated. Then we define:

$\mathcal{B}(\alpha_\tau\beta_i) = \mathcal{P}_c(\mathcal{B}(\alpha_\tau), \mathcal{B}(\beta_i), \mathcal{M}(\beta_i)) =$

$$\begin{aligned}\lambda s, l\, (\mathcal{B}(\alpha_\tau)(s,l)\, [T/\ & (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots\\ & \qquad (\sigma_k(s_{q_k}) : [x_{i_k}/(\sigma(s_q) : T)]) \ldots])])\end{aligned}$$

$$\mathcal{M}(\alpha_\tau\beta_i) = \mathcal{M}(\beta_i) - \{\langle\sigma_1, x_{i_1}\rangle, \ldots, \langle\sigma_k, x_{i_k}\rangle, \langle\sigma, x_i\rangle\}.$$

The parameters $s_{q_1}, \ldots, s_{q_k}, s_q$ are for the situations supporting the quantificational infons of the quantifiers $\sigma_1, \ldots, \sigma_k, \sigma$, respectively. They might be the

[1] The restriction (QR) formulated later does not permit the vise versa in this rule. But it is possible σ_3, or σ_3 and σ_5 to have already been inserted into σ_i by some other rules of the grammar.

same as the situation s of the main verb predication, but there is the possibility they to be different.

The operator $\mathcal{P}_c$ is the device that selects and orders the quantificational pairs $\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle, \langle \sigma, x_i \rangle$, and binds the parameters $x_{i_1}, \ldots, x_{i_k}$ in that particular order. It is is highly context dependent, and generally may be applied in a particular discourse. Some particular lexical combinations may impose one or other order of the quantification, i.e. the operator $\mathcal{P}_c$ may be applied also at the level of the calculation of the linguistic meaning of an expression out of any particular context of use, see Farkas (1997a, b, c). That is the reason, all quantificational rules in this grammar to be defined in such way, that several quantifiers at once to be inserted into the basis, which might be invoked after putting together some expressions. There is though, a general structural restriction (QR) that $\mathcal{P}_c(\alpha_\tau \beta_i)$ must respect:

(QR): Quantificational Restriction

(1) In the quantifier order: $\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle, \langle \sigma_{k+1}, x_{i_{k+1}} \rangle$, where $\sigma_{k+1} = \sigma$ and $i_{k+1} = i$, there must be no $m, n \in \{1, \ldots, k+1\}$ such that $m \leq n$ and x_{i_n} is a (free) parameter in σ_m.

(2) $x_{i_1}, \ldots, x_{i_k}$ are not (free) parameters of the type meanings left in the new storage $\mathcal{M}(\alpha_\tau \beta_i)$.

2 Restrictive Relative Nouns ($\mathbf{N}_{\boldsymbol{R}}$)

This rule is giving the traditional "conjunctive" treatment of the relative clauses.

If α is a N and β is a VP, then (α THAT β) is a N. By the rules of the grammar it follows that $\mathcal{B}(\alpha)$ and $\mathcal{B}(\beta)$ are objects of the following kind:

$\mathcal{B}(\alpha) = \lambda s, l\, [x/p_1(s, l, x)]$, and

$\mathcal{B}(\beta) = \lambda s, l\, [x/p_2(s, l, x)]$, for some propositions p_1 and p_2.

It is possible both of the storages $\mathcal{M}(\alpha)$ and $\mathcal{M}(\beta)$ to be nonempty sets. Something more, it is possible they to have as elements pairs, the second components of which are same, i.e. $\langle \sigma, x_j \rangle \in \mathcal{M}(\alpha)$ and $\langle \tau, x_j \rangle \in \mathcal{M}(\beta)$ for some index j and some types σ and τ such that $\sigma \neq \tau$. That might happen when different NPs with same index occur in both expressions α and β, and by this are in an antecedent-anaphora relation, as for example:

[[WOMAN WITH (A RED HAT)$_j$]$_N$ WHO LATER TOOK OFF (THIS UGLY RAG)$_j$]$_N$

[[LOGICIAN WHO MET (THE PHILOSOPHER)$_j$]$_N$ AND WHO DISLIKED (this liar)$_j$]$_N$

The two storages $\mathcal{M}(\alpha)$ and $\mathcal{M}(\beta)$ has to be joined in such way that the pairs with same indices like $\langle \sigma, x_j \rangle$ and $\langle \tau, x_j \rangle$ to be combined into a new pair with one common quantificational type containing all information available in σ and τ. For this purpose I shall define an operation $\mathcal{R}$ over types of types of individuals so that the value $\mathcal{R}(\sigma(s), \tau(s))$ would be the combined quantificational type. The definition of $\mathcal{R}$ for some determiners shall be given at the end of the paper.

Quantificational Rule ($\mathbf{N}_R$):

Let $\mathcal{M}'(\alpha_i)$ and $\mathcal{M}'(\beta)$ be the largest sets such that $\mathcal{M}'(\alpha_i) \subseteq \mathcal{M}(\alpha_i)$, $\mathcal{M}'(\beta) \subseteq \mathcal{M}(\beta)$, and for which there are no σ, τ, and j such that $\sigma \neq \tau$, $\langle \sigma, x_j \rangle \in \mathcal{M}'(\alpha_i)$ and $\langle \tau, x_j \rangle \in \mathcal{M}'(\beta)$. I.e., $\mathcal{M}'(\alpha)$ and $\mathcal{M}'(\beta)$ are received from $\mathcal{M}(\alpha)$ and $\mathcal{M}(\beta)$, respectively, by taking out the pairs with the same index j. Let

$$\begin{aligned}\mathcal{M}_0(\alpha \text{ } \textsc{that} \text{ } \beta) = \mathcal{M}'(\alpha) \bigcup \mathcal{M}'(\beta) \bigcup \\ \{ \langle \lambda s \mathcal{R}(\sigma(s), \tau(s)), x_j \rangle / \text{ where } j, \sigma \text{ and } \tau \text{ are such} \\ \text{that } \sigma \neq \tau, \ \langle \sigma, x_j \rangle \in \mathcal{M}(\alpha) \text{ and } \langle \tau, x_j \rangle \in \mathcal{M}(\beta) \}.\end{aligned}$$

The basis and the storage of (α THAT β) are defined as follows:

$\mathcal{B}(\alpha \text{ } \textsc{that} \text{ } \beta) = \mathcal{P}_c(\mathcal{B}(\alpha), \mathcal{B}(\beta), \mathcal{M}_0(\alpha \text{ } \textsc{that} \text{ } \beta)) =$

$$\begin{aligned}&\lambda s_1, l_1 \, [x/ \ (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ &\qquad (\sigma_k(s_{q_k}) : [x_{i_k}/(\mathcal{B}(\alpha)(s_1, l_1)(x) \wedge \mathcal{B}(\beta)(s_2, l_2)(x))]) \ldots])],\end{aligned}$$

where $k \geq 0$, and $\{\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle, \} \subseteq \mathcal{M}_0(\alpha \text{ } \textsc{that} \text{ } \beta)$.

$\mathcal{M}(\alpha \text{ } \textsc{that} \text{ } \beta) = \mathcal{M}_0(\alpha \text{ } \textsc{that} \text{ } \beta) - \{\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle, \}$.

As in the previous quantificational rule, the operator $\mathcal{P}_c$ selects the pairs $\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle$, orders them, and binds the parameters $x_{i_1}, \ldots, x_{i_k}$ in that particular order. It has to respect the quantificational restriction (QR) stated here again, with a notational change. Till the end of the work we shall refer to it as it is stated here.

(QR): Quantificational Restriction

(1) In the quantifier order: $\langle \sigma_1, x_{i_1} \rangle, \ldots, \langle \sigma_k, x_{i_k} \rangle$, there must be no $m, n \in \{1, \ldots, k\}$ such that $m \leq n$ and x_{i_n} is a (free) parameter in σ_m.

(2) $x_{i_1}, \ldots, x_{i_k}$ are not (free) parameters of the type meanings left in the new storage $\mathcal{M}(\alpha \text{ } \textsc{that} \text{ } \beta)$.

We assume that when $k = 0$ the above rule for the basis is:

$\mathcal{B}(\alpha \text{ } \textsc{that} \text{ } \beta) = \lambda s_1, l_1 \, [x/(\mathcal{B}(\alpha)(s_1, l_1)(x) \wedge \mathcal{B}(\beta)(s_2, l_2)(x))]$.

Example 1. $\mathcal{B}(\textsc{driver}) = \lambda s, l \, [x/(s \models \ll \mathit{driver}, x, l; 1 \gg)]$,
$\mathcal{B}(\textsc{runs}) = \lambda s, l \, [x/(s \models \ll \mathit{run}, x, l^{[l/l \circ l^{rdl}]}; 1 \gg)]$,

$$\begin{aligned}&\mathcal{B}(\textsc{driver that runs}) = \\ &\lambda s_1, l_1 \, [x/(\ \lambda s, l \, [x/(s \models \ll \mathit{driver}, x, l; 1 \gg)](s_1, l_1) \wedge \\ &\qquad \lambda s, l \, [x/(s \models \ll \mathit{run}, x, l^{[l/l \circ l^{rdl}]}; 1 \gg)](s_2, l_2))] =\end{aligned}$$

$$\begin{aligned}&\lambda s_1, l_1 \, [x/ \ (s_1 \models \ll \mathit{drive}, x, l_1; 1 \gg) \wedge \\ &\qquad (s_2 \models \ll \mathit{run}, x, l_2^{[\![-s]\!]}; 1 \gg)],\end{aligned}$$

Example 2. Let (Alt.1) be taken for the pronoun YOU. Then, if we apply the rule (N_R) with $k = 1$ we get:

$\mathcal{B}(\text{LOGICIAN THAT MET YOU}) =$

$$\begin{aligned}
&\lambda s_1, l_1\, [x/([T/\ (T : x_i^{rlst})] : \\
&\qquad\qquad [x_i/\ (s_1 \models\ll \mathit{logician}, x, l_1; 1 \gg)\wedge \\
&\qquad\qquad\quad (s_2 \models\ll \mathit{meet}, x, x_i, l_2^{[l/l\,\prec\, l^{rdl}]}; 1 \gg)])] =
\end{aligned}$$

$$\begin{aligned}
&\lambda s_1, l_1\, [x/([x_i/\ (s_1 \models\ll \mathit{logician}, x, l_1; 1 \gg)\wedge \\
&\qquad\qquad (s_2 \models\ll \mathit{meet}, x, x_i, l_2^{[l/l\,\prec\, l^{rdl}]}; 1 \gg)] : x_i^{rlst})] =
\end{aligned}$$

$$\begin{aligned}
&\lambda s_1, l_1\, [x/\ (s_1 \models\ll \mathit{logician}, x, l_1; 1 \gg)\wedge \\
&\qquad\quad (s_2 \models\ll \mathit{meet}, x, x_i^{rlst}, l_2^{[l/l\,\prec\, l^{rdl}]}; 1 \gg)].
\end{aligned}$$

The last substitution violates the restriction (QR) because x_i is a (free) parameter of the type that is quantified into. This problem gets solved either by specifying the types of the individual terms as an exception in the restriction (QR), or by taking the alternative Alt.2 for the pronouns. The second choice seems more natural solution.

3 Quantificational Noun Phrases (NP)

If δ is a Det, β is a N, and i and j are natural numbers such that β does not contain any NPs indexed with i, then $(\delta(\beta)^j)_i$ is a NP. By this rule we are moving the semantical quantificational structure of $(\delta(\beta)^j)_i$ into the storage and leaving a parametric representative x_i to be the basis. By using the operator $\mathcal{R}$ defined at the end of the paper, it can be generalized for permitting cases when the noun β contains NPs indexed with i, as for example, THE BOY THAT HURT HIMSELF. The basis of $(\delta(\beta)^j)_i$ is defined to be:

$\mathcal{B}((\delta(\beta)^j)_i) = x_i$.

Before the definition of the storage of $(\delta(\beta)^j)_i$ we have to define the rule for calculating the type called *basic meaning type* of $(\delta(\beta)^j)_i$, and denoted by $\mathcal{BMT}((\delta(\beta)^j)_i)$:

$$\mathcal{BMT}((\delta(\beta)^j)_i) = \mathcal{P}_c(\mathcal{B}(\delta), \mathcal{B}(\beta), \mathcal{M}(\beta)) =$$

$$\begin{aligned}
&\lambda s\, [T/\ (\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\
&\qquad\quad (\sigma_k(s_{q_k}) : [x_{i_k}/(\mathcal{B}(\delta)(s)(\mathcal{B}(\beta)(s_j, l_j)))(T)]) \ldots])],
\end{aligned}$$

where $k \geq 0$, $\sigma_1, \ldots, \sigma_k$ and $x_{i_1}, \ldots, x_{i_k}$ are such that $\{\langle\sigma_1, x_{i_1}\rangle \ldots, \langle\sigma_k, x_{i_k}\rangle\} \subseteq \mathcal{M}(\beta)$. Here s_j and l_j are, correspondingly, the resource situation and the location of the domain of the quantificational relation $\mathcal{B}(\delta)$.

By the rules of the grammar, $\mathcal{B}(\beta)$ is an object: $\mathcal{B}(\beta) = \lambda s, l\, [x/p(x, s, l)]$, where $p(x, s, l)$ is a proposition. Then it follows that

$$\mathcal{BMT}((\delta(\beta)^j)_i) =$$

$$\begin{aligned}\lambda s\, [T/\ &(\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ &\qquad (\sigma_k(s_{q_k}) : [x_{i_k}/(s \models\ll \mathcal{F}(\delta), \mathcal{B}(\beta)(s_j, l_j), T; 1 \gg)]) \ldots])] =\end{aligned}$$

$$\begin{aligned}\lambda s\, [T/\ &(\sigma_1(s_{q_1}) : [x_{i_1}/ \ldots \\ &\qquad (\sigma_k(s_{q_k}) : [x_{i_k}/(s \models\ll \mathcal{F}(\delta), [x/p(x, s_j, l_j)], T; 1 \gg)]) \ldots])],\end{aligned}$$

The definition of $\mathcal{BMT}((\delta(\beta)^j)_i)$ permits some meaning types of NPs occurring in β that has been kept in the storage of β, to be quantified into the basic meaning type of $(\delta(\beta)^j)_i$. The restriction (QR) must be respected in selecting the order of the quantifications. Then the storage of $(\delta(\beta)^j)_i$ is defined as follows:

$$\begin{aligned}&\mathcal{M}((\delta(\beta)^j)_i) = \\ &(\mathcal{M}(\beta) - \{\langle \sigma_1, x_{i_1}\rangle \ldots, \langle \sigma_k, x_{i_k}\rangle\}) \bigcup \{\langle \mathcal{BMT}((\delta(\beta)^j)_i), x_i\rangle\}.\end{aligned}$$

Example 3. Let $\varphi =$ [LOGICIAN THAT MET (A PHILOSOPHER^1)$_2$]$_N$

$\mathcal{B}$((A PHILOSOPHER^1)$_2$)$= x_2$,
$\mathcal{M}$((A PHILOSOPHER^1)$_2$)$= \{\langle \sigma, x_2\rangle\}$, where
$\sigma = \lambda s\, [T/(s \models\ll a, [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], T; 1 \gg)]$.
$\mathcal{B}$(MET (A PHILOSOPHER^1)$_2$) $=$
$\lambda s, l\, [x/(s \models\ll \mathit{meet}, x, x_2, l^{[l/l \prec l^{rdl}]}; 1 \gg)]$,
$\mathcal{M}$(MET (A PHILOSOPHER^1)$_2$)$= \mathcal{M}$((A PHILOSOPHER^1)$_2$) $= \{\langle \sigma, x_2\rangle\}$.

Two possibilities now are available for the application of the rule N$_R$ for φ:

Case1 for $\boldsymbol{\varphi}$. N$_R$ is applied with $k = 0$, and the meaning type of (A PHILOSOPHER^1)$_2$ is kept in the storage. This gives a specific (*de re*) reading of (A PHILOSOPHER^1)$_2$ with respect to the noun φ:

$$\begin{aligned}\mathcal{B}(\varphi) = \lambda s, l\, [x/\ &(s \models\ll \mathit{logician}, x, l; 1 \gg) \wedge \\ &\qquad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l \circ l^{rdl}]}; 1 \gg)] \text{ and}\end{aligned}$$

$$\mathcal{M}(\varphi) = \{\langle \sigma, x_2\rangle\}.$$

Case2 for $\boldsymbol{\varphi}$. N$_R$ is applied with $k = 1$:

$$\begin{aligned}\mathcal{B}(\varphi) = \lambda s, l\, [x/(\ &[T/(s_{q_2} \models\ll a, [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], T; 1 \gg)] : \\ &[x_2/p_1(x, s, l) \wedge p_2(x, s_2, l_2)])] =\end{aligned}$$

$$\begin{aligned}\lambda s, l\, [x/(s_{q_2} \models\ll a,\ &[x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\ &[x_2/\ (s \models\ll \mathit{logician}, x, l; 1 \gg) \wedge \\ &\qquad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l \circ l^{rdl}]}; 1 \gg)]; 1 \gg)], \text{ and}\end{aligned}$$

$$\mathcal{M}(\varphi) = \emptyset.$$

Example 4. $\psi =$ (EVERY (LOGICIAN THAT MET (A PHILOSOPHER^1)$_2$)^4)$_5$.

$\mathcal{B}(\psi) = x_5$, and there are three different possibilities for the storage of ψ.

Case1 for ψ. The rule (NP) is applied with $k = 0$ for calculating $\mathcal{BMT}(\psi)$ after Case1 for φ has been applied. Both, the basic meaning type of (A PHILOSOPHER^1)$_2$ and that of ψ are left separately in the storage of ψ.

$\mathcal{M}(\psi) = \mathcal{M}(\varphi) \bigcup \{\langle \mathcal{BMT}(\psi), x_5\rangle\} = \{\langle \sigma, x_2\rangle, \langle \tau, x_5\rangle\}$, where

$$\begin{aligned}\tau = \lambda s\,[T/(s \models\ll \mathit{every},\ [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg)\wedge&\\ (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)],&\\ T; 1 \gg)].&\end{aligned}$$

The linguistic meaning of the NP ψ is:

$[\![\psi]\!] = \langle \mathcal{M}(\psi), x_5\rangle = \langle \{\langle \sigma, x_2\rangle, \langle \tau, x_5\rangle\}, x_5\rangle$.

What this semantical representation of ψ expresses is that x_5 is a representative (an instance) of the domain of the quantificational type τ. And it can be any one of all the elements of the domain. If instead of *every*, the quantificational relation was *five*, then x_5 is a representative of one of the five selected. There is not enough linguistic or extralinguistic information for taking decision about the scopes of σ and τ.

Case2 for ψ. The rule (NP) is applied with $k = 0$ after Case2 for φ. The narrow *de re* reading of (A PHILOSOPHER^1)$_2$ in φ gives rise of nonspecific (*de dicto*) reading of (A PHILOSOPHER^1)$_2$ in ψ.

$\mathcal{M}(\psi) = \{\langle \mathcal{BMT}(\psi), x_5\rangle\}$, where

$\mathcal{BMT}(\psi) =$

$$\begin{aligned}&\lambda s\,[T/(s \models\ll \mathit{every}, \mathcal{B}(\varphi)(s_4, l_4), T; 1 \gg)] =\\ &\lambda s\,[T/(s \models\ll \mathit{every},\\ &\qquad [x/(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)],\\ &\qquad\qquad [x_2/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg)\wedge\\ &\qquad\qquad\quad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)]; 1 \gg)],\\ &\qquad T; 1 \gg)].\end{aligned}$$

Case3 for ψ. The rule (NP) is applied with $k = 1$ for the calculation of $\mathcal{BMT}(\psi)$ after Case1 for φ has been applied. We get a local specific (local *de re*) reading of (A PHILOSOPHER^1)$_2$ with respect to the NP ψ.

$\mathcal{M}(\psi) = \{\langle \mathcal{BMT}(\psi), x_5\rangle\}$, where

$\mathcal{BMT}(\psi) =$

$$\begin{aligned}&\lambda s\,[T/(\ [T_2/(s_{q_2} \models\ll a, [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], T_2; 1 \gg)] :\\ &\qquad [x_2/(s \models\ll \mathit{every}, \mathcal{B}(\varphi)(s_4, l_4), T; 1 \gg)])] =\end{aligned}$$

$$\begin{aligned}&\lambda s\,[T/(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)],\\ &\qquad [x_2/(s \models\ll \mathit{every},\ [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg)\wedge\\ &\qquad\qquad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l\, \prec\, l^{rdl}]}; 1 \gg)],\\ &\qquad\quad T; 1 \gg)]; 1 \gg)].\end{aligned}$$

4 Quantificational VP Rule (VP$_3$)

This rule takes some quantifiers out of the storage of a verb phrase α and inserts them into its basis. It does not correspond to any syntactical contributions — the verb phrase itself has been already generated. The rule changes its semantical representation, i.e. its storage and the basis. That might happen because of the availability of new contextual information. While the quantificational rule (VP) "inserts" quantificational types into the object role of the verb, by this rule, the verb phrase meaning is inserted into the ranges of quantificational types. For transparency of the exposition this rule is stated separately, but it can be incorporated as a part of the two rules VP$_1$ and VP. The last would be more appropriate decision because some of the scope ambiguities are dependent on the phrase combinations and get resolved at the point of putting together the relevant expressions.

Let α be a VP, and its storage and basis be correspondingly, $\mathcal{M}'(\alpha)$ and $\mathcal{B}'(\alpha) = \lambda s, l\,[x/p(s,l,x)]$ for some proposition $p(s,l,x)$. Then the new basis and the new storage of α are defined in the following way:

$$\begin{aligned}\mathcal{B}(\alpha) = \lambda s, l\,[x/\ &(\sigma_1(s_{q_1}) : [x_{i_1}/\ldots\\ &\qquad(\sigma_k(s_{q_k}) : [x_{i_k}/(\mathcal{B}'(\alpha)(s,l) : x)])\ldots])] =\end{aligned}$$

$$\begin{aligned}\lambda s, l\,[x/\ &(\sigma_1(s_{q_1}) : [x_{i_1}/\ldots\\ &(\sigma_k(s_{q_k}) : [x_{i_k}/p(s,l,x)])\ldots])],\end{aligned}$$

where $k \geq 0$. The new storage is:

$\mathcal{M}(\alpha) = \mathcal{M}'(\alpha) - \{\langle\sigma_1, x_{i_1}\rangle, \ldots, \langle\sigma_k, x_{i_k}\rangle\}$.

As before, the restriction (QR) must be respected in selecting the order of the quantification.

Example 5. By using this rule for the VP MET (A PHILOSOPHER^1)$_2$, we get:

$\mathcal{B}$(MET (A PHILOSOPHER^1)$_2$) =

$$\begin{aligned}\lambda s, l\,[x/(s_{q_2} \models\ll a,\ &[x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)],\\ &[x_2/(s \models\ll \mathit{meet}, x, x_2, l; 1 \gg)];\\ &1 \gg)],\ \text{and}\end{aligned}$$

$\mathcal{M}$(MET (A PHILOSOPHER^1)$_2$) = $\emptyset$.

The result is a narrow nonspecific reading. Then following the (N$_R$) rule we get for the noun φ:

$$\begin{aligned}\mathcal{B}(\varphi) = \lambda s, l\,[x/\ &(s \models\ll \mathit{logician}, x, l; 1 \gg)\wedge\\ &(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)],\\ &\qquad\qquad [x_2/(s_2 \models\ll \mathit{meet}, x, x_2, l_2; 1 \gg)];\\ &\qquad\qquad 1 \gg)],\ \text{and}\end{aligned}$$

$\mathcal{M}(\varphi) = \emptyset$.

Case4 for ψ. Now using the last basis and storage for φ and following the NP rule for ψ we get a fourth possibility for its meaning type.

$$\mathcal{BMT}(\psi) = \lambda s\,[T/(s \models\ll \mathit{every}, \mathcal{B}(\varphi)(s_4, l_4), T; 1 \gg)] =$$

$$\begin{aligned}
&\lambda s\,[T/(s \models\ll \mathit{every},\ [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg)\wedge \\
&\qquad\qquad\qquad\qquad\qquad (s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad [x_2/(s_2 \models\ll \mathit{meet}, x, x_2, l_2; 1 \gg)]; \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad 1 \gg)], \\
&\qquad\qquad\qquad T; 1 \gg)],
\end{aligned}$$

$$\mathcal{B}(\psi) = x_5,$$

$$\mathcal{M}(\psi) = \{\langle \mathcal{BMT}(\psi), x_5 \rangle\}.$$

Case2 and Case4 for ψ are both nonspecific (*de dicto*) readings of A PHILOSOPHER in ψ, but there is a difference in their semantical structure.

Example 6. Going through the cases for ψ = (EVERY (LOGICIAN THAT MET (A PHILOSOPHER^1)$_2$)^4)$_5$ considered in the previous sections let find the semantical representation of a sentence with the restrictive relative clause ψ as the subject NP:

ϕ = (EVERY (LOGICIAN THAT MET (A PHILOSOPHER^1)$_2$)^4)$_5$ SMILED.

Case1 for ϕ. By the quantificational restriction (QR) there is only one way we to apply the rule S$_1$ with $k = 2$ and Case1 for ψ. The result is a specific (*de re*) reading of (A PHILOSOPHER^1)$_2$.

$$\begin{aligned}
&\mathcal{B}(\phi) = [s, l/\ (\sigma(s_{q_2}) : [x_2/ \\
&\qquad\qquad (\tau(s_{q_1}) : [x_5/(s \models\ll \mathit{smiles}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)])])] =
\end{aligned}$$

$$\begin{aligned}
&[s, l/\ (\sigma(s_{q_2}) : [x_2/(s_{q_1} \models\ll \mathit{every}, \\
&\qquad\qquad\qquad\qquad\qquad [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; \gg)\wedge \\
&\qquad\qquad\qquad\qquad\qquad\qquad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l \prec l^{rdl}]}; 1 \gg)], \\
&\qquad\qquad\qquad\qquad\qquad [x_5/(s \models\ll \mathit{smile}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)]; \\
&\qquad\qquad\qquad\qquad\qquad 1 \gg)])] =
\end{aligned}$$

$$\begin{aligned}
&[s, l/(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\
&\qquad\qquad\quad [x_2/(s_{q_1} \models\ll \mathit{every}, \\
&\qquad\qquad\qquad\qquad [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; \gg)\wedge \\
&\qquad\qquad\qquad\qquad\quad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l \prec l^{rdl}]}; 1 \gg)], \\
&\qquad\qquad\qquad\qquad [x_5/(s \models\ll \mathit{smile}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)]; \\
&\qquad\qquad\qquad\qquad 1 \gg)]; \\
&\qquad\qquad\quad 1 \gg)].
\end{aligned}$$

Case2 for ϕ. The linguistic meaning of a nonspecific (*de dicto*) reading of ϕ can be received by applying the sentence rule S_1 with $k = 1$ and using Case2 for ψ.

$$\mathcal{B}(\phi) = [s, l/(\mathcal{BMT}(\psi)(s_{q_1}) : [x_5/(s \models\ll \mathit{smile}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)])] =$$

$$\begin{array}{l}
[s, l/(s_{q_1} \models\ll \mathit{every}, \\
\qquad\qquad [x/(s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\
\qquad\qquad\qquad\qquad [x_2/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg) \wedge \\
\qquad\qquad\qquad\qquad\qquad (s_2 \models\ll \mathit{meet}, x, x_2, l_2^{[l/l \prec l^{rdl}]}; 1 \gg)]; 1 \gg)], \\
\qquad\qquad [x_5/(s \models\ll \mathit{smile}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)]; \\
\qquad\qquad 1 \gg)],
\end{array}$$

$$\mathcal{M}(\phi) = \emptyset.$$

Case3 for ϕ. Using Case3 for ψ and the rule S_1 with $k = 1$ we get the same representation as in Case1.

Case4 for ϕ. If we use the VP_3 rule, i.e. Case4 for ψ, and the rule S_1 with $k = 1$ we get the narrowest nonspecific (*de dicto*) reading of (A PHILOSOPHER^1)$_2$.

$$\begin{array}{l}
\mathcal{B}(\phi) = \\
[s, l/(s_{q_1} \models\ll \mathit{every},\ [x/\ (s_4 \models\ll \mathit{logician}, x, l_4; 1 \gg) \wedge \\
\qquad\qquad\qquad\qquad (s_{q_2} \models\ll a,\ [x/(s_1 \models\ll \mathit{philosopher}, x, l_1; 1 \gg)], \\
\qquad\qquad\qquad\qquad\qquad [x_2/(s_2 \models\ll \mathit{meet}, x, x_2, l_2; 1 \gg)]; \\
\qquad\qquad\qquad\qquad\qquad 1 \gg)], \\
\qquad\qquad\qquad [x_5/(s \models\ll \mathit{smile}, x_5, l^{[l/l \prec l^{rdl}]}; 1 \gg)]; \\
\qquad\qquad\qquad 1 \gg)],
\end{array}$$

$$\mathcal{M}(\phi) = \emptyset.$$

At a first glance, Case2 and Case4 seem to be equivalent, though they have different semantical structures, which convey subtle semantical differences.

Besides a semantical treatment of the quantificational ambiguities and scope resolutions depending on the particular context and speaker's references, there is a side consequence of introducing a semantical basis and a storage. At the level of the calculation of the semantical basis, the present approach preserves the primary intuitions about the main sentence predication as an application of the form VP(NP), i.e. the property denoted by the VP is the main predicate of the sentence, and it is predicated to the subject of the sentence denoted by the NP. The semantical difference between the simple NPs called individual terms, which typically refer to individuals, and the quantificational NPs, which do not refer to any singular individual, is explained and represented by the semantical storage. In Montague's PTQ all NPs are treated as generalized quantifiers, and the sentence predication is reversed for all types of subject NPs and VPs — NP(VP).

5 The Operation $\mathcal{R}$

The two-argument operation $\mathcal{R}$ on types of types of individuals which is defined below permits the quantificational information about two coindexed NPs to be conjoined in some appropriate way. This operation permits us to represent expressions having as constituents more than one NPs different from pronouns and with same index. This is possible when two different NPs are in an antecedent-anaphora relation and describe the referent in two different ways. Also it would be useful for languages that permit appositive NPs. In the definition that follows, $p_1(x)$, $p_2(x)$ and $p(x)$ are propositions.

(i) Let
$\sigma = [T/(s \models \ll Q, [x/p_1(x)], T; 1 \gg)]$ and
$\tau = [T/(s \models \ll Q, [y/p_2(y)], T; 1 \gg)]$, where $Q \in \{a, \mathit{the}\}$. Then,
$\mathcal{R}(\sigma, \tau) = \mathcal{R}(\tau, \sigma) =$
$[T/(s \models \ll Q, [x/p_1(x) \wedge p_2([z/y])], T; 1 \gg)]$.

This definition is useful for sentences like the following:

(The guardian)$_1$ killed the wolf that approached him$_1$, (the cruel man)$_1$.

(The man)$_1$ killed the wolf that happened to approach (the best shooter)$_1$.

(The man)$_1$, (the best shooter)$_1$, killed the wolf that approach him$_1$.

(A guardian)$_1$ killed the wolf that approached him$_1$, (a cruel man)$_1$.

(A guardian)$_1$, (a cruel man)$_1$, killed the wolf that approached him$_1$.

(ii) Let
$\sigma = [T/(s \models \ll a, [x/p_1(x)], T; 1 \gg)]$, and
$\tau = [T/(s \models \ll \mathit{the}, [y/p_2(y)], T; 1 \gg)]$. Then,
$\mathcal{R}(\sigma, \tau) = \mathcal{R}(\tau, \sigma) =$
$[T/(s \models \ll \mathit{the}, [x/p_1(x) \wedge p_2([z/y])], T; 1 \gg)]$.

(The guardian)$_1$ killed the wolf that approached him$_1$, (a cruel man)$_1$.

(The guardian)$_1$, (a cruel man)$_1$, killed the wolf that approached him$_1$.

(A cruel man)$_1$, (the best shooter in the region)$_1$, killed the wolf that approached (him$_1$ who was in bad mood)$_1$.

(A cruel man who happened to be (the best shooter in the region)$_1$)$_1$, killed the wolf that approached him$_1$.

Here we assume that the determiner the is "stronger" than a. In the last two examples, it might be that the order of the determiners is important. In such case one might prefer to define $\mathcal{R}$ as nonsymmetric and its value to be determined by its first argument.

(iii) Let
$\sigma = [T/(s \models\ll T, x_i^r; 1 \gg)]$ and
$\tau = [T/(s \models\ll Q, [x/p(x)], T; 1 \gg)]$, where $Q \in \{a, the\}$. Then,
$\mathcal{R}(\sigma, \tau) = \mathcal{R}(\tau, \sigma) =$
$[T/(s \models\ll Q, [x/p(x^r)], T; 1 \gg)]$.

The last case is needed when the individual terms are treated by the (Alt.1), and for examples like the following:

Mary wants every doll she, a spoiled child sees.
Mary, a spoiled child wants every doll she sees.
William the Conquerer brought the prosperity of his country.
Your friend Jim wants a bicycle he has dreamed about.
He, your friend Jim, wrote a book criticizing an article written by each philosopher.

Acknowledgments

I would like to thank Alexander Gelbukh for the crucial assistance and support in writing and organizing the grammar rules presented in this paper. I would like to include a special thanks to Arnold Insel for spending many hours with me on discussions of the problems covered by the grammar, patiently reading my writings, and for the Latex lessons.

References

1. Loukanova, R. 2002. *Quantification and Intensionality in Situation Semantics.* in this volume.

Sign Language Translation via DRT and HPSG

Éva Sáfár and Ian Marshall

School of Information Systems,
Norwich, NR4 7TJ, United Kingdom
{es,im}@sys.uea.ac.uk

Abstract. We present an overview of the language-processing component of an English-Text to Sign-Languages translation system[1], concentrating on the role of Discourse Representation Structures as the intermediate semantic representation and the use of HPSG for synthesis of equivalent signed sequences. A short introduction to the main characteristics of Sign Languages is also presented.

1 Introduction

The research and system components described in this paper are part of a multilingual sign translation system designed to translate from English text into a variety of national sign languages (NGT (Dutch), DGS (German) and BSL (British)). Such sign languages are 'natural' to pre-lingually deaf signers for whom an oral/written language is typically their second language. Hence, this work contrasts significantly with other text-to-sign-language translations systems, such as VCom3D [24] and Simon [12], which present textual information as SE (Signed English) or SSE (Sign Supported English)[2]. The Tessa system [9] translates from speech to BSL in a restricted domain, but is built on an inflexible template-based grammar.

The overall architecture of the English text to sign language system is illustrated in Figure 1. This is designed as a collection of automatic transformation components augmented by user-interaction. English text is input to the CMU parser [22], whose output is a set of links - a linkage - for a sentence. The CMU parser is robust and covers a significantly high proportion of English linguistic phenomena. The parser often produces a number of linkages for one sentence. Currently the user selects the correct linkage by direct intervention. The transformation from the appropriate output linkage to its DRS is performed using λ-DRS definitions associated with link types which are composed using λ-calculus β-reduction and DRS merging [1].

The morphology and syntax of sign-generation from this semantic representation is defined within the framework of Head-Driven Phrase Structure

[1] This work is incorporated within *ViSiCAST*, an EU Framework V supported project which builds on work supported by the UK Independent Television Commission and Post Office. The project develops virtual signing technology in order to provide information access and services to Deaf people.

[2] SE uses signs in English word order and follows English grammar, while SSE signs only key content words of a sentence again retaining English word order.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 58–68, 2002.

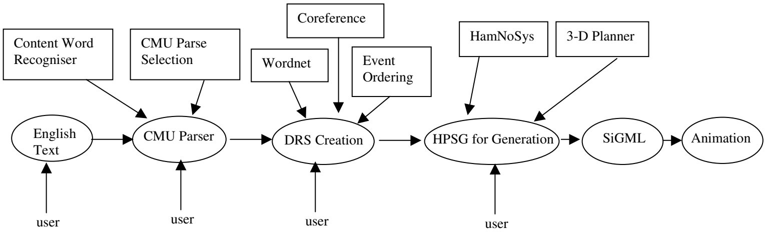
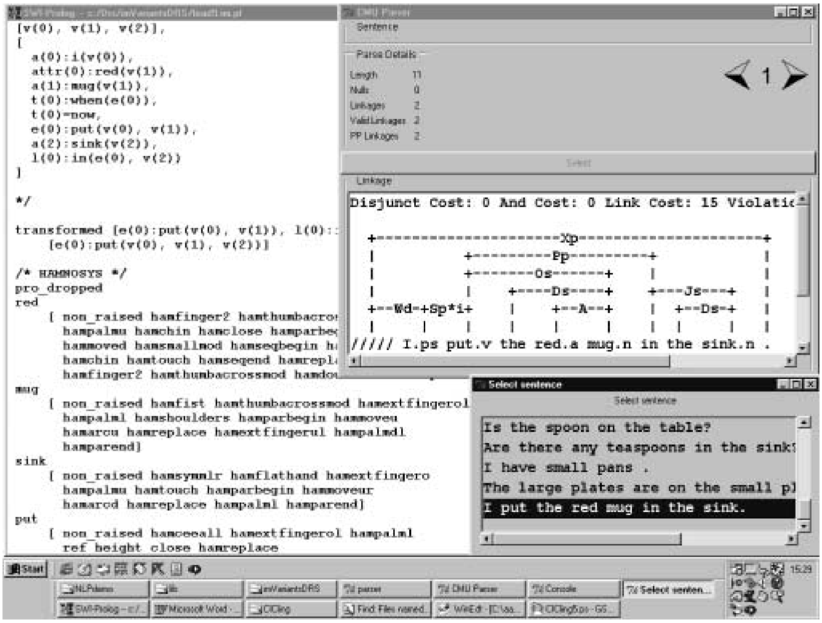


Fig. 1. Stages of English text translation to sign language



```
[v(0), v(1), v(2)],
[
  a(0):i(v(0)),
  attr(0):red(v(1)),
  a(1):mug(v(1)),
  t(0):when(e(0)),
  t(0)=now,
  e(0):put(v(0), v(1)),
  a(2):sink(v(2)),
  l(0):in(e(0), v(2))
]

*/

transformed [e(0):put(v(0), v(1)), l(0)::
    [e(0):put(v(0), v(1), v(2))]

/* HAMNOSYS */
pro_dropped
red
    [ non_raised hamfinger2 hamthumbacro
      hampalmu hamchin hamclose hamparbe
      hammoved hamsmallmod hamseqbegin h
      hamchin hamtouch hamseqend hamrepl
      hamfinger2 hamthumbacrossmod hamdo
mug
    [ non_raised hamfist hamthumbacrossmod hamextfingerol
      hampalml hamshoulders hamparbegin hammoveu
      hamarcu hamreplace hamextfingerul hampalmdl
      hamparend]
sink
    [ non_raised hamsymmlr hamflathand hamextfingero
      hampalmu hamtouch hamparbegin hammoveur
      hamarcd hamreplace hampalml hamparend]
put
    [ non_raised hamceeall hamextfingerol hampalml
      ref_height close hamreplace
```

Sentence

Parse Details

Length	11
Nulls	0
Linkages	2
Valid Linkages	2
PP Linkages	2

Linkage

```
Disjunct Cost: 0 And Cost: 0 Link Cost: 15 Violati

 +----------------------Xp----------------------+
 |          +----------Pp----------+            |
 |          +--------Os-------+    |            |
 |          |    +----Ds-----+    +---Js---+   |
 +--Wd-+Sp*i+    |    +--A--+    |  +--Ds-+   |
 |     |    |    |    |     |    |  |     |   |
///// I.ps put.v the red.a mug.n in the sink.n .
```

Select sentence

```
Is the spoon on the table?
Are there any teaspoons in the sink
I have small pans .
The large plates are on the small pl
I put the red mug in the sink.
```

Fig. 2. Input sentences, CMU Parser linkage, DRS and HamNoSys

Grammar (HPSG). This linguistic analysis is then linked with the animation technology [14] via a Signing Gesture Markup Lanugage (SiGML), that is an XML-compliant representation of gestures [12] and is based on the refined HamNoSys [19] sign notation. Figure 2 illustrates the current demonstrator system that allows selection of a sentence from a number of available sentences which is passed to the CMU parser and then to the DRS generator and HPSG synthesis systems.

2 Sign Language Features

Natural sign languages have a number of similarities to oral natural languages, though the three dimensional nature of the space around a signer offers a number of opportunities unavailable to oral languages. The following descriptions of major features of British Sign Language (BSL) are based on [2] and [23].

- Sign Order – BSL has a topic-comment structure, in which the main informational subject or topic is signed first. The flexibility of sign order follows from additional information carried in the directional verbs and non-manual features, such as facial expression and eye-gaze.
- Non-manual features (multi-modal signs) – Though the phonology of the manual components of signing is the most intensely meaning carrying component of signing, this is augmented by a rich variety of non-manual features which carry additional information. Facial expressions associated with the position of eyebrows distinguish between declarative (neutral brows), yes/no questions (raised brows) and wh-questions (furrowed brows). Mouth patterns can provide adverbial information or help disambiguate manually similar signs. Facial expression and body posture can indicate the signer's attitude to the accompanying proposition.
- Directional or Agreement Verbs – Agreement verbs incorporate within the sign information about person and number of the subject and indirect object. This is realized by the direction of the movement of the verb in the syntactic signing space. The signing of the verb usually begins at the position of the subject and ends at the position of the object(s) (GIVE, PUT, TELL, etc). Because of this agreement Sign Languages can be described as prodrop languages.
- Classifiers – Classifiers are handshapes that can denote an object from a group of semantically related objects. The handshape is used to denote a referent from a class of objects that have similar features (e.g. BICYCLE-PASS). Some verbs allow such a handshape to be incorporated within the sign of the verb.

3 Semantic Representation

The approach to English to Sign Language translation is based upon the use of Discourse Representation Theory (DRT) [13] for the intermediate meaning representation of meaning. A DRS (Discourse Representation Structure) is a two part structure involving a list of variables denoting the nominal discourse referents and conditions (a collection of propositions which capture the semantics of the discourse). The top left window of Figure 2 illustrates the DRS structure for 'I put the red mug in the sink'.

DRT was chosen as the underlying theory because it decomposes linguistic phenomena into atomic meaning components (propositions with arguments), and hence allows isolation of tense/aspect and modifying phenomena that are realized in different sign language grammatical constructs or modalities (see Section 2). In addition, the centrality of co-referentiality in DRT is reflects the need to appropriately determine how to assign fixed positions in signing space to significant discourse referents .

DRSs, described in [13], are modified to achieve a more sign language oriented representation that subsequently supports an easier mapping into a sign language grammar. In [13] only event propositions are labeled for use as arguments with temporal predicates. This has been extended by introducing labels for different kinds of semantic predicates. As in Verbmobil's VIT representation the labeling of all semantic entities allows a flat representation of the hierarchical structure of arguments and operator embeddings [10,11]. In contrast to Vermobil's uniform labeling, an ontology for all DRS propositons has been introduced to facilitate the mapping between the flat semantic structure of the DRS to the nested input structure of the target language specific HPSG, as required by the generation algorithm in ALE [5].

Higher order predicates which take labels as arguments are convenient for handling verb modifiers, negation and adverbials (e.g.: [attr1:big(X), attr2:very(attr1)]), which are realised by multiple modalities in sign languages (non-manual components parallel to manual signs) especially facial expressions which convey intensity and head nod which conveys negation. The label taxonomy also aids location of possible anaphoric referents and temporal information.

This form of representation also has the advantage, as [10] claim, that additional constraints which are important for generation in the target language, e.g. topic/focus in sign languages, may be made explicit.

The translation from a CMU linkage to its DRS representation occurs via Definite Clause Grammar (DCG) rules implemented in Prolog. A link dictionary maps each link type to a λ-expression DRS definition (λ-DRS) [15,20]. The DCG then concatenates the λ-DRSs in the appropriate order and instantiates the arguments of the predicates appropriately [1]. Functional application (β-reduction) and the DRS merge operation combine the λ-DRSs into the final DRS [4].

The DRS representation is converted to an appropriate semantic (SEM) format for the ALE generation algorithm. As this format is hierarchical, the DRS labels facilitate construction of the appropriate nested form. In addition, this conversion handles transformation of differing numbers of complements between the English derived DRS events/states and the sign language oriented equivalents. In the case of the example sentence this involves converting the two argument predicate 'put' and its destination location as an adjunct into a BSL 3 argument SEM structure containing the relation PUT.

4 HPSG

The synthesis stage involves development of sign language grammars consistent with HPSG theory [18]. We use a Semantic Head-Driven (SHD) generator [21] implemented in ALE (v3.2), an extension to Prolog[3]. The HPSG framework has not been used widely for generation, however a small number of projects have taken this approach (e.g. LinGO has been used to build a large-scale grammar for English using HPSG which is implemented in the Verbmobil machine translation system). In addition, the attempts to characterise sign language grammar have not typically elected to use the HPSG framework [17], however some sign language constructs have been analysed in an HPSG framework [8].

However, there are sound reasons in favour of HPSG for sign language modelling. One hypothesis holds that the variation in sign languages is less substantial in their grammars in comparison with their lexicons, therefore a lexicalist

[3] Our German ViSiCAST partner explores the possibilities of LinGO [7]

approach is suitable for developing grammars for the three target languages in parallel. Differences are encoded in the lexicon, while grammar rules are usually shared with occasional variation in semantic principles. A further consideration in favouring HPSG is that the feature structures can incorporate modality-specific aspects (non-manual features) of signs appropriately.

In the following, we discuss a provisional BSL grammar implemented in ALE. It involves the standard components of an HPSG grammar, the feature structure specification, the lexicon, the grammar rules and principles. For each of these we characterise the significant aspects as they relate to the sign languages.

4.1 Feature Structure

The feature structure is relatively large but consists of reasonably standard components like phonetic (PHON), syntactic (SYN) and semantic (SEM) structures. Much of the detail of the feature structure is focused on fine grain detail in the PHON component describing how signs are constructed from handshape, palm orientation, finger direction and movement information. The argument structure and the agreement components of the SYN structure determine conditions under which signs can be combined into a grammatically correct physical realisation. SEM structures include semantic roles and indexing as in LinGO, which proved to be necessary despite the nested goal definition in ALE to determine syntactic roles and agreement for a relatively free word order language (see also 4.2 and 4.4).

4.2 The Generation Algorithm and the Semantic Input

ALE's internal generation algorithm is semantic head driven SHD, a natural approach to generation with HPSG. It operates by discovering a pivot, which is the lowest node in a derivation tree that has the same semantics as the root.

Grammar rules are divided into two kinds, chain rules (which have a semantic head - whose head daughter's logical form is identical to the logical form of the mother) and non-chain rules (which have no semantic head or are lexical entries). The pivot is identified as the mother node of a non-chain rule operating in a top-down fashion. After the pivot has been found, it generates bottom-up using chain-rules to connect the semantic-heads to the pivot [5].

A consequence of the ALE algorithm is that it requires a nested semantic (SEM feature) input structure illustrated in Example (1). In the remaining text we will use the term *semantic input* for the input goal description:

```
(1) sent, sem: (mode:decl, index:SENT,
          restr:[(sit:SIT, reln:put, addressee:(ref,Indv0),
                                     act:(ref,Indv2), thm:(ref,Indv1),
               args:[(index:(ref,Indv0), count:sg,
                              restr:[(sit:SIT, reln:i)]),
                     (index:(ref,Indv1),
                              restr:[(sit:SIT, reln:red,
                    args:[(index:Indv1, count:sg,
                        restr:[(sit:SIT, reln:mug)])])]),
        (index:(ref,Indv2), count:sg,
              restr:[(sit:SIT, reln:sink)])])]))
```

Indices are introduced in the same way as with other generation algorithms such as the Shake-and-Bake algorithm [6]. Eventually these will be exploited for agreement and for associating discourse objects with particular positions in signing space for the purposes of co-reference.

4.3 Lexical Entries and Rules

ALE provides not only the type hierarchy declaration, format for lexical entries and the mechanism for unification, but also a way to change morphological realization of lexical entries using lexical rules. The standard ALE implementation generates a result which is a sequence of lexical items derived from the left hand sides of lexical rules and application of lexical rules used in the derivation.

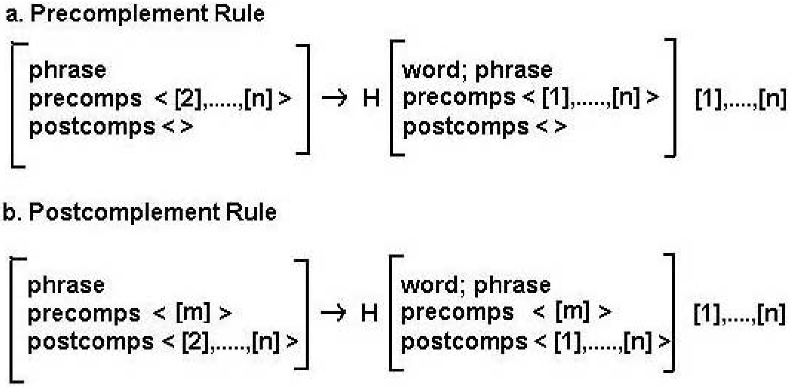
```
(2) [Brow, hamfist, hamthumbacrossmod, hamextfingerol, hampalml,
     hamshoulders, hamparbegin, hammoveu, hamarcu, hamreplace,
     hamextfingerul, hampalmdl, hamparend]  --->   PHON,SYN,SEM
```

We have adapted the left hand side (LHS) of ALE lexical items to be a list of HamNoSys phonetic transcription symbols, so that successful generation produces a list of signs (each of the latter being a list of its phonology). Example (2) illustrates a typical lexical entry, here ‘mug’. One consequence of this, however, is that the use of ALE’s lexical rules to characterise phonological relationships is prohibited. This is due to a restriction that lexical rules are applied during lexicon compilation, when new lexical entries are derived from existing ones. During the generation process the input word is simply looked up in the static lexicon with two different sets of variables. Because of the referencing mechanism in ALE the bindings of those variables are lost in the generation process. However, via unification and using principles, it is possible to instantiate the phonetic structure (PHON) on the right hand side, and propagate this to the LHS. In example (2), the uninstantiated non-manual (eye-)Brow movement that accompanies the manual features of the sign is determined by the mode of the sentence via the first non-chain rule, which associates this semantic input feature with the phonological Brow movement. This solution has the positive side effect, that a dynamic lexicon is created without increasing compilation time.

Currently the lexicon is relatively small, consisting of entries for 50 signs (mainly from a kitchen domain). However, these entries contain a variety of challenging verbs, nouns and modifiers which permit investigation of significant sign language grammatical constructs.

4.4 Rules

Currently the ALE implementation has 8 rules for BSL. These rules deal with sign order of (pre-/post-)modifiers (adjuncts) and (pre-/post-)complements.



a. Precomplement Rule

$$\begin{bmatrix} \text{phrase} \\ \text{precomps} < [2],.....,[n] > \\ \text{postcomps} < > \end{bmatrix} \rightarrow H \begin{bmatrix} \text{word; phrase} \\ \text{precomps} < [1],.....,[n] > \\ \text{postcomps} < > \end{bmatrix} \; [1],....,[n]$$

b. Postcomplement Rule

$$\begin{bmatrix} \text{phrase} \\ \text{precomps} < [m] > \\ \text{postcomps} < [2],.....,[n] > \end{bmatrix} \rightarrow H \begin{bmatrix} \text{word; phrase} \\ \text{precomps} < [m] > \\ \text{postcomps} < [1],.....,[n] > \end{bmatrix} \; [1],....,[n]$$

Fig. 3. Precomps and Postcomps rules

Rules of the standard HPSG model, which were designed mainly for English, have been modified to reflect the character of sign languages.

BSL is a topic-comment language (mainly but not necessarily SOV), hence a SUBJ, COMPLEMENT distinction is less appropriate. In the SYN component of a lexical entry, PRECOMPS and POSTCOMPS features permit it to subcategorize for its own kinds of complements. From this follows the introduction of recursive precomp- and postcomp-rules which permit an arbitrary number of complements. To compensate for the lack of a Subject-Head rule or schema, a terminating rule - the Last-Complement rule - has been introduced. The last complement is therefore not necessarily the subject. The subject is just one of the complements, which can be identified by feature-sharing between the lists of complements and the SEM substructure (see example (1), where Indv2 = actor/agent).

4.5 Principles

Currently there are 4 kinds of principles, which deal with mode, pluralization of nouns and verbs, subject and object pronoun drop.

The mode principle inspects the MODE feature in the semantic component and returns a value for the facial expression which has to accompany the signing. In example (1) the mode of the sentence is declarative (MODE:decl), therefore the feature of BROW is instantiated to a neutral expression, which is *non_raised.* Brows have to be *furrowed* or *raised* in wh-questions and yes-no questions respectively.

In BSL nominal plurals can be expressed in several different ways. Some plurals are formed by repeating the sign (usually three times) each repetition beginning at the location where the previous finished. Neither singular signs which involve internal repetition nor body anchored signs (ones which involve contact between the dominant hand and another part of the body) can be pluralized in this way. However, such signs can take a proform (a 'pronominal' handshape classifier) which can be repeated in a comparable fashion. Quantifiers, which occur before the noun in BSL, are also used for pluralization, but quantifiers can also be expressed as part of the internal construction of a sign. The distributive movement of the verb expresses that members of a group are involved individually in an action, but sweeping movement indicates the collective involvement of the whole group. Repetition of some verbs can mean either that one individual repeats the action or that many individuals do the same action [23]. Currently, of these possibilities of noun pluralization, we handle nouns, which can be repeated and non-repeatable ones with external quantifiers. However, for pluralization of the remaining group of nouns the feature structure design contains the relevant classifier information about the possible proforms (substitutors) for future development.

The current implementation of the plural_principle for nouns takes the SEM:COUNT information from the SEM component (COUNT in example (3)). The lexical item determines whether it allows plural repetition, if so, then the PHON:MAN:MOV:REPEAT feature (MOV = movement) is instantiated to the HamNoSys symbol expressing repetition in different locations and COUNT from the semantic input is propagated as the current value to the noun's SYN:AGR:NUM feature while PHON:MAN:MOV:REPEAT is instantiated to 'no'. In all other cases PHON:MAN:MOV:REPEAT remains uninstantiated.

```
(3)
 plural_principle_noun(syn:(allow_pl_repeat:yes_loc_indiv_finite,
                            head:(noun,agr:(num:Sg,per:Per))),
                sem:count:pl,
                syn:(allow_pl_repeat:no,
                     head:(noun,agr:(num:pl,per:Per))),
                man:mov:[(repeat:[hamrepeatcontinueseveral])])
        if true.
```

Verb pluralization is handled in a similar way, however the repeated verb motion is only permitted if the index of the semantic role and the index of the appropriate complement are identical.

Sign languages typically contain verbal signs which allow pronoun drop (prodrop), where one or more of the subject, object or indirect object (or actor, theme, addressee respectively) are omitted and incorporated within the sign for the verb itself. The actor and addressee are included within the sign for the verb as starting and/or end position of the movement (so-called directional verbs). In the case of a direct object pronoun the handshape of the sign for the verb is inherited from the object/theme (so called classifier proforms). This phenomenon reflects a similar relation between rich agreement and non-overt expression of subject/object pronomina [3], as in many languages, such as Italian and Hungarian. Indeed prodrop is found also in Chinese, which allows for 'topic-drop' without such rich morphology. Topic-drop is also possible in sign languages, but this phenomenon is not currently addressed in our grammar.

The non-overt realization of the pronomina (prodrop) is catered for by an empty lexical entry whose LHS is instantiated to an empty list and has non-instantiated RHS feature structure values. When the complement rules are processed, the prodrop principles check the semantic head for the values of subject and object prodrop features in all three persons. The possible values are *can*, *can't* and *must*. If it is *must*, the empty lexical item is chosen, that is not of type *word* but *dropped* to avoid ambiguity. The feature structure for such a lexical item is looked up in the lexicon using the RELN feature in the SEM component, (in fact to achieve this we drop out of ALE into in-line Prolog and use its ALE representation of lexical entries using the ALE operator (`--->`)[4]. In this way, the required SYN information of the empty string, which has to be unified with the complement information of the verb, is instantiated. This is an important step, as the verb may need the index of the noun for start and end position of the movement or the classifier information for the handshape as discussed above. If the prodrop value is *can't*, generation proceeds normally, generating the daughter in the usual way for separate signs. If the value is *can*, both solutions are generated, however a preferred order is realized by arranging the order of Prolog predicates accordingly.

5 Current Status and Conclusions

The system is a modular architecture which successfully integrates CMU link grammar, DRSs, HPSG and supporting NLP resources such as WordNet [16] and name lists.

Currently the translation system of CMU linkages into the DRS-based intermediate semantic representation handles the following linguistic phenomena including an unrestricted number of noun and verb modifiers, subject and object type relative clauses, prepositional phrases as adjunct of verb phrases and of noun phrases, actives, passives, wh-questions, yes-no questions and negation. This is approximately a 50% coverage of the CMU grammar link, though these are involved in common syntactic constructions.

The HPSG based synthesis sub-component involves a small sign lexicon but with a sufficient variety of different kinds of signs to allow us to explore the use of constraint based unification for sign language generation. The initial indications are that despite some technicalities which have had to be overcome in using ALE

[4] ALE supports empty categories, however they could not been used for prodrop. Empty categories in ALE are declared as lexical entries in a special format, therefore they suffer from a similar deficiency as lexical rules (see Section 4.3). Inheriting syntactic information from another lexical entry would not be possible in this way without duplicating lexical entries and therefore increasing the size of the static lexicon. We also considered Wilcock's suggestion [25] to eliminate empty categories as non-preferable entities in the theory. His ProFIT/SAX/SGX implementation uses a Complement Extraction Lexical Rule as proposed by [18], which is not re-implementable in our grammar, because of the general problems with lexical rules. Another way was to write a general lexical item with very few instantiated values, and include strict constraints in the grammar rules by goals. Currently, this solution has been suspended as a non-general and non-elegant solution to the prodrop problem.

as an implementation platform, this approach is fruitful. In general, grammatical concepts which have been used with oral languages (such as prodrop) and techniques which address these have been productive in developing the HPSG from sign language. The main contrast between the two kinds of languages lies in the more complex morphophonological components of sign language for which the HPSG lexicalist approach is highly appropriate.

References

1. Blackburn, P., Bos, J.: Representation and Inference for Natural Language. A First Course in Computational Semantics. Vol. II. http://www.coli.uni-sb.de/~bos/comsem/book1.html (1999)
2. Brien,D. (Ed.): Dictionary of British Sign Language/English. London,Boston. 1992
3. Bos,H.: Agreement and Prodrop in Sign Language of the Netherlands. In: Drykoningen,F, Hengeveld,K (Ed.):Linguistics in the Netherlands. Amsterdam/Philadelphia. 1993
4. Bos,J., Mastenbroek,E., McGlashan,S., Millies,S., Pinkal,M.: A Compositional DRS-based Formalism for NLP Applications. Report 59. Universitaet des Saarlandes. 1994
5. Carpenter,B., Penn,G.: The Attribute Logic Engine. User's Guide. Version 3.2 Beta. Bell Labs. 1999
6. Copestake,A., Flickinger,D., Malouf,R., Riehemann,S., Sag,I.: Translation Using Minimal Recursion Semanitics. In: Proceedings of the 16th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95). 1995
7. Copestake,A., Carrol,J., Malouf,R., Oepen,S., and others: The (new) LKB System. Version 5.2. Documentation. 1999
8. Cormier,K.A.: Grammatical and Anaphoric Agreement in American Sign Language. Graduate School of the University of Texas at Austin. Master Thesis. 1998
9. Cox,S.J., Lincoln,M., Nakisa,M.J., Coy,J.: The Development and Evaluation of a Speech to Sign Translation System to Assist Transactions. In: Submitted to International Journal of Human Computer Studies. 2001
10. Dorna,M., Emele,M.: Semantic-based Transfer. Report 143. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING-96). Copenhagen. 1996
11. Dorna,M.: Semantikbasierter Transfer. (Doctoral Thesis). In: Arbeitspapiere des Instituts fuer maschinelle Sprachverarbeitung (AIMS). Vol.6, No.2. 2000
12. Elliott,R., Glauert,J.R.W., Kennaway,J.R., Marshall,I.: The Development of Language Processing Support for the ViSiCAST Project. In: Assets 2000. 4th International ACM SIGCAPH Conference on Assistive Technologies. New York. 2000
13. Kamp,H., Reyle,U.: From Discourse to Logic. Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers. 1993
14. Kennaway,J.R.: Synthetic Animation of Deaf Signing Gestures. In: The Fourth International Workshop on Gesture and Sign Language Interaction (GW2001). City University, London, UK. 2001
15. Marshall,I., Safar, E.: Extraction of Semantic Representations from Syntactic CMU Link Grammar linkages. In: Recent Advances in Natural Language Processing (RANLP), G. Angelova et al (ed), Tzigov Chark Bulgaria. 2001

16. Miller,G.A., Beckwith,R., Fellbaum,Ch., Gross,D., Miller,K.: An Introduction to WordNet. An On-line Lexical Database. http://www.cogsci.princton.edu/∼wn/. 1993
17. Neidle C, Kegl J, MacLaughlin D, Bahan B, Lee R.G.: The Syntax of American Sign Language. MIT Press. 2000
18. Pollard, C., Sag,I.A.: Head-Driven Phrase Structure Grammar. The University of Chicago Press, Chicago. 1994
19. Prillwitz,S., Leven,R., Zienert,H., Hanke,T., Henning,J., others: Hamburg Notation System for Sign Languages - An Introductory Guide. International Studies on Sign Language and the Communication of the Deaf, Volume 5. Institute of German Sign Language and Communication of the Deaf, University of Hamburg. 1989
20. Safar,E., Marshall,I.: Translation of English Text to a DRS-based Sign Language Oriented Semantic Representation. In: Conference sur le Traitment Automatique des Lanmgues Naturelles (TALN) vol 2, pp297-306. 2001
21. Shieber,M., van Noord, G., Moore, C., Pereira, F.C.N.: A Semantic-head-driven Generation Algorithm for Unification-based Formalisms. In: 27th Annual Meeting of the Association for Computational Linguistics. Vancouver. pp 7-17. 1989
22. Sleator,D., Temperley,D.: Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196. 1991
23. Sutton-Spence,R., Woll,B.: The Linguistics of British Sign Language. An Introduction. University Press, Cambridge. 1999
24. Wideman,C.J.: Internet-enabled, 3D, SigningAvatar Software Offers Accessility for the Deaf. http://www.signingavatar.com/nc/presentations/april_fose.html. 2000
25. Wilcock,G.: Natural Language Generation with Head-Driven Phrase Structure Grammar. Centre for Computational Linguistics, Department of Language Engineering. University of Manchaster. PhD. 1998

Multilayered Extended Semantic Networks as a Language for Meaning Representation in NLP Systems

H. Helbig and C. Gnörlich

University of Hagen,
Applied Computer Science VII,
Intelligent Information and Communication Systems,
58084 Hagen, Germany
{hermann.helbig/carsten.gnoerlich}@fernuni-hagen.de

Abstract. Multilayered Extended Semantic Networks (abbreviated: MultiNet) are one of the few knowledge representation paradigms along the line of Semantic Networks (abbreviated: SN) with a comprehensive, systematic, and publicly available documentation. In contrast to logically oriented meaning representation systems with their extensional interpretation, MultiNet is based on a use-theoretic operational semantics. MultiNet is distinguished from the afore-mentioned systems by fulfilling the criteria of homogeneity and cognitive adequacy. The paper describes the main features of MultiNet and the standard repertoire of representational means provided by this system. Besides of the structural information, which is manifested in the relational and functional connections between nodes of the semantic network, the conceptual representatives of MultiNet are characterized by embedding the nodes of the network into a multidimensional space of layer attributes. To warrant cognitive adequacy and universality of the knowledge representation system, every node of the SN uniquely represents a concept, while the relations between them have to be expressed by a predefined set of about 110 semantic primitive relations and functions. The knowledge representation language MultiNet has been used as an interface in several natural language processing systems. It is also suitable as an interlingua for machine translation systems.

1 Introduction

Prior to the design of a knowledge representation system (abbreviated: KRS) which is to be broadly acceptable, one should have a collection of criteria such a KRS must fulfill. This claim is especially important, if the planned KRS is to be used as an interlingua for the meaning representation of natural language information, which can be employed in different NLP systems. Unfortunately, there is no general consensus with regard to the criteria such a system has to meet. Nevertheless, designing the knowledge representation language of Multilayered Extended Semantic Networks (the so-called MultiNet paradigm (Helbig, 2001)),

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 69–85, 2002.

which has been developed along the line of tradition of the well known Semantic Networks going back to the work of Quillian (Quillian, 1968), we started with a predefined set of criteria. To the best of our knowledge, there is no KRS satisfying these criteria in every respect. But – as we believe – MultiNet comes very close to these requirements. The most important of the above mentioned criteria to be met by the representational means of a KRS or of an interlingua are the following:

Global requirements

- Universality: The representational means are applicable in every domain (i.e. they are not adapted to a special field of discourse). They have also to be independent of a specific natural language.
- Cognitive adequacy: They put the concept into the center of the semantic representation where every concept has a unique representative. All other expressional means, especially the relations between them, have to be considered as constructs of a metalanguage with regard to the concept level.
- Homogeneity: They can be used to describe the semantics of lexemes as well as the semantics of sentences or texts.
- Interoperability: They are the carriers of all NLP processes (be it lexical search, syntactic-semantic analysis, logical answer finding, natural language generation, or the translation into a foreign language).
- Automatability: They must allow for an automatic (or at least computer assisted) knowledge acquisition.
- Practicability: They should be technically treatable without inappropriate effort and also be easily communicable in a certain community or in a team.

Internal structural requirements

- Completeness: There should be no meaning of a natural language construct which can not be represented properly.
- Optimal granularity: On the one hand, the system should be fine-grained enough to allow for the representation of all essential differences in meaning. On the other hand, the system need not mirror the tiniest nuances of meaning, otherwise it will not be manageable on a computer.
- Consistency: Contradictions must not be derivable from the basic definitions of the representational means.
- Stratification: It must be possible to represent the different semantic aspects (like intensional vs. extensional aspects, or immanent vs. situational aspects) in different layers of the KRS.
- Local interpretability: Each elementary construct (especially nodes and links in a network representation) must have its own context-independent interpretation and must be connected with special logical devices (inference rules, inheritance principles, etc.)

MultiNet is distinguished from other semantic network representations like KL-ONE (Brachman, 1978) and its successors (e.g. (Allgayer and Reddig, 1990), (Peltason, 1991)) as well as from logically oriented knowledge representations

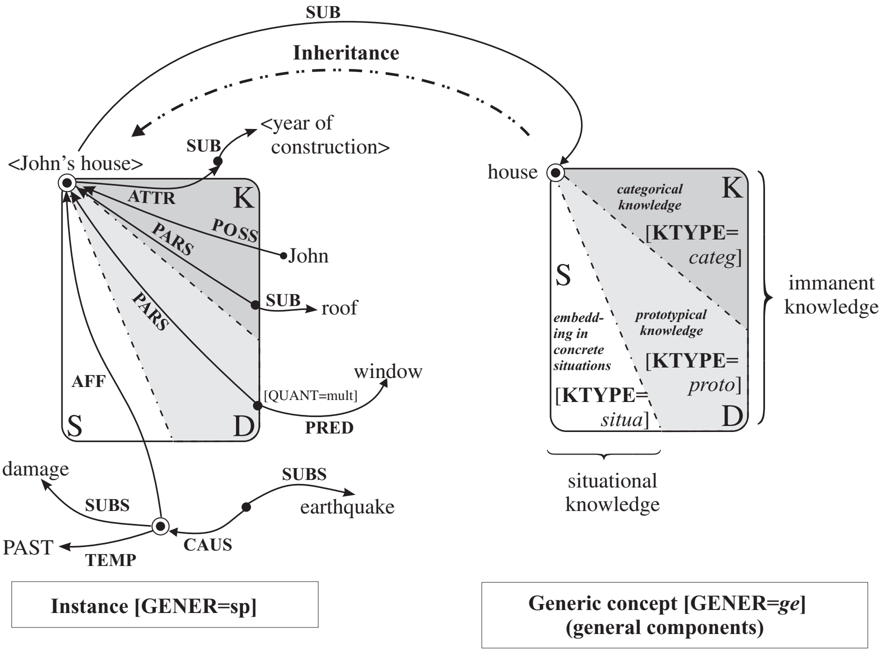


Fig. 1. The representation of concepts in MultiNet

like DRT (Kamp and Reyle, 1993) or Description Logic (Baader et al., 1998) by the criteria of cognitive adequacy and homogeneity. All these KRS have a model-theoretic extensional foundation which can not be upheld for many concepts (like "intension", "charm") or even for common properties (like "tall", "happy"). It is not known that the above cited systems have been used for the semantic description of large stocks of lexical information, while MultiNet has been the base for the full syntactic-semantic description of more then 14000 lexemes ((Schulz, 1999), this work is being continued). From all semantic network paradigms, MultiNet comes closest to SNePS (Shapiro, 1999) but is essentially distinguished from this system by its multilayered structure and the encapsulation of concepts.

2 The Main Characteristic of the MultiNet Paradigm

As with other semantic networks, concepts are represented in MultiNet by nodes, and relations between concepts are represented as arcs between these nodes (see Figure 1). Aside of that, MultiNet has several characteristic features, the most important of them are:

1. Every node is classified according to a predefined conceptual ontology forming a hierarchy of sorts (see Figure 2). From that hierarchy, a sort is assigned to every node of the SN.

2. The nodes have a well-defined inner structure which is essentially given by the assignment of the nodes to certain layers of the network specified by the attribute-value structure of special features (see Section 5).
3. The arcs may only be labeled by members of a fixed set of relations and functions, which belong to a metalanguage with regard to the conceptual level. The relations and functions are exemplarily described in Section 4 and summarized in Appendix A (their full specification can be found in Part II of (Helbig, 2001)).
4. MultiNet distinguishes an **intensional layer** from a **preextensional layer** where the latter is modelling the extension of the first (if the concepts involved can be extensionally interpreted at all). It should be emphasized that certain aspects of the extensional meaning of concepts have to be modeled in the knowledge representation itself (not entirely outside of it as it is the case with logically oriented KRS) to deal properly with such expressions like "*the one ... and the others*", "*three of them*", etc.
5. The whole knowledge about a certain concept C represented by a node N_C is enclosed in a conceptual capsule which is divided into three parts described by the layer feature K-TYPE with the values *categ*, *proto*, and *situa*, respectively (see Figure 1):
   - Component K: This part comprises all arcs connected to N_C which represent categorical knowledge about C. This knowledge, which is marked by the feature value [K-TYPE=*categ*], is valid without any exceptions and is connected with monotonic methods of reasoning.
     Example: "*Every house has a roof*" is categorical knowledge with respect to the concept "*house*"[1].
   - Component D: This component characterizes the prototypical knowledge, which has to be considered as a collection of default assumptions about C. This type of knowledge is characterized by the value [K-TYPE=*proto*] and is connected with methods of nonmonotonic reasoning.
     Example: "*A house (typically) has several windows.*"
   - Component S: Arcs of the SN starting or ending in a node N_C which have no influence on the basic meaning of the corresponding concept C constitute the situational knowledge about C. They indicate the participation of concept C in certain situations. This type of knowledge is characterized by
     [K-TYPE=*situa*].
     Example: "*John's house had been damaged by an earthquake.*"

[1] It should be remarked that the relation POSS starting from the concept "*John*" on the left-hand side in Figure 1 has to be characterized as categorical with regard to the node "*John's house*". Even if in general the possession of things is changing situationally, a house which is not owned by John can not be characterized as "*John's house*". An individual concept like "*John's house*" generally does not have a default part of knowledge. This can be only the case, if it is inherited from general concepts (in this case from the concept "*house*" from which it is known that a house (typically) has several windows; but there are also storehouses without any windows).

Categorical knowledge and prototypical knowledge together form the **immanent knowledge** which – in contrast to the situational knowledge – characterizes a concept inherently. The distinction between immanent and situational knowledge in MultiNet roughly corresponds to the distinction between definitional and assertional knowledge met in other papers (e.g. in (Allgayer and Reddig, 1990)).

6. The relations and functions (which are labels of the arcs at the concept level) are themselves nodes at a metalevel. They are interconnected by means of axiomatic rules (meaning postulates), which are the foundation for the inference processes working over a MultiNet knowledge base. The signatures (i.e. the domains and value restrictions) of relations and functions are defined by means of the sorts mentioned in point 2.

MultiNet has been used and is being used as a meaning representation formalism in several projects (one example is the "Virtual Knowledge Factory" (Knoll et al., 1998)). The most important application at the moment is its use as an interlingua for representing the semantic structure of user queries in natural language interfaces to information providers in the Internet and to dedicated local data bases (Helbig et al., 2000), (Helbig et al., 1996). For that purpose, transformation modules have been developed which translate the semantic structure of these queries formulated by means of the MultiNet language into the Internet protocol Z39.50 and into SQL, respectively.

3 Sorts and Features of Concepts

The classification of nodes, i.e. of the semantic representatives of concepts, into sorts is an important basis for the definition of the domains and value restrictions of relations and functions establishing the interconnections between nodes in a semantic network (see Section 4). The upper part of the conceptual ontology used in MultiNet is shown in Figure 2. The sorts being characterized by this ontology are not only crucial for the formal definition of the representational means, they are also an important source for the semantic interpretation of natural language constructs (e.g. prepositional phrases). This especially holds for the semantic disambiguation of relations underlying a natural language construct, since not all relations may connect a certain pair of conceptual nodes. This decision is supported by the specification of the signatures of the relations involved in the semantic representation of the natural language construct that has to be interpreted (see Appendix A).

Example: In the phrase "*The holidays in the spring*", the preposition "*in*" must be interpreted by the temporal relation TEMP (and not for instance by a local relation), since the semantic representative of the phrase "*in the spring*" bears the sort t (a temporal interval).

From the point of view of the syntactic-semantic analysis of natural language expressions the sorts described above are not sufficient to specify the selectional

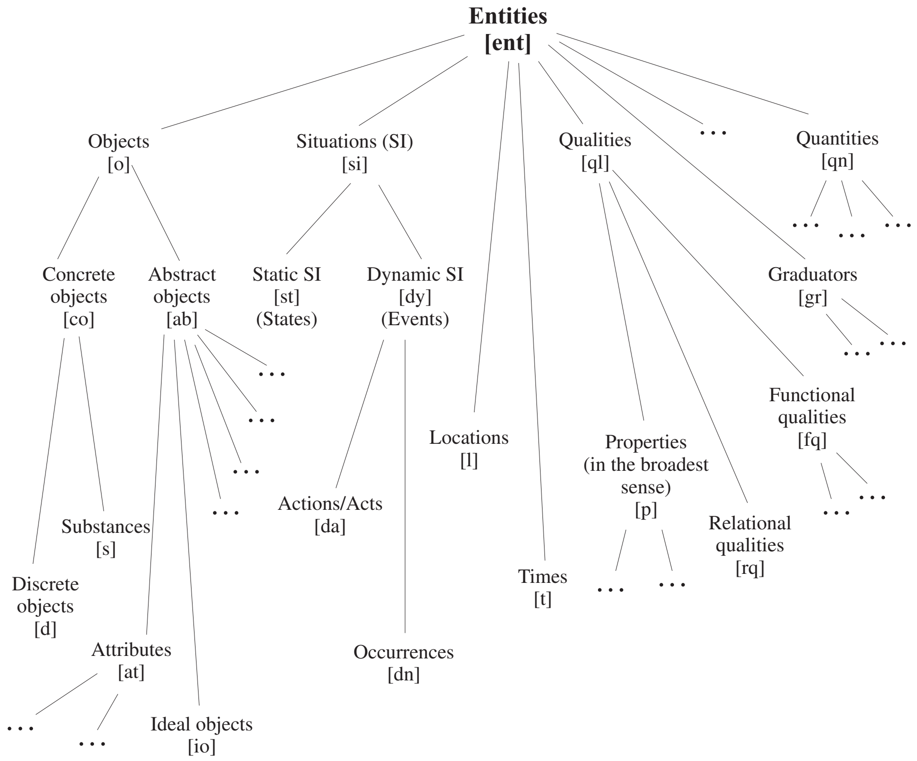


Fig. 2. The Upper Part of the Hierarchy of MultiNet Sorts

restrictions or valencies connected with certain words (especially with verbs). For that we need additional features, like being animated (feature: [ANIM +]), being an artifact (feature: [ARTIF +]), having a distinguished axis (feature: [AXIAL +]), being a geographical object (feature: [GEOGR +]), being movable (feature: [MOVABLE +]), and others. Actually, these features can be described by other expressional means of MultiNet too (like the subordination of concepts or the assignment of properties to objects). However, because of their importance for the description of valencies in a computer lexicon and their prominent role in finding the proper constituents filling the valencies during syntactic-semantic analysis the semantic features have been given a special status. As representational means of the lexicon, they are at the same time marking the interface between lexical knowledge and world knowledge. A complete description of the system of sorts and features connected with MultiNet can be found in (Helbig, 2001). The restrictions imposed by them on the specification of relations and functions or on the valency frames of verbs, nouns, and adjectives are automatically observed in the workbenches for the knowledge engineer and for the computer lexicographer, respectively (see Section 6).

4 Relations and Functions of MultiNet

The formal devices for interlinking the conceptual nodes of a semantic network are relations and functions which are properly described in MultiNet by means of the following characteristic components (for a typical example see Figure 3):

- A short caption with a name as expressive as possible
- The algebraic signature of the relation or function leaning on the MultiNet hierarchy of sorts
- A verbal characterization of the relation or function
- A mnemonic hint supporting the communicability
- Patterns of queries aiming at the relation
- A detailed description showing how to use the relation or function and what logical axioms define the inferential properties of them.

MultiNet provides about 110 semantic primitive relations and functions which can roughly be classified into the following groups:

- Relations and functions of the intensional level. They are used to describe conceptual objects and situations with their inner structure and their relationships to other entities. Typically for the description of objects are the characterization of their material structure (by the part-whole relationship, relation PARS, or by their material origin, relation ORIGM) or their qualitative characterization (by means of properties, relation PROP, or attribute value specifications, relations ATTR, VALR, VAL) and others. It is typical for the description of situations to characterize them by means of the roles the participants in these situations are playing (expressed by deep case relations like agent, relation AGT, or experiencer, relation EXP, etc.) Additionally, they are characterized by their spatio-temporal embedding (by means of local relations, like LOC or DIRCL, or by the temporal relations like TEMP or ANTE). The representational means of the intensional level are briefly described in an overview shown in Appendix A.
- Lexical relations. They describe connections between generic concepts and play an important role in the specification of lexical entries (whence their name). To this group belong the relations specifying synonyms or antonyms, converse concepts and complementary concepts, etc.). To this group also belong the relations characterizing a change of sorts from one concept to a related concept (like the relation CHEA between an event, e.g. “produce”, with [SORT=dy], and an abstract situation, e.g. “production”, with [SORT=ab]).
- Relations and functions of the preextensional level. They characterize the necessary modelling of sets and extensional representatives, which have to be included in the knowledge representation itself to deal properly with the meanings of constructs involving sets (like “most of them”).

Relations and functions connecting nodes at the conceptual level can themselves be seen as nodes of a metalevel, which are connected by axioms written in the form of predicate calculus expressions (to be more exact, in the form of implications). In that, we discern two types of axioms:

- **Title:** Causality, Relation between Cause and Effect
- **Signature:** $[\mathrm{si}' \cup \mathrm{abs}'] \times [\mathrm{si}' \cup \mathrm{abs}']$ (for sorts, see Figure 2)
- **Verbal Description:** The relation (s_1 CAUS s_2) indicates that the real situation s_1 is the cause for the real situation s_2. (The value [FACT = *real*] is symbolized by a prime at the corresponding symbol.) s_2 is the effect which is actually brought about by s_1. The relation CAUS is transitive, asymmetric, and not reflexive.
- **Mnemonics:** (x CAUS y) – [x is the cause of y]
- **Query patterns:**
  {Why/How is it that} ⟨s_2⟩?
  {Of what/From which ⟨s_1⟩} {[die]/[suffer]/[fall ill]/. . . } ⟨d⟩?
  By what [being caused] ⟨s_2⟩?
  What is the cause for ⟨s_2⟩?
  Which effect {does/did} ⟨s_1⟩ have?
  {Thanks to/Because of} ⟨WH⟩ ⟨s_1⟩ {[happen]/[occur]/. . . } ⟨s_2⟩?
- **Commentary:** The causal relationship is closely connected to the temporal successor relation ANTE, since the effect can not take place before the cause:
  - (x CAUS y) → ¬(y ANTE x)

  There exists also a connection between the relations CSTR and CAUS:
  - (s_1 CSTR d) → ∃ s_2 ([(s_2 AGT d) ∨ (s_2 INSTR d)] ∧ (s_2 CAUS s_1))

  The following example sentences are typical for the causal relation. The third of them shows clearly that the relation CAUS – in contrast to COND and IMPL – always connects real (not hypothetical) situations, which are characterized by the attribute value [FACT = *real*] .

  [The excitement]$^{\mathrm{CAUS}_{arg2}}$ about [the strange event]$^{\mathrm{CAUS}_{arg1}}$.
  [Peter suffers]$^{\mathrm{CAUS}_{arg2}}$ [from gastritis]$^{\mathrm{CAUS}_{arg1}}$.
  Because [Peter went carelessly across the street,]$^{\mathrm{CAUS}_{arg1}}$
  [he had been run over by a car]$^{\mathrm{CAUS}_{arg2}}$.

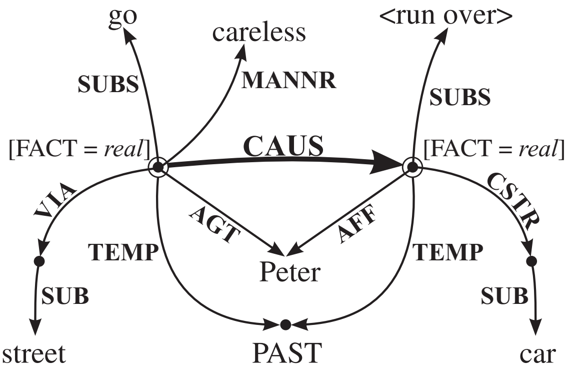


Fig. 3. Description of the causal relationship

- B-axioms: They connect relations and functions with representatives of natural language concepts. As an example, we give the axiom stating that the part has a weight minor to that of the whole:
  $(k_1$ PARS $k_2) \wedge (k_2$ ATTR $m_2)$
  $\wedge\ (m_2$ SUB weight$) \wedge (m_2$ VAL $q_2) \longrightarrow$
  $\exists m_1\ \exists q_1$: $[(k_1$ ATTR $m_1) \wedge (m_1$ VAL $q_1)$
  $\wedge\ (m_1$ SUB weight$) \wedge (q_1$ MIN $q_2)]$
- R-axioms: They connect relations and functions with each other and do not contain natural language concepts. Example:
  Inheritance of the part-whole relationship:
  (d1 SUB d2) ∧ (d3 PARS d2) ⟶
  ∃d4 [(d4 SUB d3) ∧ (d4 PARS d1)]

An overview of the different types of axioms used in MultiNet for the formal specification of relations and functions can be found in appendix E of (Helbig, 2001).

5 The Stratification of the Semantic Network

One aim of the MultiNet design has been to distance oneself from those network paradigms that press qualitatively entirely different aspects of meaning into one flat structure. For this purpose, the nodes and arcs of MultiNet are embedded in a multidimensional space of so-called layer attributes. The layer specifications for arcs are comprised into an attribute K-TYPE (see point 5, Section 2) and for nodes into another attribute LAY (see Figure 4).

The values of K-TYPE help to distinguish the immanent from the situational knowledge in the semantic network, as discussed in Section 2. The specifications for the attribute LAY are organized along several dimensions, which can itself be described by special attributes having their own values :

- GENER: The **degree of generality** indicates whether a conceptual entity is generic (value: *ge*) or specific (value: *sp*).
  Example: "*(A car)* [GENER=*ge*] *is a useful means of transport.*"
  "*(This car)* [GENER=*sp*] *is a useful means of transport.*"
- FACT: This attribute describes the **facticity** of an entity, i.e. whether it is really existing (value: *real*), not existing (value: *nonreal*), or only hypothetically assumed (value: *hypo*).
  Example: "*John* [FACT=*real*] *thought that (he was ill)* [FACT=*hypo*]."
  "*John* [FACT=*real*] *remembered that (he was ill)* [FACT=*real*]."
- REFER: This attribute specifies the **determination of reference**, i.e. whether there is a determined object of reference (value: *det*) or not (value: *indet*). This type of characteristic plays an important part in natural language processing in the phase of knowledge assimilation and especially for the resolution of references.
  Example: "*(The passenger)* [REFER=*det*] *observed (an accident).* [REFER=*indet*]."

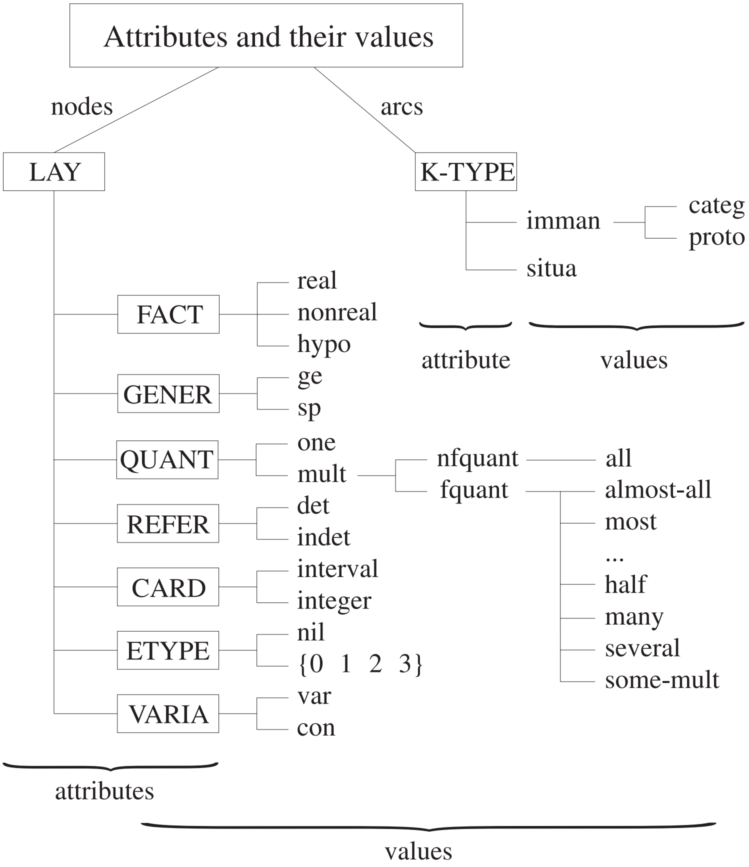


Fig. 4. The multidimensional space of layer attributes

- QUANT: The intensional **quantification** represents the quantitative aspect of a conceptual entity (whether it is a singleton (value: *one*) or a multitude (values: *two*, *three*, . . . *several*, *many*, . . . *most*, *all*)). Within the set of values characterizing multitudes, we discern between fuzzy quantifiers like *several*, *many*, . . . *most* with value [QUANT = *fquant*] and non-fuzzy quantifiers like *two*, *three*, . . . , *all* with value [QUANT = *nfquant*].
  Example: "*(Some house)* [QUANT = *one*] *had been destroyed.*"
- ETYPE: This attribute characterizes the **type of extensionality** of an entity with values: *nil* – no extension,
  0 – individual with an extension that is not a set (e.g. Elizabeth I),
  1 – entity with a set of elements from type [ETYPE=0] as extension (e.g.⟨many houses⟩, ⟨the family⟩),
  2 – entity with a set of elements from type [ETYPE=1] as extension (e.g. ⟨many families⟩), etc.

- CARD: The **cardinality** as characterization of a multitude at the preextensional level is the counterpart of the attribute QUANT at the intensional level. Thus, the intensional characterization ⟨several members of the crew⟩ sometimes can be made more precise by specifying a concrete cardinality (maybe [CARD=4]) or at least an interval (lets say [CARD=(5, 7)]) for the underlying extension on the basis of additional knowledge or on the basis of a referring expression (e.g. "*six of them ...*").
  Example: "*(A group)* [CARD=1] *of four archaeologists discovered (many tablets)$_i$.*
  Six of (*them* [CARD>6])$_i$ *had been spoiled by the transport.*"
- VARIA: The **variability** finally describes whether an object is conceptually varying (value: *var*) – i.e. it is a so-called parameterized object – or not (value: *con*).
  Example: "*(This policeman)* [VARIA=*con*] *checked (every passport)* [VARIA =*var*]."

The idea of layers is motivated by an analogy to the mathematics of an n-dimensional space. If one fixes a value along one of the axes of an n-dimensional coordinate system, one gets a (n-1)-dimensional hyperplane. In the same way, if one is fixing one value of a layer attribute (let us assume [GENER=*ge*] or [FACT=*hypo*]), then one gets a special layer or stratum (in this case the layer of all generic concepts or the layer of all hypothetical entities, respectively). Analogously, fixing the value [K-TYPE=*imman*] yields the immanent knowledge about all conceptual entities stored in the knowledge base.

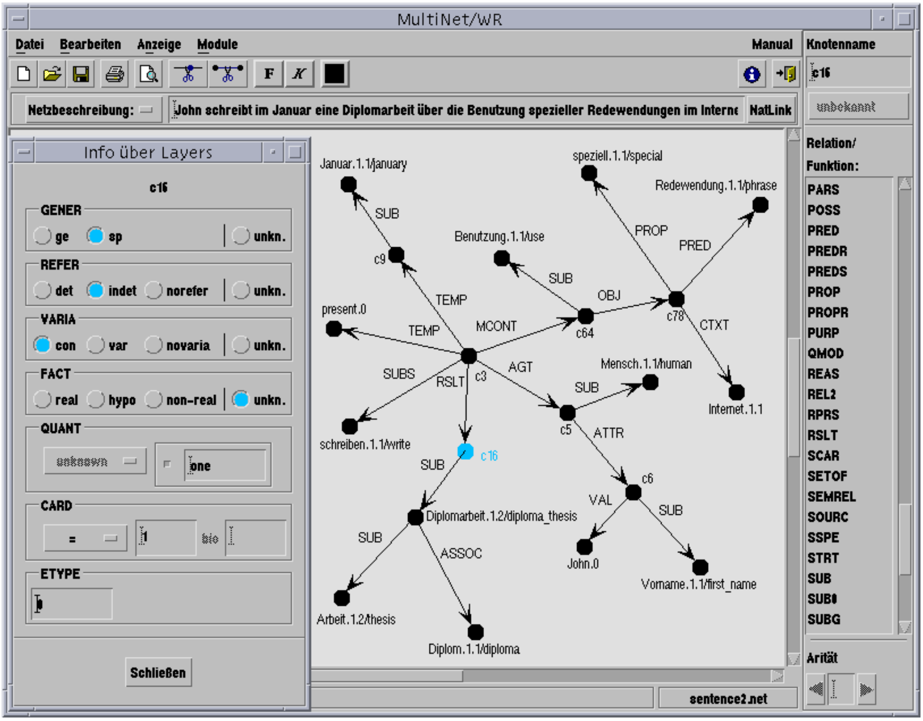
6 The Software Tools Connected with the MultiNet Paradigm

To support the effective work with MultiNet and the generation of large stocks of information on the basis of this knowledge representation paradigm, MultiNet has been provided with several software tools (a guided tour through these tools can be found on the CD attached to (Helbig, 2001) or at the Internet site *http://pi7.fernuni-hagen.de/research/*):

Multinet/WR: A workbench for the knowledge engineer supporting the graphical representation and manipulation of MultiNet networks as well as the accumulation and management of MultiNet knowledge bases. This tool has been developed by Carsten Gnörlich (Gnörlich, 2000).

NatLink: An interpreter which translates natural language sentences into MultiNet semantic networks by means of a word-class controlled syntactic-semantic analysis. NatLink has been developed by Sven Hartrumpf (Helbig and Hartrumpf, 1997).

LIA: An interactive workbench for the computer lexicographer which is used to create large semantically oriented computer lexica based on the expressional means of MultiNet. The workbench LIA was initially developed by Marion Schulz (Schulz, 1999) and is now being redesigned and newly developed by Rainer Osswald.



MultiNet/WR

Netzbeschreibung: John schreibt im Januar eine Diplomarbeit über die Benutzung spezieller Redewendungen im Interne

Knotenname: c16

unbekannt

Info über Layers

c16

GENER: ○ ge ● sp ○ unkn.

REFER: ○ det ● indet ○ norefer ○ unkn.

VARIA: ● con ○ var ○ novaria ○ unkn.

FACT: ○ real ○ hypo ○ non-real ● unkn.

QUANT: unknown one

CARD: = 1 bis

ETYPE: 0

sentence2.net

Fig. 5. The manipulation and representation of semantic networks with MultiNet/WR

Figure 5 presents a snapshot of the work with the software tool MultiNet/WR showing the semantic representation of the sentences:
(S-G) German: "*John schreibt im Januar eine Diplomarbeit über die Benutzung spezieller Redewendungen im Internet.*"
(S-E) English: "*In January, John writes a diploma thesis about the use of special phrases in the Internet.*"
NatLink can be activated directly from MultiNet/WR (button on the top, right-hand side) taking the sentence presented in the field to the left of this button as input. The result of the analysis is automatically written back on the main working panel of MultiNet/WR. Thus, the basic structure of the network had been automatically created by means of NatLink on the basis of the sentence (S-G). (The English translation (S-E) of the sentence has the same semantic structure as it can be seen from Figure 5. Since NatLink is working at the moment for German only, the labels at the terminal nodes have been added manually by means of MultiNet/WR.)

The networks shown at the working panel of MultiNet/WR can be further edited and manipulated by carrying out several operations:

- Changing the topology of the network by inserting and deleting nodes and arcs
- Changing the layout by moving the nodes and edges, or changing the labels of nodes and arcs, or accessing additional information like viewing and editing the sort or layer information of an activated node (see the pop-up menu at the left side in Figure 5 for the activated node c16 showing its layer specification). Additionally, a complex help system provides the documentation for most elements shown in the working panel, including the explanation of the relations or functions labelling an activated arc (cf. Figure 3 showing the explanation coming up if the help system is activated for an arc labelled by the relation CAUS).

There are also more complicated operations, which can be evoked by means of the buttons at the top bar. They include among others:

- Concatenation of different networks to assimilate them into one knowledge base
- Checking the formal consistency of the network
- Initiating pattern matching processes and inference processes over the semantic network
- Transforming the deep structure of natural language queries into the retrieval language of a data base management system (e.g. into SQL).

7 Conclusion

MultiNet is one of the few knowledge representation systems along the line of semantic networks with a comprehensive, systematic and publicly available description (Helbig, 2001). It has been practically applied in several projects like natural language access to digital libraries in the Internet or as a conceptual interface for information retrieval in multimedia data bases (Knoll et al., 1998). This knowledge representation paradigm is connected with a collection of software tools supporting its use in different application domains. Since MultiNet has been designed as a system for the semantic representation of natural language information, it is especially appropriate for being used as an interlingua in natural language processing systems, which has been proven by the successful application of MultiNet in the above mentioned projects.

References

Allgayer, J. and Reddig, C. (1990). What KL-ONE lookalikes need to cope with natural language – scope and aspect of plural noun phrases. In *Sorts and Types in Artificial Intelligence* (edited by Bläsius, K.; Hedstück, U.; and Rollinger, C.-R.), pp. 240–285. Berlin, Germany: Springer.

Baader, F.; Molitor, R.; and Tobies, S. (1998). On the relation between conceptual graphs and description logics. Technical Report LTCS-Report 98-11, Aachen University of Technology, Aachen, Germany.

Brachman, R. (1978). Structured inheritance networks. Technical Report No. 3742, Bolt Beranek & Newman, Cambridge, Massachusetts.

Gnörlich, C. (2000). MultiNet/WR: A Knowledge Engineering Toolkit for Natural Language Information. Technical Report 278, University Hagen, Hagen, Germany.

Helbig, H. (2001). *Die semantische Struktur natürlicher Sprache: Wissensrepräsentation mit MultiNet.* Berlin: Springer.

Helbig, H.; Gnörlich, C.; and Leveling, J. (2000). Natürlichsprachlicher Zugang zu Informationsanbietern im Internet und zu lokalen Datenbanken. In *Sprachtechnologie für eine dynamische Wirtschaft im Medienzeitalter* (edited by Schmitz, K.-D.), pp. 79–94. Wien: TermNet.

Helbig, H.; Gnörlich, C.; and Menke, D. (1996). Realization of a user-friendly access to networked information retrieval systems. Informatik-Bericht 196, FernUniversität Hagen, Hagen, Germany.

Helbig, H. and Hartrumpf, S. (1997). Word class functions for syntactic-semantic analysis. In *Proceedings of the 2nd International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, pp. 312–317. Tzigov Chark, Bulgaria.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Number 42 in Studies in Linguistics and Philosophy. Dordrecht: Kluwer Academic Publishers.

Knoll, A.; Altenschmidt, C.; Biskup, J.; Blüthgen, H.-M.; Glöckner, I.; Hartrumpf, S.; Helbig, H.; Henning, C.; Karabulut, Y.; Lüling, R.; Monien, B.; Noll, T.; and Sensen, N. (1998). An integrated approach to semantic evaluation and content-based retrieval of multimedia documents. In *Proceedings of the 2nd European Conference on Digital Libraries (ECDL'98)* (edited by Nikolaou, C. and Stephanidis, C.), number 1513 in Lecture Notes in Computer Science, pp. 409–428. Berlin: Springer.

Peltason, C. (1991). The BACK system –An overview. *SIGART Bulletin*, 2(3):114–119.

Quillian, M. R. (1968). Semantic memory. In *Semantic Information Processing* (edited by Minsky, M.), pp. 227–270. Cambridge, Massachusetts: MIT Press.

Schulz, M. (1999). *Eine Werkbank zur interaktiven Erstellung semantikbasierter Computerlexika.* Ph.D. thesis, FernUniversität Hagen, Hagen, Germany.

Shapiro, S. C. (1999). SnePS: A logic for natural language understanding and commonsens reasoning. In *Natural Language Processing and Knowledge Representation: Language for Knowledge und Knowledge for Language* (edited by Iwanska, L. and Shapiro, S.). Cambridge, Mass.: The MIT Press.

Appendix A: Specification of the Relations and Functions of MultiNet and Their Signatures

(This table does not include the lexical relations, the metarelations, and the representational means of the preextensional level. The complete hierarchy of sorts can be found in (Helbig, 2001).)

Relation	**Signature**	**Short Characteristics**
AFF	$[si \cup abs] \times [o \cup st]$	C-Role – Affected object
AGT	$[si \cup abs] \times o$	C-Role – Agent
ANLG	$[\ddot{si} \cup \ddot{o}] \times at$	Similarity relation
ANTE	$tp \times tp$	Temporal succession
ANTO	$sort \times sort$	Antonymy relation
ASSOC	$ent \times ent$	Relation of association
ATTCH	$[o \setminus at] \times [o \setminus at]$	Attachment of objects
ATTR	$[o \cup l \cup t] \times at$	Specification of an attribute
AVRT	$[dy \cup ad] \times o$	C-Role: Turning away
BENF	$[si \cup abs] \times [o \setminus abs]$	C-Role – Beneficiary
CAUS	$[si^{'} \cup abs^{'}] \times [si^{'} \cup abs^{'}]$	Relation between cause and effect (Causality)
CIRC	$si \times [ab \cup si]$	Relation between situation and circumstance
CONC	$[si \cup abs] \times [si \cup ab]$	Concessive relation
COND	$\tilde{si} \times \tilde{si}$	Conditional relation
CONF	$si \times [ab \cup si]$	External frame, to which a situation conforms
CORR	$sort \times sort$	Relation of qualitative or quantitative correspondence
CSTR	$[si \cup abs] \times o$	C-Role – Causator
CTXT	$[si \cup abs] \times [o \cup si]$	Relation specifying a restricting context
DIRCL	$[si \cup o] \times [l \cup o]$	Relation specifying a local aim or a direction
DISTG	$[\ddot{si} \cup \ddot{o}] \times at$	Distinction between entities
DUR	$[si \cup o] \times [t \cup si \cup abs \cup tn \cup qn]$	Relation specifying a duration
EQU	$sort \times sort$	Equality/equivalence relation
EXP	$[si \cup abs] \times o$	C-Role – Experiencer
FIN	$[t \cup si \cup o] \times [t \cup ta \cup abs \cup si]$	Relation specifying the temporal end
GOAL	$[si \cup o] \times [si \cup o \cup t]$	Generalized goal
IMPL	$[si \cup abs] \times [si \cup abs]$	Implication between situations
INIT	$[dy \cup ad] \times [o \cup si]$	Relation specifying an initial state
INSTR	$[si \cup abs] \times co$	C-Role – Instrument
JUST	$[si \cup abs] \times [si \cup abs]$	Relation specifying a justification
LEXT	$[si \cup o] \times [l \cup m]$	Relation specifying a local extension

Relation	**Signature**	**Short Characteristics**
LOC	$[o \cup si] \times l$	Relation specifying the location of a situation
MAJ{E}	$qn \times qn$	Greater-then-[or equal]
MANNR	$si \times [ql \cup st \cup as]$	Relation specifying the manner of existence of a situation
MCONT	$[si \cup o] \times [\tilde{o} \cup \tilde{si}]$	C-Role – Relation between a mental process and content
METH	$[si \cup abs] \times [dy \cup ad \cup io]$	C-Role – Method
MEXP	$[st \cup abs] \times d$	C-Role – Mental carrier of a state
MIN{E}	$qn \times qn$	Smaller-then-[or equal]
MODE	$[si \cup abs] \times [o \cup si \cup ql]$	Generalized mode of an occurrence
MODL	$\tilde{si} \times md$	Relation specifying a restricting modality
NAME	$ent \times fe$	Relation specifying the name of an object
OBJ	$si \times [o \cup si]$	C-Role – Neutral object
OPPOS	$[si \cup abs] \times [si \cup o]$	C-Role – Entity being opposed by a situation
ORIG	$o \times [d \cup io]$	Relation specifying an informational source
ORIGL	$[o \cup si] \times [l \cup o]$	Local origin
ORIGM	$co \times co$	Material origin
ORNT	$[si \cup abs] \times o$	C-Role – Orientation towards something
PARS	$[co \times co] \cup [io \times io] \cup [t \times t] \cup [l \times l]$	Part-whole-relationship
POSS	$o \times o$	Relation between possessor and possession
PRED	$[\ddot{o} \setminus a\ddot{b}s] \times [\overline{o} \setminus \overline{abs}]$	Predicative concept characterizing a plurality
PROP	$o \times p$	Relation between object and property
PROPR	$\ddot{o} \times rq$	Relation between a plurality and a semantic relational quality
PURP	$[si \cup o] \times [si \cup ab]$	Relation specifying a purpose
QMOD	$[s \cup \ddot{d}] \times m$	Quantitative specification
REAS	$[si \cup abs] \times [si \cup abs]$	Generalized reason
RPRS	$o \times o$	Relation specifying a manifestation of an object
RSLT	$[si \cup abs] \times [o \cup si]$	C-Role – Result
SCAR	$[st \cup as] \times o$	C-Role – Carrier of a state
SOURC	$[si \cup o] \times [si \cup o \cup l]$	Generalized source
SSPE	$[st \cup as] \times ent$	C-Role – Entity specifying a state
STRT	$[si \cup o \cup t] \times [t \cup ta \cup abs \cup si]$	Relation specifying a temporal begin

Relation	Signature	Short Characteristics
SUB	$[o \setminus \{abs \cup re\}] \times [\overline{o} \setminus \{\overline{abs} \cup \overline{re}\}]$	Relation of conceptual subordination (for objects)
SUBS	$[si \cup abs] \times [\overline{si} \cup \overline{abs}]$	Relation of conceptual subordination (for situations)
SUBST	$[o \times o] \cup [si \times si]$	Relation specifying a representative
SUPPL	$[si \cup abs] \times o$	Supplement relation
TEMP	$[si \cup t \cup o] \times [t \cup si \cup abs \cup ta]$	Temporal embedding of a situation
VAL	$\dot{at} \times [o \cup qn \cup p \cup fe \cup t]$	Relation between attribute and its value
VALR	$\overline{at} \times o$	Relation between attribute and its value restriction
VIA	$[d \cup dy \cup ad] \times [l \cup d]$	Relation specifying a path

Function	Signature	Short Characteristics
*COMP	$gq \times o \rightarrow tq$	Function describing the comparison of properties
*FLP$_J$	$d \times l$	Family of functions generating locations
*ITMS	$pe^{(n)} \times \ldots \times pe^{(n)} \rightarrow pe^{(n+1)}$	Function enumerating a set
*MODP	$[p \cup m \cup lg] \times p \rightarrow q$	Modification of properties
*MODQ	$ng \times qf \rightarrow qf$	Function modifying quantities
*MODS	$[gr \cup m] \times dy \rightarrow dy$	Function modifying actions
*NON	$md \rightarrow md$	Metafunction for representing negation
*OP$_J$	$qn^w \rightarrow qn$	Arithmetic and other mathematical operations
*ORD	$nu \rightarrow oq$	Function specifying ordinal numbers
*PMOD	$aq \times o \rightarrow o$	Modifcation of objects with properties
*QUANT	$qf \times me \rightarrow m$	Generation of quantities
*SUPL	$gq \times [\overline{o} \cup \dot{o}] \rightarrow tq$	Function describing the superlative
*TUPL	$sort \times \ldots \times sort \rightarrow sort$	Function generating a tuple from its components

The sort symbols can be marked by the following signs:
$\tilde{o}$ – hypothetical entity with [FACT *hypo*];
$\overline{o}$ – generic concept with [GENER *ge*];
$\dot{o}$ – individual concept with [GENER *sp*].

Constructing a Sensuous Judgment System Based on Conceptual Processing

Atsushi Horiguchi[1], Seiji Tsuchiya[1], Kazuhide Kojima[1], Hirokazu Watabe[2], and Tsukasa Kawaoka[2]

[1] Department of Knowledge Engineering and Computer Sciences,
Graduate School of Engineering, Doshisha University,
1-3 Miyakodani, Tatara, Kyotanabe, Kyoto, 610-0394, Japan
{dtb0711,dta0732,eta1702}@mail4.doshisha.ac.jp
[2] Department of Knowledge Engineering and Computer Sciences,
Doshisha University, 1-3 Miyakodani, Tatara, Kyotanabe, Kyoto, 610-0394, Japan
{watabe,kawaoka}@indy.doshisha.ac.jp

Abstract. When we humans receive uncertain information, we interpret it suitably, to understand what the speaker is trying to say. This is possible because we have "commonsense" concerning the word, which is built up from knowledge that is stored from long time experience. In order to understand the meaning, we think that the construction of a "Commonsense Judgment System" is necessary. Of the commonsense we use in our every day lives, one concerns the characteristics of words, such as an apple is red. This paper proposes a mechanism to associate the characteristics of a word based on our five senses with a knowledge base consisting of basic words. By using a concept-base, which is automatically constructed from dictionaries, this mechanism can understand a word that does not exist in the knowledge base. This study aims to retrieve the meaning concerning sense, and deepen semantic understanding.

1 Introduction

In the future, it is thought that the computer needs to judge with more flexibility through interactive communication with humans. To do so, the meaning of the conversation must be understood. The mechanism of communication between human beings must be modelized and put into the interface of computers and human beings.

When we receive uncertain information, assuming appropriate circumstances we interpret it properly (or close to properly), in order to understand what the speaker is trying to say. This is possible because we have "commonsense" concerning the word, which is built up from the knowledge of our language that is stored from long time experience. With this knowledge of the commonsense, we are able to understand the meaning of what is being said.

Of the commonsense we use in our ever day lives, we think that there are the commonsense relevance to quantity (such as size, weight, or speed), characteristics (such as type of shape, color, scent, or taste), and emotion (such as happy or sad). In order to understand meaning, we think that the construction of a "Commonsense Judgment System" is necessary. An element technology needed for this system is the

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 86–95, 2002.

judgment concerning commonsense knowledge based on our five senses (sense of sight, hearing, smell, taste, and touch). In interactive conversation, we unconsciously associate a word's characteristics. For example, when we hear "her cheeks are like apples," we interpret that "her cheeks are red as apples." By doing so, we are able to understand the meaning of what is being said.

In this paper, a mechanism to associate a word with all its possible characteristics based on our five senses using a knowledge base of common words is proposed. Also, a method to output the characteristics of words not in the knowledge base, by use of the concept-base and association between concepts, is proposed.

2 Sensuous Judgment

In interactive communication, we understand then judge what the speaker is trying to say, and through the conversation, new knowledge is gained. This process is repeated when we communicate. During a conversation, we unconsciously associate a word's characteristics, understand it, and by doing so, the conversation continues smoothly.

For example, when somebody says, "In the summer, I want ice cream," we know by commonsense that summer is a hot season and ice cream is a cold food. By associating its characteristics, we can understand that the speaker wants cold food during a hot season, and therefore we can judge that the speaker is not saying something that does not makes sense.

It can be seen that the knowledge of commonsense concerning our five senses is important for understanding meaning. We call associating its characteristics from a word, "Sensuous Judgment", and we aim to judge a word's characteristics without contradiction within the limits of commonsense. The subject of this study is limited to Sensuous Judgment of common nouns.

2.1 Sense

A characteristic of a noun is an adjective, and therefore the result of Sensuous Judgment is in the form of an adjective. For example, apples are red, round, and sweet. Of the few thousand adjectives in the Japanese language, the uncommon ones were removed, the ones related to our five senses were chosen, and synonyms were grouped, leaving 109 adjectives, which are called "senses," such as red, loud, fragrant, delicious, or hot. These senses are classified into two levels according to our five senses. For example, red is an adjective related to "the sense of sight" and is a "color". Therefore, the sense "red" is classified "red : color : sight." This study aims to build a Sensuous Judgment mechanism, which connects a noun with its senses. (Example shown in Table 1)

Table 1. Relation of a word and its sense. The senses are classified into detailed groups

word	sense		
apple	red	color	sight
stove	hot	temperature	touch

3 Sensuous Knowledge Base

In order to realize a Sensuous Judgement System, knowledge of the relation between a word and its characteristics is needed. But to store the knowledge of the relation of all words is very difficult and inefficient. Therefore, only the relation between 685 commonly used nouns, "representative words," and the 109 senses were put into the knowledge base, such as apple and red, or summer and hot. The Sensuous Kwoledge Base consisits of a total of 1305 relations of a representative word and a sense.

Other than the knowledge of the relations, the Sensuous Knowledge Base holds detailed information of the senses and representative words. Concerning the senses, classification information of the sense, and synonyms of the 109 senses in the forms of adjectives, nouns, and verbs are held as knowledge. It is possible to pick up words concerning sense other than the representing 109 adjectives with the synonyms.

Concerning the representative words, classification information is held as knowledge. The thesaurus [NTT97] was used to classify the representative words, into "classifications," which are nodes in tree structure. The thesaurus shows upper-lower and part-whole relations of 1927 nodes in a tree structure, which classifies general nouns by their usage in meaning. These nodes were used to classify the representative words of the Sensuous Judgment Knowledge Base, so there are a total of 1927 classifications in the knowledge base. Relations between the classification and the corresponding senses are stored in the knowledge base. Senses of the classifications can be uncertain, for example the classification "food," has the sense "smell : scent," implying that food has a scent, but it is not certain what kind of scent it has. Of the 1927 classifications, 153 have sense information. The image figure of the knowledge base is shown in Figure 1.

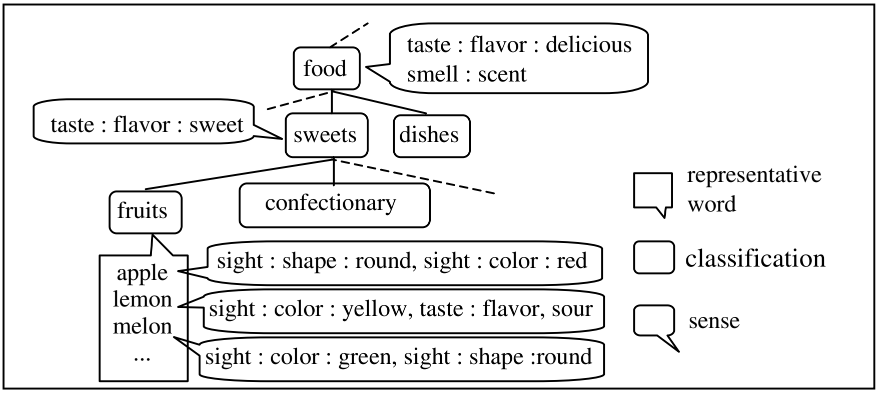

taste : flavor : delicious
smell : scent
food
taste : flavor : sweet
sweets
dishes
fruits
confectionary
apple
lemon
melon
...
sight : shape : round, sight : color : red
sight : color : yellow, taste : flavor, sour
sight : color : green, sight : shape :round
representative word
classification
sense

Fig. 1. Image figure of the Sensuous Judgment Knowledge Base. The senses of the classifications are inherited to the representative words

With this Sensuous Judgment Knowledge Base, it is possible to associate the characteristics of a word, concerning the representative words. The sense of a word with respect to its classification and the particular sense of the word itself can be outputted. For example, the sense of an apple as a fruit is sweet and delicious, and as itself, round and red. These are all senses of an apple.

All the sense information in the Sensuous Judgment Knowledge Base was given by human hand, and was verified. Therefore, the information is accurate. Consequently, Sensuous Judgment concerning the representative words is accurate. The commonsense knowledge enables us to understand the meaning of words concerning the sense.

4 Unknown Words

With the Sensuous Judgment Knowledge Base, the association of the characteristics of a word, concerning the representative words, is possible. But, of the few hundred thousand nouns that exist in the Japanese language, the knowledge base only consists of a very small percentage. Most of the words that appear in conversation are words that are not representative words. Understanding of meaning concerning these words is important, and Sensuous Judgment concerning all words is needed. Words that are not in the knowledge base are called unknown words, and a mechanism to perform Sensuous Judgment concerning unknown words is needed.

To perform Sensuous Judgment on unknown words, the semantic association between words is used. To realize this association, the concept base and the degree of association are used (See section 4.1). In order to use the Sensuous Judgment Knowledge Base concerning unknown words, with the concept base and the degree of association, which make up the concept association mechanism, the semantic relation and the strength of the relation are evaluated and the representative word that has the strongest association is decided (Figure 2). This process, which searches for a representative word to replace the unknown word putting meaning into consideration, is called "Unknown Word Processing. " Since the concept base is used, an unknown word must be a concept of the concept base.

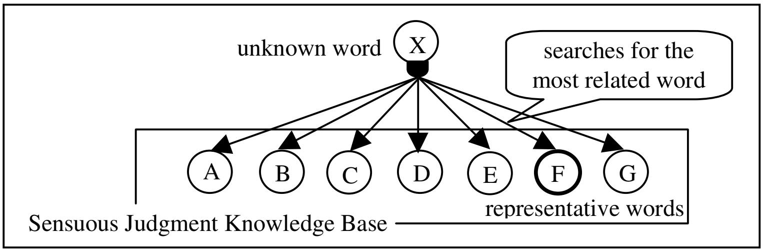

unknown word
X
searches for the most related word
A
B
C
D
E
F
G
representative words
Sensuous Judgment Knowledge Base

Fig. 2. Image Figure of Unknown Word Processing

4.1 Concept Base and the Degree of Association

A word A is defined as a set of words a_i, which have strong semantic relation with the word. A is called a concept, and a_i, its attributes. Every attribute has weight information w_i, representing the strength of how relevantly it explains the concept.

$$A = \{(a_1, w_1), (a_2, w_2), \dots , (a_m, w_m)\} \tag{1}$$

The meaning of a concept is defined by its attributes, and the concept base is made up of many sets of concepts and their attributes.

The concept base [Kasahara97], consisting of 40 thousand concepts, made automatically from several dictionaries, was the basis of the concept base used in this paper [Manabe01], which consists of 120 thousand concepts, where concepts were added and refined (addition and revision of attributes) and the proper weights were decided by rules aiming toward better quality.

The degree of association is a numerical value, which expresses the strength of relation in semantic correspondence between two concepts [Watabe01], unlike in semantic networks, where only the type of relation can be retrieved as information. The degree of association is calculated by the degree of match between 2 concepts, which is a value between 0 and 100. Examples are shown in Table 2.

Table 2. Examples of the Degree of Association

concept	The Degree of Association between the Concept "fruit"
apple	26.074
sun rize	1.833

5 Unknown Word Processing

Concerning representative words, the knowledge base can be used to output the sense of a word as a classification and as itself. When thinking about Unknown Word Processing, a few points must be put into mind. For example, "pear" is an unknown word, so Unknown Word Processing must be performed. Using the association mechanism of the concept base and degree of association, the closet representative word must be found. The word "apple" could be the closest representative word, but an "apple" cannot entirely replace a "pear" concerning the sense. Where the pear has close relation to an apple, in sense, is in its classification as a fruit, which sense is sweet and delicious. The sense of the pear itself is not the same as the sense of the apple itself, such as being red. So the Unknown Word Processing must have two steps: the first to decide the unknown word's classification and output the sense of the classification, and the second to output the sense of the unknown word itself. The first step is called Classification Decision and the second is called Sense Retrieval.

5.1 Classification Decision

When an unknown word is inputted, the unknown word's classification must be decided. The degree of association is used to decide which classification the unknown word belongs to. The degree of association between the unknown word, and 838 words in the Sensuous Judgment Knowledge Base (685 representative words and 153 classifications with senses) is calculated. Statistics is used, and the 838 words are considered the population and the degree of association is converted into the deviation. If the deviation is higher than a certain value, 88, it is judged that the unknown word is in strong relation with that word. If there is more than one word that has a deviation value higher than 88, the unknown word is related to the word of the knowledge base with the highest deviation. The deviation value of 88 was derived from experiments.

By converting the degree of association into deviation, and not just relating the unknown word to the word with the largest value of degree of association, the unknown word can be associated to a word in the Sensuous Judgment Knowledge Base that is thought to have especially strong relation with.

If the unknown word is related to one of the 685 representative words, the classification of the unknown word is decided to be that of the representative word. Some representative words belong to more than one classification, so in such a case, the degree of association between the unknown word and the several classifications is calculated again to decide a single classification with the strongest relation. One classification is decided concerning an unknown word.

If the unknown word is related to one of the 153 classifications, that classification is decided to be the unknown word's classification.

By deciding which classification the unknown word classifies under, it is possible to get its sense concerning the classification. If the Classification Decision is performed properly, it is possible to obtain accurate sense information of the unknown word with the Sensuous Judgment Knowledge Base (See Figure 3).

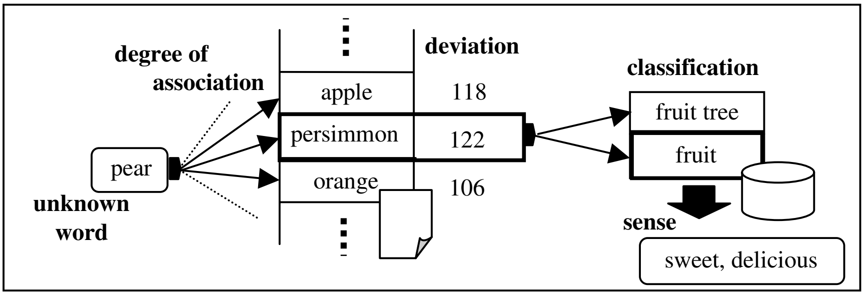


Fig. 3. Example of Classification Decision. The unknown word, "pear" is related to one of the representative words "persimmon," by selecting the word with the highest division over 88, which was calculated from the degree of association. The classification of the unknown word is decided to be "fruit," which was selected by the higher degree of association. The sense of the unknown word "pear" concerning the classification is judged to be sweet and delicious

5.2 Sense Retrieval

By performing Classification Decision, the sense concerning the classification of the unknown word can be obtained, but the sense of the word itself cannot.

The concept base is used to obtain sense of the word itself. The concept base defines a concept with a group of words, its attributes, and that group consists of words that have strong relation to the concept, words that explain the concept, and words that are the characteristics of the concept.

Of the attributes, if there are words that have strong relation to the senses, those words can be considered the sense of the concept. The information of synonyms concerning the senses in the Sensuous Judgment Knowledge Base is used to judge whether or not the attribute expresses a sense. Therefore, not just adjectives but also words in forms of verbs and nouns can be extracted. Considering the unknown word as a concept of the concept base, the sense of the unknown word itself can be obtained. (Example shown in Table 3)

Table 3. Example of extracting words related to senses from the attribute of a concept. The high lighted words can be considered the sense of the concept. In this example, the sense of the concept " panda" would be black, white, and large

Concept	Attributes
Panda	bear, animal, white, lion, wild, live, Tibet, stuffed animal, feet, black, mountain, China, big, woods, bamboo, giant, beast, like, ...

Senses can be obtained from the attributes of the concept, but the sense might not characteristize the concept. Since attributes are words that have close relation to the concept, the attribute does not necessarily characterize the concept. Problems that occur are the following.

1. There are times when a pair of senses of the opposite meaning are both attributes of a concept.
   ex) concept " winter" : attributes "**hot**", "**cold**", "**chilly**"
2. In the concept base, some senses are not distinguished like they are in the Sensuous Knowledge Base.
   ex) In the Japanese language, hot and cold with the meaning of temperature of an object and atmospheric temperature are distinguished.
3. The attribute does not characterize the concept.
   ex) concept " sunset" : attributes "red", "**long**", "beautiful"

To solve these problems a method of refining is needed. The following refining methods aim to select the senses that are thought to be a proper characteristic of the concept. After the refinement those selected senses are considered and judged to be the final senses of the concept.

1. When there are a pair of senses of the opposite meaning in the attributes of a concept, the number of both of the words is counted. The sense with greater number is selected to be the sense. If there are the same number within the attributes, the degree of association is calculated, and the higher one is selected.
2. The senses that are not distinguished in the concept base are distinguished by the classification the word belongs to.
   ex) If the concept classification is "season," the sense hot or cold will be the one concerning atmospheric temperature.
3. Senses characterizing the concept are selected by the sense information inherited from the classification.
   ex) The classification of the concept "sunset" does not inherit the sense "sight : shape," so the attribute "long" is not selected as a sense of the concept.

With these refining methods, it is thought that the senses of the unknown word itself can be obtained from the concept base with high precision.

With the mechanisms, Classification Decision and Sense Retrieval, Sensuous Judgment concerning unknown words can be performed. By using the association mechanism of the concept base, the sense of the unknown words can be obtained. By using the Sensuous Judgment Knowledge Base and Unknown Word Processing, Sensuous Judgment can be performed on any nouns that exist in the concept base.

6 Evaluation of Unknown Word Processing

6.1 Evaluation of Classification Decision

600 nouns were randomly selected from the concept base as samples, and Classification Decision was performed on these words.

Of the 600 nouns, 88 were one of the representative words. Therefore, 512 nouns were handed over to the Classification Decision mechanism. The unknown word and the decided classification were compared and evaluated by human hand. The pair were evaluated into 3 ranks, rank A, rank B, and rank C.

When the decided classification can be judged as the classification of the unknown word, the pair is evaluated into rank A, such as a pair and fruit, or a sock and clothing.

When the decided classification cannot be judged as the classification of the unknown word, the pair is evaluated into either rank B or rank C.

Even though the decided classification may be not accurate, the sense of the decided classification can describe the unknown word. This study aims for accurate Sensuous Judgment, so this combination can be evaluated as correct concerning Sensuous Judgment. This kind of pair is evaluated as rank B. An example of a pair ranked into rank B is agar and vegetable. Agar is not a vegetable, but the sense of the classification vegetable is delicious. Agar is also delicious, so this pair is evaluated into Rank B. When the sense of the decided classification matches that of the unknown word, it is in rank B.

When the decided classification is not accurate, and the sense of the classification does not describe the unknown word, the pair is evaluated into rank C. An example of a pair that is evaluated into rank C is desk and food. The classification food has the sense delicious, and it would be inaccurate if the output of the Sensuous Judgment of desk were to be delicious.

Concerning the unknown words evaluated into rank A and rank B, the Sensuous Judgment will output senses within the limits of commonsense, and could be said that it is accurate. Sensuous Judgment concerning words evaluated into rank C will be inaccurate. The accuracy of the Sensuous Judgment concerning unknown words would be the percentage of rank A and rank B.

The 512 unknown words and decided classifications were evaluated into the three levels of rank. 200 were in Rank A, 204 were in Rank B, and 108 were in Rank C (See Figure 4(a)). If the 88 representative words of the 600 samples were also put into consideration and put into Rank AA, the evaluation result would be as seen in Figure 4(b).

From the results of the evaluation, it can be said that, with the Classification Decision accuracy of 78.91% concerning the unknown words and 82% over all, the Sensuous Judgment concerning the classification can judge well within the limits of commonsense.

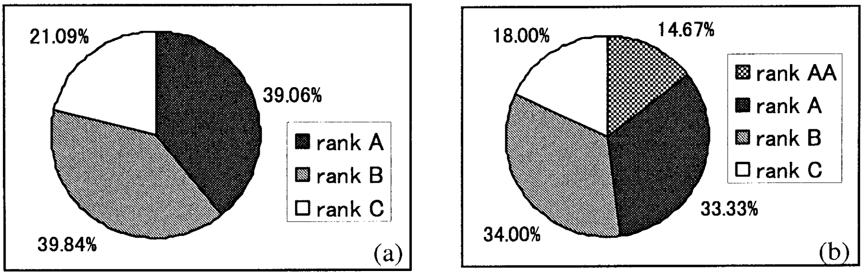


Fig. 4. (a) Evaluation of Classification Decision for Unknown Words. The accuracy of Sensuous Judgment concerning the classification of unknown words is 78.91%. (b) Evaluation of Classification Decision for all 600 words. The accuracy of Sensuous Judgment concerning the classification of words is 82%

Of the unknown words in Rank C, 65% were abstract nouns. More than half the Classification Decision that did not go well were concerning abstract nouns. This may result from the fact that there are not many abstract nouns among the representative words. An abstract noun is defined as a noun which names anything which you can not perceive through your five physical senses. Therefore, if it is related to a concrete noun, which is defined a noun which names anything that you can perceive through your physical senses, unnecessary sense information is related. This results in the evaluation into rank C. Representative words that are abstract, may need to be added, to relate the unknown word to an abstract noun.

6.2 Evaluation of Sense Retrieval

300 nouns were selected randomly from the concept base as samples, and Sense Retrieval was performed on these words. As a result, a total of 392 senses were retrieved and judged to be the sense of the words. Sense Retrieval can be performed on any concept that exist in the concept base, so to evaluate Sense Retrieval mechanism, the distinction between a representative and unknown word was not made.

The relation between the sample noun and outputted sense was evaluated by human into 3 levels. If the sense clearly characterizes the word, the relation is evaluated rank A. Some examples are lights and bright, and blood and red. If the sense could characterize the word in the limits of commonsense, such as hat and black, the relation is evaluated as rank B. If the sense does not characterize the word, and would be wrong if it did, such as tomato and long, the relation is evaluated as rank C.

As a result, 224 relations were evaluated into rank A, 53 into rank B, and 115 into rank C. Rank A and Rank B are considered accurate Sensuous Judgment.

From the results of the evaluation, it can be said that, with the Sense Retrieval accuracy of 70.66%, the Sensuous Judgment concerning the Sense Retrieval can judge properly within the limits of commonsense.

Of the relation evaluated into rank C, there is tomato and long. Tomato's classification inherits an uncertain sense "sight," so the sense "sight : shape : long" is judged not to be removed. 90% of rank C are senses that are left because of the inherited sense concerning only the type of the five sense. Senses of the classification must be made more concrete to lessen the number of evaluations in rank C.

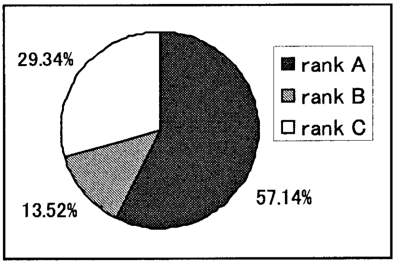


Fig. 5. Evaluation of Sense Retrieval. The accuracy of Sensuous Judgment concerning Sense Retrieval is 70.66%

7 Conclusion

With the Sensuous Judgment System (Sensuous Knowledge Base and Unknown Word Processing), senses of nouns can be obtained at a high precision. This system realizes the retrieval of meaning concerning the sense, for semantic understanding.

This work was supported by a grant to RCAST at Doshisha University from the Ministry of Education, Japan.

References

[NTT97] NTT Communication Science Laboratory: "*NIHONGOGOITAIKEI*", Iwanami Shoten (1997)

[Kasahara97] Kasahara, K., Matsuzawa, K. and Ishikwa, T.: "A Method for Judgment of Semantic Similarity between Daily-used Words by Using Machine Readable Dictionaries", IPSJ Journal, Vol. 38 , No.7, pp. 1272-1283 (1997)

[Manabe01] Manabe, Y., Kojima, K., Watabe, H. and Kawaoka, T.: "A Mechanical Refinment Method of a Concept-Base Using the Thesaurus and a Degree of Association among Concepts", The Science and Engineering Review of Doshisha University, Vo.42, No.1, pp.9-20 (2001)

[Watabe01] Watabe, H. and Kawaoka, T.: "Measuring Degree of Association between Concepts for Commonsense Judgments", Journal of Natural Language Processing, Vol.8, No.2, pp.39-54 (2001)

Towards a Natural Language Driven Automated Help Desk

Melanie Knapp[1] and Jens Woch[2]

[1] Institute of Computer Science III
University of Bonn, Germany
[2] Department of Computer Science
University of Koblenz, Germany

Abstract. In this paper, we present the linguistic components required for a natural language driven automated help desk. This work is significant for two reasons: First, the combination of neural networks and supertagging represents a novel and very robust way to classify non-trivial user utterances. Second, we show a novel way of integrating known linguistic techniques for the analysis of user input, knowledge processing, and generation of system responses, resulting in a natural language interface both for input and output. Our approach separates domain specific, language specific and discourse specific knowledge.

1 Introduction

The rapid development of technologies associated with the World Wide Web offers the possibility of a new, relatively inexpensive and effective standard user interface to help desks and appears to encourage more automation in help desk service. Typically, a help desk is defined as centralized help to users within an enterprise. Independent from the actual domain, help desks have to deal with two main problems: (1) efficient use of the know-how of an employee and (2) cost-efficient handling of many support requests. In this light, we present a natural language driven approach for modeling an automated help desk. This objective is motivated by the evaluation of support requests which showed that for 80 percent of all requests no specialized knowledge is needed. Hence, a solution database is sufficient for routine requests. Under this condition, our research concentrates on a computer-based so-called first support level.

Modeling a first support level requires the definition of all processing steps in a generic help desk system. We define a system structure with three main components. Within this design we do not distinguish among various input capabilities (e.g. telephone call, email, chat, fax or letter) and their appropriate analysis techniques. The first step in finding solutions is to analyze the textual input (independent of the extraction method) and to reduce the support request to a specific problem class. The second step is to request missing task parameters from the user. If the user's initial input is explicit, this step may be skipped. The third step in a generic help desk system is the verification of the specified solution. If the user is not satisfied with the solution, more task parameters for finding the solution must be extracted. In cases where no more task parameters can

This work partly is funded by the German Research Foundation.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 96–105, 2002.

be asked, the user request has to be delegated to a higher support level together with the already existing query information.

Our claim is that all three steps in the aforementioned generic system can be processed automatically. The automation should be based on a linguistically motivated solution, because empirical evaluations demonstrate that adaption to the user's dialogue preference leads to significantly higher user satisfaction and task success (cf. [Litman et al., 1998]). Wizard-of-Oz experiments by Boje (cf. [Boje et al., 1999]) also point out that users of automatic dialogue systems would like to take the initiative in many dialogues instead of answering a long list of tiny little questions. For modeling user-initiative dialogue systems, one important objective is to avoid leaving a user without a clear understanding of his/her options at a given point in the dialogue. Hence, for the design of the algorithm we define the following criteria: (1) the formulation of the user request should not be restricted, (2) no unnatural breaks between the user input and the result of the computer (especially for telephone calls, real time response must be guaranteed) and (3) no further inquiries into already explicitly or implicitly mentioned facts. A first approach of modeling user-initiative in an automatic help desk is described in [Harbusch et al., 2001]. Based on that experiences, this paper presents a further developed approach.

The paper is organized as follows. In the next section we describe the linguistic techniques used for the modeling of an automated help desk by delineating the components query extraction, inferencing and feedback generation, as well as their integration. Since this is work in progress, the paper closes with a discussion of open problems and future work.

2 Architecture of a Natural Language Driven Automated Help Desk

In this section we discuss the three tasks query extraction, inferencing and feedback generation and the difficulties which arise under the constraints of the aforementioned criteria for user-initiative dialogue systems.

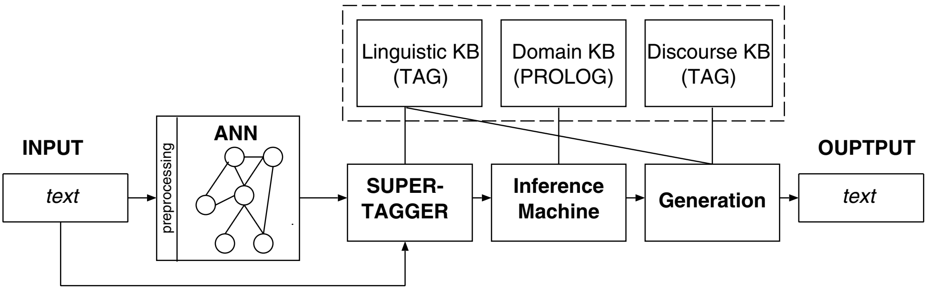


Fig. 1. Design of the automated help desk approach

We propose a system design illustrated in Fig. 1 which combines four main techniques. Starting with the text of the user input (analysis depends on the input medium), the artificial neural networks (ANN) allow a flexible classification of the user input to a specific problem class without any restrictions. Thereafter, the classified problem together with the input text is processed by a supertagger. The result is a logical representation of the user input. After that, the inference mechanism is used to extract all missing task parameters for a finer restriction of the problem class (if necessary), as well as to find a solution insofar it is supported by the prolog-based domain knowledge. Finally, the integrated automated text generation component, based on discourse and syntactic knowledge as well as on the output of the inference mechanism, generates a natural language utterance which, depending on the medium, is either printed out, e-mailed, or sent through a speech synthesizer. A detailed illustration of the linguistic techniques follows in the next sections.

2.1 Artificial Neural Networks (ANN)

For our approach the use of neural networks is motivated by the demand of a user initiative system. Neural networks allow the design of an unrestricted user interface. Supporting the user with a free natural language formulated problem specification increases the acceptance of such systems. On the other hand, it requires a great deal of energy to prepare the training and test corpus for a new domain.

The context and consequently the importance of words is measured by a hierarchy of recurrent plausibility networks ([Wermter, 1995]). Such a neural net (NN), which basically compares to simple recurrent networks by [Elman, 1990] consists - in addition to one input and one output layer - of $n > 0$ hidden layers, each of which has recursive link(s) to its context layers.

For the classification of the user input to a problem class, the following steps must be executed to design an ANN:

- The n main problem classes must be specified for a support domain and sample dialogues must be labeled in order to refer to their correct problem class.
- A reduced vocabulary for special word groups (i.e. general concepts) must be defined.
- Each word group w must be represented by a significance vector $(c_1, c_2, c_3, \ldots, c_n)$ with c_i corresponding to one of the n problem classes. For each c_i of a word group w the significance is computed by:

$$c_j = \frac{\text{frequency of a word from } w \text{ within class } j}{\sum_{i=1}^{n}(\text{frequency of words from } w \text{ within class } i)}$$

- Design of the net topology and training of the neural network with the labeled dialogues.

In order to build a prototypical system we have labeled 379 dialogues with the correct problem class out of the following seven problem classes in the domain *computer hardware* (note that the three major classes were intentionally selected to be easily differentiated by the topmost neural net (NN), the subclasses are selected to lay closely together to prevent all NNs from simply reacting to some key words but nevertheless enforce the learning of differentiating significance vectors):

a)	Problems with the hard disk(s): (1) hard disk's size identified not correctly (2) two hard disks interfere with each other (3) other hard disk problem
b)	Problems with the monitor: (4) monitor glimmering (5) colour shifting (6) other monitor problem
c)	(7) Other hardware problems

Any class has its own co-set (i.e. main-rest, monitor-rest and disk-rest). See Fig. 2 for an illustration of the hierarchy of plausibility nets which divides the classification into four problem classes (class 1, 2, 4, 5) and 3 cosets, respectively, at the individual levels in the hierarchy (class 3, 6, 7).

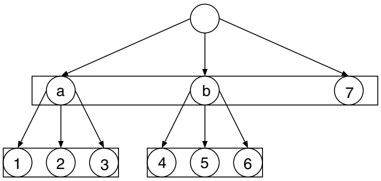


Fig. 2. Hierarchy of three plausibility nets

For our domain, we have defined 131 word groups, i.e. general concepts in this domain (such as “cable”, “monitor”, “setup”, . . .) with a total of 616 English and German words which can be considered being more or less synonymous to a concept. The table below lists some of the defined general concepts.

word groups	**corresponding words**
cable	cable, connection, . . .
monitor	monitor, screen, TFT, . . .
setup	setup, install, uninstall, . . .

Followed by the computation of all significance vectors of the word groups, examples for some concepts are outlined in the following table:

word group w	**significance vector**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
hard disk	.40	.26	.27	.00	.00	.00	.07
monitor	.01	.02	.01	.28	.33	.34	.02
setup	.28	.13	.19	.01	.01	.01	.36

The examples illustrate that words of individual word classes are more likely to occur in specific problem classes (e.g., ‘setup’ has a high probability of occurring in the context of a hard-disk problem or a general problem of the main rest-class and only has a low probability of occurring in the context of a monitor problem).

Hence, a text is represented by a sequence of significance vectors. Although different words could be theoretically represented by the same significance vector, the probability is small that such a vector sequence describes different phrases. After a series of tests, we dimensioned each recurrent plausibility network for our 7 problem classes into an input layer with 7 nodes, one hidden layer with 5 nodes, one context layer with 5 nodes connected to the hidden layer, and an output layer with 3 nodes.

The topology of the hierarchical structure directly depends on the overall number of problem classes in the respective domain and is a free parameter of the system to increase its ability to classify more reliably.

We have trained our system with 242 of the labeled and reduced turns and tested it with 137 randomly taken test dialogues. The system classifies quite reliably on all three levels of the hierarchy on the basis of context consideration. Particularly for the two sub-networks so far, our results are promising (cf. [Harbusch et al., 2001]).

2.2 Supertagging

While ANNs of reasonable complexity are capable of taking context into account, they are restricted in recognizing the grammatical structure of the context. For example, in *I have no problems with my screen* the ANN would classify that there is a problem with the user's screen – much to the contrary of what the user actually said. In *Cannot get this printer to work with this computer. I have follow all of the set up instructions from the book and on the screen and still nothing. Can you help?* the classification between printer, computer and screen is not obvious without deeper analysis. Parsing, and even partial parsing of the input would help but cannot be applied to solve the problem, since the user input most often is incomplete and/or grammatically incorrect. Assigning an elementary structure (supertag) to each lexical item of the user's input, such that the sequence of supertags is the most probable combination of structures a parser would need to actually *parse* the input, is the job of a supertagger [Joshi and Srinivas, 1994]. In our architecture, the sequence of tokens of the user's utterance (be it spoken, or written) are tagged with a supertagger which is trained on a domain specific corpus. Those supertags are elementary structures of a lexicalized Tree Adjoining Grammar (LTAG, cf. [Schabes et al., 1988]).

The result of the supertagger is aligned with the result of the neural network. If no token of the input sequence matches with the classification of the neural network, i.e. there is no anchor which has the same label as the classification, then either the classification or the supertagging failed, and the user is requested to paraphrase his statement. Otherwise, the anchor is analyzed in its structural context given by the supertag combination, i.e., the input is partially and only "almost" parsed [Srinivas, 1997]. This is no real parsing, since only the combination is checked whether the input could be derived from it.

With respect to the screen example above, negation of NPs could be applied to the input, as shown in Fig. 3. Here, the selective adjoining at I_3 allows negation. The complete structure now reveals, that the user in fact does not have a problem with his screen. The system does not know at this point what the problem actually is and should try to re-iterate on this question, but for now it suffices to state that the classification

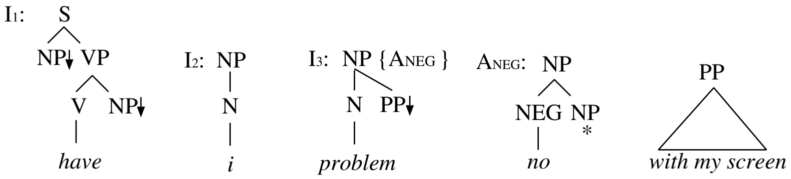


Fig. 3. A combination of supertags for *I have no problems with my screen.* (A '*' marks the foot node of auxiliary trees)

of having a problem with the screen, as suggested by the neural network, should be rejected.

2.3 Inference Mechanism

The domain knowledge is represented as a prolog database which can be queried about facts or things to do and can be extended with new facts derived from the conversation with customers. Generally, facts and clauses are applicable. How the domain knowledge itself is modeled exactly is irrelevant as long as the set of predicates ∩ anchors is not empty and their connection is meaningful to the domain knowledge; in other words, the naming of predicates determines what's utterable and what's not. For example, the rules

$$flicker(screen, occasional) := check(cables) \tag{1}$$

$$flicker(screen, permanent) := switch\_off(fridge) \tag{2}$$

state the recommendation that the cables should be checked if the screen flickers occasionally, or the fridge should be switched off if it flickers permanently.

The mapping from the tagged word list to a logical form happens on a simple but effective way: each anchor of the supertags is interpreted as a logical predicate. Additionally, adjoining is interpreted as bracketing, as well as valence positions of verbs. Conjunctions are interpreted as equivalent logical operations, and so on. Thus, the logical representation $have(i, no(problem, PP))$ is derived from the supertags as depicted in Fig. 3 and sent to the prolog machine for further inferencing.

This approach does not map any supertag set to a correct logical form, and that is why and where the syntactic realization of rules do play a role in modeling domains, but it turns out to be mostly sufficient for the kind of requests common in first support level conversations.

However, the logical form equivalence problem (cf. [Shieber, 1993]) arises here, i.e. different logical forms of the same semantics lead to different surface realizations. This is an inherent weakness of the proposed mapping to logical formulae, since it kills portions of the generator's flexibility[1].

[1] It has to be investigated whether generic mapping or at least extended lexical choice could help to remedy this gap. For example, in rule (2), $switch\_off$ should be mapped to grammar structures with different suffix positions.

In this way, the domain knowledge (the so-called what-to-say knowledge) is represented as a set of prolog clauses and a set of corresponding grammar structures, and is therefore separated from the linguistic and discourse knowledge (the so-called how-to-say knowledge). Inference mechanisms such as backtracking on solutions or decisions about what to do next, are implicitly given by the prolog machine. Thus, surfing through the problem space (discourse) is inherently guided by the prolog clauses. The advantages obviously are a relatively high domain independence: Switching the prolog and TAG database and extending the linguistic database for domain specific vocabulary are the only steps required to adapt the help desk to another domain. Additionally, the possibility of automatically checking the domain knowledge base for consistency by a theorem prover helps immensely to reduce maintenance costs.

The output of the inference mechanism is then fed to the generation process without further need of processing. The generator in principle does a reverse mapping by interpreting the predicates of the logical input as being anchors of lexicalized TAGs (see below for an extended example).

2.4 Automated Text Generation

The generation of the system's response is based on an integrated natural language generation system (cf. [Harbusch and Woch, 2002] in this proceeding). Basically, in an integrated or uniform generation system the linguistic knowledge is represented in the same formalism as the domain specific knowledge, i.e. the what-to-say component, and runs the same processing unit. A main advantage of such a system is that negotiation strategies on revisions can easily be imposed. This means that any communication between generation components is modeled implicitly by the overall decision making and backtracking mechanisms according to competing rules taken from the individual knowledge bases (i.e. no explicit communication language is required). In this approach the rules that cause the production of sentence initial elements, i.e. rules that are left-branching are collected on a lower level than right-branching rules. If a specific solution, which applied rules on a lower level, cannot continue with a new piece of input then, according to the general strategy, the more fine-grained rules are tried before more general decisions are backtracked. Thus, an overall solution is found with as few revisions as possible. Therefore, as in hierarchical constraint satisfaction systems, a fine-grained hierarchy of rules is assumed within any component. This means that any rule belongs to a hierarchical level of the respective component indicating how general or specific the rule is. According to these levels the granularity of rules becomes comparable. The generation process tries to find an optimal solution by satisfying as much general rules over the components as possible. In cases of backtracking, more fine-grained rules are revised before more general rules are considered. The definition of these hierarchies is done by the provider and leads to differently behaving systems.

A strictly sequential model for conceptualization, micro-planning and sentence formulation results from defining three hierarchical levels. All conceptual rules are put on the most general level, all micro-planning rules on the second level and all sentence formulation rules comprise the set of the most fine-grained rules. Hence, the overall system will first apply all conceptualization rules followed by all applicable micro-planning rules and finally the syntactic shaping is done[2].

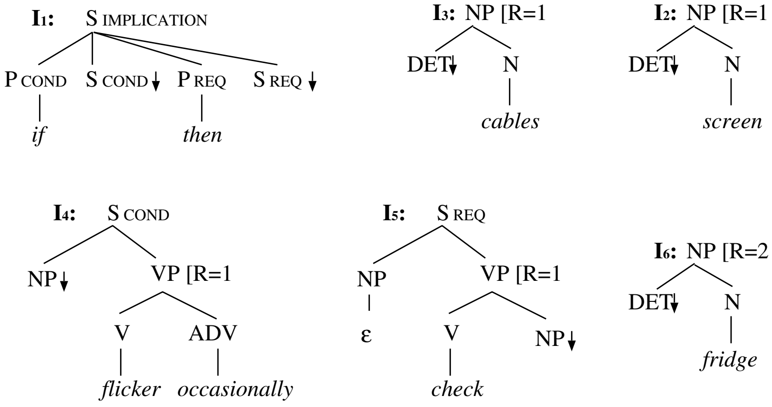


Fig. 4. A grammar fragment for the surface realization of rule (1)

Fig. 4 shows the grammar for the mapping of rule (1) to *If the screen flickers then check the cables*. The features for the syntactic alignment of number, gender, case etc. are omitted in the picture. After gathering all relevant trees according to the mapping of rule (1), the generator tries to build up a structure by applying any tree until none is applicable anymore (cf. [Harbusch and Woch, 2002]). Although each tree is tried, I_6 fails because of the failing unification of the attached feature R which is responsible for the correct selection according to the logical input. Thus, *If the fridge flickers occasionally then check the screen* is prevented[3].

In summary, the generation component is able to perform conceptualization and formulation tasks in an integrated approach. The advantage, besides those for the generation process itself, is the relinquishment of an explicit dialogue graph, whose poor flexibility vis-a-vis modifying the domain knowledge is a well-known problem.

[2] Another strategy is *incrementality*. In its simplest case, no lattice is specified at all, i.e. any rule is as general as another and so any rule of any component is applied as soon as possible. Thus, the parts of an utterance can completely be formulated out of already available constituents whereas some parts still undergo sentential planning and some constituents are not yet handed to the system. As known from incremental systems, already made local decisions about a prefix of the whole utterance can lead to dead-end situations which cannot resolve without rule revision. Those situations are fixed by trying other rules of the same level before higher level rules are revised.

[3] The abdication of such features is possible, but one would be forced to write less decomposed and therefore more redundant grammars which eventually culminates in highly specialized trees for each and every rule.

3 Conclusions and Future Work

In this paper, we have developed an architecture for a natural language driven automated help desk by addressing and integrating its specific requirements.

The problem of providing a less restrictive, more user-initiative input has been tackled twofold:

- A neural network captures the problem of classifying the user input according to significance vectors specific to the domain knowledge.
- A supertagger supports the classification by considering parts of the sentence's structure. Completeness is not required at this stage, particularly if the user input is spoken language which more often is incomplete and/or grammatically incorrect.

In general, [Litman et al., 1998] and [Boje et al., 1999] have shown that natural language interfaces do have a positive impact on the user acceptance, which in turn is profitable for the supporters.

Whether or not the supertagger may eventually replace the use of the neural network completely (and thereby remedy the need of its training) is part of our current research. For our simple domain, neural networks were serviceable, but their adjustment to other and probably bigger domains substantially equires a complete rewrite (on account of their topology, labeling and training) with unpredictible success.

The problem of generating system output is strongly related to the problem of representing knowledge. We have shown how domain knowledge (what to say) has been separated from linguistic and discourse knowledge (how to say it) and provided a mechanism for generating natural language sentences on the basis of the three of them. Switching the domain has little impact on the system. However, the problem of different lexicons per domain is not solved yet, but there is hope that by the time the growth of the lexicon approximates zero.

As a side effect, the dialogue graph, a common component of other help desk systems, which functions as a guide through the problem space of the domain, is realized implicitly, thereby remedying the problems associated with extending the domain knowledge:

- In the case of spoken output the new utterances are provided by the same "speaker", i.e. a speech synthesizer, and
- the dialogue model is not affected if the domain knowledge is extended. Additionally, the consistency of domain knowledge can be automatically checked by a theorem prover.

Therefore, we expect a significant reduction in maintenance costs for companies. Whether or not the realization of such a system is profitable depends on the proportion of first support level requests in relation to the overall support burden of a company. However, the tight interconnection of prolog clauses and supertags uses uncommon formalisms and probably needs familiarization for adopters. As already mentioned above, the paper describes work in progress, thus we do not have any third-party experiences on that topic.

The system's architecture in general has been developed with the goal of modularity in mind. By simply switching the input and output modules the system can be adapted to a wide range of different media. Thus, adapting the system to email driven information systems, web based chat applications, or telephony services impacts neither the domain nor on the system's inherent processing characteristics.

Whether the output is printed or spoken, is not just a matter of feeding a speech synthesizer: Despite the fact that high quality speech synthesis is not near at hand and therefore it might be necessary to enrich the string with control sequences, evidence ([Nass and Lee, 2001]) has been given that the user acceptance is highly influenced by prosodic parameters. However, studies have yet to be made whether the analysis of the actually spoken user input suffices to parameterize the speech synthesis in real time to gain better user acceptance.

References

[Boje et al., 1999] Boje, J., Wirén, M., Rayner, M., Lewin, I., Carter, D., and Becket, R. (1999). Language-processing strategies and mixed-initiative dialogues. In *Procs. of IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems.*

[Elman, 1990] Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

[Harbusch et al., 2001] Harbusch, K., Knapp, M., and Laumann, C. (2001). Modelling user-initiative in an automatic help desk system. In *Proc. of the 6th Natural Language Processing Pacific Rim Symposium, Tokyo, Japan.* in press.

[Harbusch and Woch, 2002] Harbusch, K. and Woch, J. (2002). Integrated natural language generation with schema-tree adjoining grammars. In *Procs. of the 3rd International Conferences on Intelligent Text Processing and Computational Linguistics (CICLING)*, Mexico, Mexico City. Springer-Verlag. forthcoming.

[Joshi and Srinivas, 1994] Joshi, A. K. and Srinivas, B. (1994). Disambiguation of super parts of speech (or supertags): Almost parsing. In Nagao, M., editor, *Procs. of the 15th International Conference on Computational Linguistics (COLING)*, volume 2, pages 154–160, Kyoto, Japan.

[Litman et al., 1998] Litman, D. J., Pan, S., and Kearns, M. S. (1998). Evaluating response strategies in an web-based spocken dialogue agent. In *Procs. of the 36th Annual Meeting of the ACL and the 17th COLING*, Montreal, Canada.

[Nass and Lee, 2001] Nass, C. and Lee, K. M. (2001). Does computer generated speech manifest personality? experimental test of of recognition, similarity-attraction, and consistency-attraction. *Journal of Experimental Psychology, Applied.* in press.

[Schabes et al., 1988] Schabes, Y., Abeillé, A., and Joshi, A. K. (1988). Parsing strategies with 'lexicalized' grammars. In Hajicova, E., editor, *Procs. of the 12th International Conference on Computational Linguistics (COLING)*, volume 1, Budapest, Hungary.

[Shieber, 1993] Shieber, S. M. (1993). The problem of logical–form equivalence. *Computational Linguistics*, 19(1):179–190.

[Srinivas, 1997] Srinivas, B. (1997). Performance evaluation of supertagging for partial parsing. In *Procs. of the 5th International Workshop on Parsing Technologies (IWPT)*, Boston/USA.

[Wermter, 1995] Wermter, S. (1995). *Hybrid connectionist natural language processing*. Chapman and Hall, International Thomson Computer Press, London, UK.

Lexical Tuning

Yorick Wilks and Roberta Catizone

University of Sheffield, UK
yorick@dcs.shef.ac.uk

Abstract. The paper contrasts three approaches to the extension of lexical sense: what we shall call, respectively, lexical tuning; another based on lexical closeness and relaxation; and a third known as underspecification, or the use of lexical rules. These approaches have quite different origins in artificial intelligence(AI) and linguistics, and involve corpus input, lexicons and knowledge bases in quite different ways. Moreover, the types of sense extension they claim to deal with in their principal examples are actually quite different. The purpose of these contrasts in the paper is the possibility of evaluating their differing claims by means of the current markup and test paradigm that has been recently successful in the closely related task of word sense discrimination (WSD). The key question in the paper is what the relationship of sense extension to WSD is, and its conclusion is that, at the moment, not all types of sense extension heuristic can be evaluated within the current paradigm requiring markup and test.

1 Introduction

The principal aim of this paper is to discuss what Lexical Tuning (LT) is, broadly defined and selectively practised, and to discuss its relationship to word sense disambiguation (WSD), with the aim of making it, too, quantitatively evaluable as WSD now is within the SENSEVAL regime (Kilgarriff 1998).

Automatic word-sense disambiguation (WSD) is now an established modular task within empirically-based computational linguistics and has been approached by a range of methods (Ide and Veronis, 1999) sometimes used in combination (Wilks and Stevenson, 1998). These experiments are already showing success rates at the desired ninety-five-per-cent levels attained by established modules like part of speech tagging in the mid-Nineties: over a few text words Yarowsky has claimed mid nineties (1995), and with systems that claim to deal with all text words Sheffield and NMSU-CRL now also claim similar figures (Nirenburg 1997).

These methods have included some, such as the use of the agent, object etc. preferences of verbs, that go back to those used in the earliest toy AI systems for WSD, such as (Wilks, 1968, 1972). Yet even those toy systems were set up with an explicit recognition that WSD was different in a key respect from tasks like POS: namely, that lexicons need to adapt dynamically in the face of new corpus input.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 106–125, 2002.

The contrast here is in fact quite subtle as can be seen from the interesting intermediate case of semantic tagging: attaching semantic, rather than POS, tags to words automatically, a task which can then be used to do more of the WSD task (as in Dini et al., 1998) than POS tagging can, since the ANIMAL or BIRD versus MACHINE tags can then separate the main senses of "crane". In this case, as with POS, one need not assume any novelty required in the tag set—in the sense of finding in the middle of the task that one needed additional tags—-but one must allow for novel assignments from the tag set to corpus words, for example, when a word like "dog" or "pig" was first used in a human sense. It is just this sense of novelty that POS tagging does have, of course, since a POS tag like VERB can be applied to what was once only a noun, like "ticket". This kind of assignment novelty, in POS and semantic tagging can be premarked up with a fixed tag inventory, hence both these techniques differ from genuine sense novelty which, we shall argue, cannot be premarked in any simple way.

This latter aspect, which we shall call Lexical Tuning, can take a number of forms, including:

(a) adding a new sense to the lexical entry for a word
(b) adding an entry for a word not already in the lexicon
(c) adding a subcategorization or preference pattern etc. to any existing sense entry

and to do any or all of these on the basis of inductive (corpus) evidence. (a) was simulated in the early work just referred to, and (b) was first attempted in Granger (1976). (c) is at first sight more problematical in that it could be argued that it cannot be defined in a theory-free way, since what can be added automatically to a lexical entry on the basis of corpus evidence necessarily depends on the structure of the lexicon to be augmented, e.g. the nature of the features the lexicon contains. This is undoubtedly correct, but the general notion of what lexical tuning is can still be captured in a non-trivial theory-free way by means of the "etc." above, the general notion proposed being one of a very general function mapping an existing lexicon and a corpus to a new (tuned) lexicon.

In practice, the three types above are neither exclusive nor exhaustive, although task (b) may be quite different in nature in that it excludes straightforward use of a well-known technique that appears under many names, such as "lexical rules" (Briscoe 1989, Buitelaar 1993), and which strictly falls outside the function, described above, by which new senses of a word are induced from knowing not only a corpus but an existing lexical entry. The lexical rules (LR) tradition goes back at least to Givon's (1967) work on extending dictionary entries independently of corpus context, and can be seen as a direct inheritor of the generative linguistics tradition, in the sense in which that is now often contrasted with the corpus linguistics tradition. We shall argue below that this is not altogether fair, since LR researchers do often refer to and call upon corpora, but always that special set of corpora that should more properly be described as meta-corpora, namely the resources of facts about usage, such as (machine readable) dictionaries, thesauri and wordnets. Note that these are all machine readable and the difference here is not about computation, only about where one seeks one's evidence and to what extent all corpus forms can be accounted for in advance, and within lexical constructions.

Combining these three types of LT is also a source of potential confusion: if a word is unknown to a lexicon, then any computational system can see that immediately, but many would say (clinging firmly to the force of the word "homonymy") that the three main senses of "post" (post1 = mail; post2 = stake; post3 = job) are, in effect, different words, arbitrarily linked by English spelling. So, some would say that inferring a new sense of "post" (if using a lexicon from which one of the three above senses was missing) is identical to task (b) above, and not properly task (a), since one could not expect to induce a new, major, sense of "post" from its existing senses, by any system that could so extend senses in cases of so-called "regular polysemy" (Briscoe, 1989).

This problem is independent of a more general one affecting tasks (a) and (c): namely, when does a new context for a word give rise to a description that should be deemed a new feature or new pattern, rather than a 'relaxed' version of an existing one. This is, like all forms of the problem, the ultimately insoluble general learning problem and a matter in the end of arbitrary choice or parameter setting within an algorithm.

To summarise: this formulation of LT assumes we already have a human-created resource we shall call structure1, i.e. the lexicon we started with, perhaps together with an associated knowledge base or ontology. LT is thus the process or mapping function:

I: structure1 + corpus → structure2

which indicates that an earlier state of the structure itself plays a role in the acquisition, of which structure2 is then a proper extension (capturing new concepts, senses etc). This is a different model from the wholly automatic model of lexicon acquisition often used in, say, TIPSTER related work (Riloff, 1990), which can be written:

II: corpus → structure

Here one does not update or "tune" an existing lexicon but derives one directly and automatically from a corpus. There is no doubt II. can be an effective tool, certainly in the case of unknown languages or domains, but the assumption made here about the quite different function I. is that we cannot understand the nature of the representation of meaning in lexicons, or elsewhere, unless we can see how to extend lexicons in the presence of incoming data that does not fit the lexicon we started with. The extension of representations, one might say, is part of an adequate theory of representation.

2 Evaluating WSD and Its Relationship to Lexical Tuning

A central issue in any application of empirical methods to computational linguistics is the evaluation procedure used, which is normally taken to consist in some form of experiment using premarked-up text divided into training and (unseen) test portions. Standard supervised learning for WSD involves attaching tags to each text word (or more often each content, or open-class, word) corresponding to one and only one sense from a chosen set of senses from a lexicon.

Apart from the well-known problem of the difference between sense-sets (if we can continue to use that phrase unexamined, for the moment) for a given word in different lexicons — although they are not arbitrarily different, and that is a vital fact — there are problems concerned with subjects having difficulty assigning a corpus word occurrence to one and only one sense during the markup phase.

Kilgarriff (1993) has described such problems, though his figures suggest the difficulties are probably not as serious as he claims. However, we have to ask what it means to evaluate the processes of Lexical Tuning as defined above : this seems to require annotating in advance a new sense in a corpus that does not occur in the reference lexicon. The clear answer is that, on the description of WSD markup given above, the sense extension (task (1) above: tuning to a new sense) CANNOT be pre-tagged and so no success rate for WSD can possibly exceed [100% MINUS the percentage of extended sense occurrences].

One question about Lexical Tuning that is not often discussed is made explicit by the last expression: what is the percentage of senses needing tuning in normal text? One anecdotal fact sometimes used is that, in any randomly chosen newspaper paragraph, each sentence will be likely to have an extended sense of at least one word, usually a verb, which is a use that breaks conventional preferences (Wilks 1972) and which might therefore be considered extended or metaphorical use, and which may or may not be in a standard lexicon. This is a claim that can be easily tested by anyone with a newspaper and a standard dictionary.

That, even if true, does not give us a firm figure to work with. However, it could suggest that any figure for basic WSD of over 95% must be examined with great care, because it almost certainly cannot have been done by any method using pre-tagging, and the onus on anyone making such a claim is to show what other explanation of his high success figures can be. Subsequent examination of actual machine WSD for a posteriori satisfactoriness can never be a plausible measure: i.e. anything along the lines of this is what our system gave as new sense contents for this corpus and we liked what we got! Another possibility, that will be considered in more detail later, is that novel sense might be detected by an occurrence that cannot be identified with any of the list of senses for the word available to the human tagger. The problem here may be just one of an inadequate dictionary list–though that is no objection in principle as novelty is always with respect to the state of a lexical structure, but also that this will conflate regular novelty, that could have been produced by LR, from any other kind. However, none of these objections are are insuperable and, indeed, (Kilgarriff 2001) used such a measure in an attempt to evaluate the Generative Lexicon (GL, q.v.) approach to lexical novelty. On a small sample, Kilgarriff estimated the occurrence of novel senses at 2% over and above anything due to regular polysemy.

3 How Then to Evaluate Lexical Tuning Claims?

If Lexical Tuning (alias LT) is a real phenomenon, it must be possible to evaluate it in some reasonable way. To make headway here, let us first set out possible basic paradigms or methods for sense extension and seek for clues as to how to evaluate them. One such early paradigm was set out in (Wilks 1978) under the title "Making preferences more active", and which was implemented at the "toy" levels of that period, though it may still be false as to the relationship of new senses to existing ones. Let us call that historical example: **Method A**. It was based on the notion of:

i. The cuing function (for LT) of the preference failure of a word W1 in a text (e.g. a verb used with an unexpected agent class);

ii. The location of a W2 in a knowledge structure, that defined how the world for that word sense normally is;

iii. The substitution in the text representation of the "failed" word by a new, more satisfactory word sense W2 (in the lexicon) which has the right lexical preferences;

iv. The claim that W1 should have its lexicon extended by the structure for the appropriate sense of W2, where appropriate structure may mean its preferences, subcategorization patterns, semantic or other links, explanatory gloss etc.

The main 1978 example was "My car drinks gasoline", which has a failed [human] agent preference, which is then (criterion i above) the trigger to locate a fact representable as [cars use gasoline] in a knowledge base about cars (ii and iii above), so that "use" can provide a plausible new sense of "drink" (iv above). However, this heuristic not wholly satisfactory, since it does not capture the idiomatic force of "drink → use a lot of" implicature of this usage. Moreover, the process must not just locate any action or process of cars associated with gasoline, for that will include "leak", as in [cars leak gasoline]. We can suppose this is achieved either (or both) by assuming leaking gasoline is not described in a stereotypical car function knowledge base or that drink/use are linked by some underlying semantic structure (such as a shared type primitive or some degree of closeness, however defined, in a synonym/WordNet list classification) and in a way that drink/leak are not.

This location of a preference-satisfying KB entity to substitute for a failing semantic structure was called PROJECTION in 1978, and is the kind of inference that has played a substantial role in the later work of Pustejovsky and others under names like "coercion". The method illustrated above based on "preference failure" would apply only to verbs and adjectives, which were the grammatical types coded with preferences in that system, although another possibility set out in the 1978 paper was that either participant of the failed preference link could be substituted by something better fitting (ie. the verb or its agent): the sense extension proposed above is of the action because of what was in the knowledge base (KB), and within the standard AI assumption of knowledge-based processing, but one could also take the same example as ascribing a human quality to cars. However, the KB does not support any substitution based on the agent, because one would expect to locate in the car-KB forms like [person drive cars], but not any KB form like like [person drink gasoline], which is what would be needed to support an alternative, competing, tuning of "car".

Method A2: However, this sort of possibility is the one that underlies a metonymic preference failure like

THE CHAIR opened the meeting.

Again we have agent-action failure, but now there is no KB support for any form with a value for ACTION satisfying [chair ACTION meeting) of the kind we saw for drink/use. However, projection to [person sit-on chair] would be supported in a standard KB, as would [person open meeting] as part of a general knowledge structure for the conduct of meetings, and the preference of the corresponding sense of "open". So, in this class of case as well we might expect the same procedures to tune to a new sense of "chair" as "person" (who opens meetings).

Now let us contrast the above paradigm for sense extension with that used in recent CRL work (Nirenburg 1997), one intended as more fine grained than the "consumer driven" (Sergei Nirenburg's term) approach, or that of "final task" driven projects, such as the ECRAN project, namely that of carrying out a "final task" such as information extraction before and after tuning a lexicon against a domain corpus and then seeing if Information Extraction results are improved. "Final task" here is to be contrasted with "intermediate tasks", such as WSD, which are often evaluated directly in competitions but which have no real NLP function outside some final task, one that serves a purpose for a consumer.

The CRL basic methodology (using the Mikrokosmos KB, which we shall call MK for short , Nirenburg and Raskin 1996) is quite different from A above. Let us (at the inevitable risk of error in summarising someone else's work) describe it in two ways as follows:

Method B1:

1. Preference failure of an occurrence of word W1 in the corpus
2. Seek the closest EXISTING sense of W1 in the MK lexicon by relaxing the preference constraints of W1.
3. Consider later how to subdivide the expanded-relaxed occurrences of W1 to create a new sense if and when necessary, perhaps when the "expanded" occurrences form a new cluster, based on related relaxations, so that a new sense of W1 can be separated off in terms of a new set of constraints in the MK lexicon.

OR

Method B2:

1. Preference failure of a an occurrence of word W1 in the corpus
2. Seek in the MK KB for a word sense W2 hierarchically below W1, but whose preferences are satisfied in the example.
3. Take W2 to be the sense of W1 in the given context.

It is not wholly clear in the context of the paper referred to whether B1 and B2 result in adaptations to the lexicon, which is what we are asking as the minimal, necessary, condition for anything to be called LT, so as to avoid including in LT all hapax occurrences of unusual con junctions. However, these heuristics are of interest whether or not the lexicon is permanently adapted, as opposed to deriving a new sense representation for a word for immediate use. These methods make less radical use of world knowledge than A, but one which runs far less chance of making wrong extensions. The key notion in B1 is the search for a CLOSEST EXISTING SENSE of the same word, which may well represent a core aspect of meaning extension missing from the earlier approach, and which will in any case be essential to task (c) (though it cannot, by definition, be used for task (b) which is that of the "unknown word"). It also cannot help in true homograph/homonym cases, like "post", where the approach A might stand a chance, but we proposed at the beginning to exclude consideration of such extension for now - or rather to accommodate it to task (b) and not (a).

Method B2 shows an interesting notion of preference breaking somewhat different from that of A: a canonical CRL example is:

He PREPARED the bread.

where the declared aim of the adaptation (Nirenburg 1997) is to tune the sense of "prepare" , for this occurrence, to the appropriate sense of "bake", which is the verb in the Mikrokosmos KB for the preparation of bread and whose preferences fit a BREAD object as those of "prepare" do not. The process here is close to Method A in that a stored item in a KB licenses the tuning and, again like Method A, the result is the substitution of one word sense by the sense of another word. As with method A, this will only count as LT (on the strict definition used in this paper) if the lexicon is changed by this process so as to install "bake" as a sense of "prepare" and it seems this is not done in the CRL system.

However, the most interesting feature of the B method, is that the constraint satisfaction of "bake" is not passed up the hierarchy of actions and sub-actions. This is an idea going back to Grice (as a failure of the quantity maxim, Grice 1964) but one little used in lexical semantics: that the too general is semantically ill-fitting, just as complete misfitting is. In preference terms, it means that the over general is also a preference failure (quite contrary to the way that notion has usually been used to include subclasses of fillers, e.g. that to prefer a FOOD object is normally to accept a BREAD object, given that bread is a kind of food.

As we noted, Method B2 is not LT if the lexical entry for "prepare" is not altered by the acceptance of "He prepared the bread", but this is mere definition. Relaxation to "higher classes" can, however, be explicitly marked in a lexicon, and is therefore LT, as would be the case with "The Chair opened the meeting" example, if viewed as relaxation to accept PHYSOBJ agents and not just HUMAN ones. There is always a price to pay in relaxation accounts of tuning because once a preference is relaxed it cannot subsequently be used to select as a constraint.

Consider the following:

The whole office waited for the boss to arrive

The two men cleaned the offices as ?they waited for the janitor to arrive

One cannot both relax the lexical entry for "wait" so as to accommodate its agent in the first sentence and use the standard preferences of "wait" for [human] agents to resolve ?they in the second. This point is an argument not only against relaxation but against any method for deriving preferences by corpus analysis (Grishman 1987, Resnik, 1992) in any simple manner since both sentences could well be attested in the same corpus.

The CRL researchers deny there is any such lexical adaptation but the puzzle remains that, as "bake" is already listed as a sub-action of "prepare" in their lexicon there is no need for data to instantiate the situation, though it might be useful if something indicated that this"prepare" is in the food domain, as opposed to preparing walls for painting, for example. This could be done by making "prepare" a subaction of a higher verb "cook", but then "bake", like"roast","boil" and "grill" would not fall under "prepare" at all but under a subsequent stage of "book" after preparation.

There will naturally be disputes about how widely this kind of quantity restriction can be enforced: one might also say that preparing bread is a sequence of subactions, including mixing and leaving-to-rise (rather like Schank scripts of old, Schank and Abelson, 1977); in which case the type BREAD is the proper object for all of them including "prepare", so that the B methods can never be called in because there is no preference failure trigger them.

Method B1 should lead to a quite different interpretation of this example: on B1 "prepare bread" (if deemed preference breaking as they claim, and in their sense) should lead to a relaxation to an EXISTING sense of "prepare" (and not "bake" at all), yet what is that existing sense?

Is the car/drink example (Method A) one of lexical extension when compared to the B methods; which is to say, do we want to deem "use" a sense of "drink" in the context of a car's consumption of gasoline and retain that modification in a lexicon? Identifying this as a possible extension is a necessary but not sufficient condition for a full LT lexicon modification which requires further confirming instances of entities of a machine type drinking fuel-like liquids, e.g. steam engines drinking water, aeroengines drinking kerosene and so on. This is a different type of extension from the B-type examples involving possible relaxations of agents and objects of fixed verbs. Both A and B type extensions, if real, are different from what others are calling regular polysemy, in that they cannot be precoded for in lexical entries by rules or any similar method.

4 Closest Sense Heuristics and Text Markup

The CRL approach measures success, at least initially, by human mark up to the closest existing lexical sense (though see below on "Chateaubriand"). This may make it possible to achieve a generally acceptable type of evaluation procedure for lexical tuning (whether or not one adapts the lexicon, in the face of any particular example, need not affect the use of the procedure here) if there can be inter-subjective agreement on what is a lexically closest sense in a training text. That would then the phenomenon being tested, along with the general (and attested) ability to assign a sense to a text word when the sense is in the lexicon used, though the human marker should also obviously have the choice of declining to mark a closest sense, given a particular state of the lexicon, if he believes it inappropriate in the context. If LT is to be evaluated in such a way, a marker will have to be able to indicate closest sense separately from appropriate sense.

Examples can be produced (due in this case to Steve Helmreich) within the well-known Restaurant Metonymy example paradigm to suggest that the extended sense to be constructed by this Method B1, leading to the closest existing sense, may not always be appropriate.

Consider:

The Chateaubriand wants a drink

where "Chateaubriand" is lexically coded both as a steak (what the diner ordered) and an C18 French politician of that name. The latter may well be chosen as the closest sense (since it satisfies the [human] agent constraint on "want") but the extended or relaxed sense should actually be related to steak, the first sense.

Restaurant Metonymies (RMs), though attested, have perhaps played too strong a role in the field, given their infrequency in real life and proper name RMs could perhaps be dismissed as a tiny subclass of a tiny subclass and a proper subject only for AI. Perhaps the closest sense heuristic can be saved by some careful analysis of "the" in the last example; it is always the cue for a Restaurant Metonymy, but rarely in politics, and we shall assume in what remains that the heuristic can be saved in some such way. After all, there need be no similar problem here with RMs that are not proper names, as in:

The lasagna wants a drink.

5 Pustejovsky's Position on Lexical Expansion

In The Generative Lexicon (1995, TGL for short) Pustejovsky (JP for short) sets out a position that has features in common with work already described, but offers a distinctive view of the lexicon and in particular its underspecification in crucial respects; and the aspect that will concern us in this paper is whether or not that underspecification is any form of LT as described here, as implying the augmentation of the lexicon in the face of sense novelty in a corpus. It seems JPs position that his key class of examples does not imply the creation of a new sense from an existing one in the face of corpus evidence, but rather the incorporation of a prestored ambivalence within a lexical entry. That this can be misunderstood can be seen from an attack on JPs TGL by Fodor and LePore (FL for short, Fodor and Lepore, 2000) in which they attribute to him a sense ambiguity for such examples, and indeed an unresolvable one.

They claim that JP's:

He baked a cake.

is in fact ambiguous between JP's "create" and "warm up" aspects of "bake", where baking a cake yields the first, but baking a potato the second. JP does not want to claim this is a sense ambiguity, but a systematic difference in interpretation given by inferences cued by features of the two objects, which could be labels such as ARTIFACT in the case of the cake but not the potato.

But in fact, "bake a cake" is ambiguous. To be sure, you can make a cake by baking it; but also you can do to a (preexistent) cake just what you can do to a (preexistent) potato: viz. put it in the oven and (non creatively) bake it." (op.cit. p.7)

From this FL conclude that "bake" must be ambiguous, since "cake" is not. But all this is absurd and untrue to the simplest facts about cakes, cookery and English. Of course, warming up a (preexistent) cake is not baking it; who ever could think it was? That activity would be referred to as warming a cake up, or through, never as baking. You can no more bake a cake again, with the other interpretation, than you can bake a potato again and turn it into an artifact. The only obvious exception here might be "biscuit", whose etymology is, precisely, "twice cooked", though not baked.

FL like syntactically correlated evidence in semantics, and they should have noticed that while "baked potato" is fine, a "baked cake" sounds less good, which correlates with just the difference JP requires (cf. baked fish/meat, which are also commonplace).

FL's key argument against TGL is that it is not possible to have a rule, of the sort JP advocates, that expands the content or meaning of a word in virtue of (the meaning content of) a neighbouring word in a context, namely, a word in some functional relation to the first. This is precisely the kind of rule that everyone in the AI/NLP tradition, including all those mentioned in this paper, agree is fundamental, as indeed did Fodor in his "selection restriction" period long ago (1966).

Again, JP, like many in what we could call the NLP tradition, argues that in:

John wants a beer.

the meaning of "wants" in that context, which need not be taken to be any new or special or even existing sense of the word, is to be glossed as "wants to drink a beer", and this is done by a process that varies in detail from NLP researcher to researcher, but always comes down to some form close to Method A at the beginning of this paper, such as:

X wants Y → X wants to do with Y whatever is normally done with Y

where the last clause is normally instantiated from some form of KB or rich lexicon. An issue over which AI researchers have differed is whether this knowledge of normal or default use is stored in a lexical entry or in some other computational knowledge form, such as what was sometimes called a script (Schank and Abelson, 1977) and was indexed by words but was a KB rather than a conventional lexical entry.

Nothing in this paper requires us to discriminate between types of structures, however complex, if they are indexed by a word or words, though that difference is important to some researchers discussed here, such as Nirenburg, for whom the Mikrokosmos KB and lexicon are quite distinct. JP stores the inference captured in the rule above within the lexical entry under a label TELIC that shows purpose. In earlier AI systems, such information about function might be stored as part of a lexical semantic formulas attached to a primitive GOAL (Charniak and Wilks, 1976) and later, as we noted earlier, within larger knowledge structures called pseudo-texts (Wilks 1980) (so named to emphasise the continuity of language and world knowledge).

JP's specific claim is not that the use of rules like the one above produces a new sense, or one would have to have a new sense corresponding to many or most of the possible objects of wanting, a highly promiscuous expansion of the lexicon. JP resisted augmentation of the lexicon, though other researchers would probably accept it and this difference may come down to no more than the leaving of traces in a lexicon and what use is made of them later. Nor is this like the application of "normal function" to the transformation of

My car drinks gasoline.

discussed earlier where it was suggested that "drink" should be replaced by the structure for "consume" in a context representation containing broken preferences (unlike the "want" case if its preferences are set appropriately, so that almost anything can be wanted) and where augmentation of the lexicon would be appropriate if such cases became statistically significant.

Is underspecification just language-specific lexical gaps?

Let us look at the key Pustejovsky example in a new way: the bake cake/ bread/ potato examples may not draw their power from anything special to do with baking but with lexical gaps and surplus in English connected with cake and bread. Suppose we adopt, just for a moment, a more Wierzbickian approach to baking and assume as a working hypothesis that there is only one, non-disjunctive, sense of bake and it is something like:

"to cook a food-substance X in a heated enclosed space so as to produce food-substance Y"

Thus we have, for X and Y for our usual suspect substances in English:

potato [potato, baked potato]
bread [dough, bread]
cake [cake mixture, cake]
pie [pie, pie]

as well as:

fish [fish, (baked) fish]
ham [ham, baked ham]

There is no mystery here,. but looking at a range of well-known substances can take us out of the rather airless zone where we discuss the relationship of "bake" and "prepare" away from all data, and without considering in parallel "roast", "boil", "grill" etc. We would argue that there is no pressing need to gloss the implicit structure here as a disjunction of senses or aspects of "bake"; it is imply that the lexical repertory of English varies from food to food, thus

We bake ham and get baked ham
We bake dough and get bread
We bake cake mixture and get cake
We bake (a) potato and get a (baked) potato

There is no reason to believe that these cases fall into two classes, the creative and non-creative at all: it simply that we have words in English for baked dough (bread) and baked cake mixture (cake) but not a word for a baked potato. If we did have such a word , baking a potato would seem more creative than it does. Contrast Kiswahili, which has a word for uncooked rice (mchele) and a word for cooked rice (wali). In English

We cooked rice

does not seem creative in English but rather matter of mere heating since there is only one word for the object transformed. So, on an underspecification approach to Kiswahili:

We cooked wali/mchele

are two sentences (if all in Kiswahili) bearing two differing interpretations of "cook", only one of them TELIC, and hence

We cooked rice

must also be ambiguous/underspecified/disjoined in interpretation in English. But this is surely not true, indeed absurd, and a language cannot have its verb semantics driven by its lexical gaps for nouns! If this analysis is plausible there is no disjunction present at all in baking cakes and potatoes either , and if by chance "dough" meant dough or bread in English (as is surely the case in some language) this whole issue could never have arisen.

We should not exaggerate the differences between the main approaches discussed so far: all subscribe to

i. sense is resolvable by context
ii. we can create/extend sense in context by various procedures

but not all to

iii. the methods of (ii) are close to WSD and lead naturally to lexical adaptation/tuning
iv. the adaptation produced by (ii) leaves some record in the lexicon.

That i and ii are broadly agreed here can be seen by contrasting them with positions in the logical representation of meaning (quite outside NLP/AI and much of CL) who do not subscribe to i and ii.

Let us attempt to cross classify the three methods A(LT), B/(relaxation), C/(LR) against the following procedural aspects:

- claimed adaptation of lexicon:
  Yes (necessary condition for), uncertain, No.
- assumed hierarchies of verbs:
  No (available not used), Yes, Yes.
- decomposed structure of verbs assumed:
  Yes, No, Yes.
- claimed extension separate from lexicon:
  Yes, Yes, No.
- access to a separate knowledge base:
  Yes, Yes, No.
- extension extension driven by a new corpus example (vs. preplanned in):
  Yes, Yes, No.
- extension triggered by some preference failure:
  Yes, Yes, No.

6 Generalising the Contrast of LT with Lexical Rules (LR)

Lexical Tuning (LT) is closely related to, but rather different from, a group of related theories that are associated with phrases like “lexical rules” (LR); all of the latter seek to compress lexicons by means of generalisations, and we take that to include DATR (Gazdar 1993), methods developed under AQUILEX (Briscoe and Copestake 1989), as well as Pustejovsky’s TGL discussed above and Buitelaar’s more recent research on underspecified lexicons (1993). LR we take to be any approach, such as Pustejovsky or Briscoe, in the tradition of Givon that seeks to extend lexical entries not only in the face of corpora but independently of them. To take a classic example, lexical entries for animals that can be eaten can be contracted and marked only ANIMAL, given a rule that extends on demand to a new sense of the word marked MEAT. This is an oversimplification of course, and problems arise when distinguishing between eatable and uneatable animals (by convention if not in strict biology). Very few want to extend “aardvark” with MEAT though there is no logical reason why not, and an ethnographic survey might be needed for completeness in this area; foods are simply not universal.

All this work can be traced back to early work by Givon (1967) on lexical regularities, done, interestingly to those who think corpus and MRD research began in the 1980s, in connection with the first computational version of Websters Third at SDC in Santa Monica under John Olney in 1966. It can also can be brought under a heading "lexical compression" whether or not that motive is made explicit. Givon became interested in what is now called "systematic polysemy", as distinguished from homonymy which is assumed to be unsystematic: his key examples were like "grain" which is normally given a count noun or PHYOBJ sense in a (diachronic) dictionary cited earlier than the mass noun sense of "grain in the mass". This particular lexical extension can be found in many nouns, and resurfaced in Briscoe and Copestake's "grinding rule" (1989) that added a mass substance sense for all animals, as in their "rabbit all over the road" example. The argument was that, if such extensions were systematic, they need not be stored individually but could be developed when needed unless explicitly overridden. The paradigm for this was the old AI paradigm of default reasoning: Clyde is an elephant and all elephants have four legs BUT Clyde has three legs, and the latter fact must take precedence over the former inference. It has been some thing of a mystery why this foundational cliche of AI was greeted later within computational linguistics as remarkable and profound.

Gazdar's DATR is the system that makes lexical compression the most explicit, drawing as it does on fundamental notions of science as a compression of the data of the world. The problem has been that language is one of the most recalcitrant aspects of the world and it has proved hard to find generalisations above the level of morphology in DATR; those to do with meaning have proved especially elusive. Most recently, there has been an attempt to generalise DATR to cross-language generalisations which has exacerbated the problem. One can see that, in English, Dutch and German, respectively, "house", "huis" and "Haus" are the "same word", a primitive concept DATR seems to require. But, whereas "house" has a regular plural, "Haus" ("Haeuser") does not, so even at this low level, significant generalisations are very hard to find.

Most crucially, there can be no appeals to meaning from the concept of "same word": "town" (Eng.) and "tuin" (Dut.) are plainly the same word in some sense, at least etymologically and phonetically, and may well obey morphological generalisations although now, unlike the "house" cases above, they have no relation of meaning at all, as "tuin" now means garden in Dutch, unless one is prepared to move to some complex historical fable about related "spaces surrounded by a fence". There has been no attempt to link DATR to established quantitative notions of data compression in linguistics, like Minimum Description Length (Risannen 1981) which gives a precise measure of the compaction of a lexicon, even where significant generalisations may be hard to spot by eye or mind, in the time honoured manner.

The systems which seek lexical compression by means of rules, in one form or another, can be discussed by particular attention to Buitelaar, since Briscoe and Pustejovsky differ in matters of detail and rule format but not in principle. Buitelaar continues Pustejovsky's campaign against the "unstructured list" view of lexicons: viewing the senses of a word merely as a list as dictionaries are said to do, in favour of a clustered approach, one which distinguishes "systematic polysemy" from mere homonymy (like the ever present senses of "bank").

Clustering a word's senses in an optimally revealing way is something no one could possibly object to, and the problem here is the examples Buitelaar produces, and in particular his related attack on WSD programs (including the present authors) as assuming a list-view of sense, is misguided. As Nirenburg and Raskin (1997) have pointed out in relation to Pustejovksy, those who criticise list views of sense then normally go on in their papers to describe and work with the senses of a word as a list, and Buitelaar continues this tradition. Moreover, it must be pointed out that opening any modern English dictionary, especially one for learners like LDOCE, shows quite a complex grouping of the senses it contains and not a list at all.

Buitelaar's opening argument against standard WSD activities rests on his counter-example where two senses of "book" must be kept in play and so WSD should not be done: the example is "A long book heavily weighted with military technicalities, in this edition it is neither so long nor so technical as it was originally".

Leaving aside the question of whether this is a sentence, let us accept that Buitelaar's list (!) of possible senses (and glosses) of "book" is a reasonable starting point (with our numbering added):

(i) the information content of a book (military technicalities);
(ii) its physical appearance(heavily weighted),
(iii) and the events involved in its construction (long) (ibid. p. 25).

The issue, he says, is to which sense of "book" does the "it" refer, and his conclusion is that it cannot be disambiguated between the three.

This seems to us quite wrong, as a matter of the exegesis of the English text: "heavily weighted" is plainly metaphorical and refers to content (i) not the physical appearance (ii) of the book. We have no trouble taking "long" as referring to the content (i) since not all long books are physically large; it depends on the print size etc. On our reading, the "it" is univocal (to sense (i)) between the senses of "book". However, nothing depends on an example, well or ill-chosen,and it may well be that there are indeed cases where more than one sense must remain in play in a word's deployment; poetry is often cited, but there may well be others, less peripheral to the real world of the Wall Street Journal.

The main point in any answer to Buitelaar must be that, whatever is the case about the above example, WSD programs have no trouble capturing it: many programs, and certainly that of (Stevenson and Wilks, 1997) that he cites and its later developments, work by casting out senses and are perfectly able to report results with more than one sense still attaching to a word, just as many part-of-speech taggers result in more than one tag per word in the output. Historians of the AI approach to NLP will also remember that Mellish (1983), Hirst (1984) and Small (1988) all proposed methods by which polysemy might be computationally reduced by degree and not in an all or nothing manner. Or, as one might put it, underspecification, Buitelaar's key term, is no more than an implementation detail in any effective tagger!

Let us turn to the heart of Buitelaar's position: the issue of systematicity (one within which other closely related authors' claims about lexical rules can be taken together). Buitelaar lists clusters of nouns (e.g. blend, competition, flux, transformation) that share the same top semantic nodes in some structure like a modified WordNet (act/evt/rel in the case of the list just given).

Such structures, he claims, are manifestations of systematic polysemy but what is one to take that to mean, say by contrast with Levin's (1986) verb classes where, she claims, the members of a class share certain syntactic and semantic properties and, on that basis, one could in principle predict additional members? That is simply not the case here: one does not have to be a firm believer in natural kinds to see that the members of the cluster above have nothing systematic in common, but are just arbitrarily linked by the same "upper nodes" in Wordnet. Some such classes are natural classes, as with the one Buitelaar gives linked by being both animate and food (all of which, unsurprisingly, are animals and are edible, at least on some dietary principles), but there is no systemic relationship here of any kind. Or, to coin a phrase, one might say that the list above is just a list and nothing more!

In all this, we intend no criticism of his useful device, derived from Pustejovsky, for showing disjunctions and conjunctions of semantic types attached to lexical entries, as when one might mark something as act AND relation, or an animal sense as animate OR food. This is close to older devices in artificial intelligence such as multiple perspectives on structures (in Bobrow and Winograd's KRL 1967), and so on. Showing these situations as conjunctions and disjunctions of types may well be a superior notation, though it is quite proper to continue to point out that the members of conjuncts and disjuncts are, and remain, in lists!

Finally, Buitelaar's proposal to use these methods (via CoreLex) to acquire a lexicon from a corpus may also be an excellent approach, and one of the first efforts to link the LR movement to a corpus. It would probably fall under type II. acquisition (as defined earlier), and therefore not be LT, which rests essentially on structural modification by new data. Our point here is that that method (capturing the content of e.g. adjective-noun instances in a corpus) has no particular relationship to the theoretical machinery described above, and is not different in kind from the standard NLP type II. projects of the 1980s like Autoslog (Lehnert and Riloff 1987), to take just one of many possible examples.

7 Vagueness

The critique of the broadly positive position on WSD in this paper, and its relationship to LT, comes not only from those who argue (a) for the inadequacy of lexical sense sets over productive lexical rules (as above) but also from proponents of (b) the inherently VAGUE quality of the difference between senses of a given word. We believe both these approaches are muddled IF their proponents conclude that WSD is therefore fatally flawed as a task.

The vagueness issue is again an old observation, and one that, if taken seriously, must surely result in a statistical or fuzzy-logic approach to sense discrimination, since only probabilistic (or at least quantitative) methods can capture real vagueness. That, surely, is the point of the Sorites paradox: there can be no plausible or rational qualitatively-based criterion (which would include any quantitative system with clear limits: e.g. tall = over 6 feet) for demarcating "tall", "green" or any inherently vague concept.

If, however, sense sets/lists/inventories are to continue to play a role vagueness can mean no more than highlighting what all systems of WSD must have, namely some parameter or threshold for the assignment to one of a list of senses versus another, or setting up a new sense in the list. Talk of vagueness adds nothing specific to help that process for those who want to assign,on some quantitative basis, to one sense rather than another; it is the usual issue of tuning to see what works and fits our intuitions.

Vagueness would be a serious concept only if the whole sense list for a word (in rule form or not) was abandoned in favour of statistically-based clusters of usages or contexts. There have been just such approaches to WSD in recent years (e.g. Bruce and Wiebe, 1994, Pedersen and Bruce, 1997, Schuetze & Pederson, 1995) and the essence of the idea goes back to Sparck Jones 1964/1986) but such an approach would find it impossible to take part in any competition like SENSEVAL (Kilgarriff, 1998) because it would inevitably deal in nameless entities which cannot be marked up for.

Vague and Lexical Rule based approaches also have the consequence that all lexicographic practice is, in some sense, misguided: dictionaries for such theories are fraudulent documents that could not help users whom they systematically mislead by listing senses. Fortunately, the market decides this issue, and it is a false claim. Vagueness in WSD is either false (the last position) or trivial, and known and utilised within all methodologies.

8 Lexical Rules and Pre-markup

Can the lexical rules approach to some of the phenomena discussed here be made evaluable, using some conventional form of pre-markup, in the way that we saw is difficult for straightforward LT of new senses, but which may be possible if LT makes use of some form of the "closest sense" heuristic? The relevance of this to the general WSD and tuning discussion is that the very idea of pre-markup would presumably require that all lexical rules are run, so that the human marker can see the full range of senses available, which some might feel inconsistent with the core data compression notion behind lexical rules. However, this situation is no different in principle from POS tagging where a language, say English, may well have a tag meta-rule that any word with N in its tag-lexicon could also have the tag ADJ (but not vice versa). Clearly any such rule would have to be run before pre-markup of text could be done, and the situation with senses is no different, though psychologically for the marker it may seem so, since the POS tag inventory can usually be kept in memory, whereas a sense inventory for a vocabulary cannot.

9 Conclusion: Which of These Methods Lead to Evaluation?

What is the conclusion here on the relationship of lexical extension, in whatever form, to the task of WSD, given that the thrust of the paper has been to see if the now evaluable methods of WSD apply to LT, and can be adapted to make it evaluable too? It is clear that the LR approach, at least as represented by Buitelaar, sees no connection and believes WSD to be a misleading task. And this is not a shocking result, for it only brings out in a new way the division that underlies this paper, and is as old as the generative vs. corpus linguistics divide, one that has existed for decades but was effectively disguised by the denial by the dominant Chomskyan generative paradigm that anything akin to corpus linguistics existed.

Our reply to this is that Buitelaar's examples do not support his attack on WSD, since underspecification is largely a misnomer. Corpora could be premarked for the senses coded in such a lexicon, if treated as real disjunctions, but there is no way of knowing which of these are attested or attestable in data and we argued that the two aspects of the key example "bake" are not in fact related to sense distinction or polysemy phenomena at all.

On the other hand, the method A phenomena are impossible to premark and therefore could be tested only within a final task like IE, IR or MT. The relaxation phenomena of method B, on the other hand, could possibly be premarked for (and then tested as part of a WSD program) but by doing so do not constitute extended sense phenomena, like LT, at all, since by relaxing to an existing sense one denies a new sense is in operation. In the B2 type cases, with data like that of the LR researchers, the extension of "prepare" to "bake" (of bread) should result in the representation of "bake" being added as possible sense of "prepare" (by analogy with Method A) whether or not this effects a one-off or permanent (LT) adaptation.

There is some evidence for the positive evaluation of tasks of a WSD/LT type within what we have called "final" (as opposed to intermediate) tasks: within the SPARKLE project Grefenstette (1998) produced a form of lexical augmentation that improved overall information retrieval precision and recall by a measurable amount. It is most important to keep some outcome like this in mind as an active research goal if the markup paradigm becomes impossible for LT, because our aim here is to separate clearly here evaluable approaches from weaker notions of computer-related lexical work.

Acknowledgments

The authors are much indebted to comments from Evelyne Viegas, Steve Helmreich and Wim Peters, but the errors are their own, as always.The work has been supported by the UK EPSRC grant MALT, and the EC IST HLT projects ECRAN and NAMIC.

References

1. Bruce, R. and Wiebe, J. (1994) Word-sense disambiguation using decomposable models, Proc. ACL-94.
2. Dini, L., di Tommaso, V. and Segond, F. (1998) Error-driven word sense disambiguation. In Proc. COLING-ACL98, Montreal.
3. Fodor, J. and Lepore E. (2000). The emptiness of the Lexicon: critical reflections on J. Pustejovsky's The Generative Lexicon. In Bouillon and Busa (eds.) Meaning and the Lexicon. New York: Crowell.
4. Givon, T. (1967) Transformations of Ellipsis, Sense Development and Rules of Lexical Derivation. SP-2896, Systems Development Corp., Sta. Monica, CA.
5. Green, G. (1989) Pragmatics and Natural Language Understanding. Erlbaum: Hillsdale, NJ.
6. Hirst, G. (1987) Semantic Interpretation and the Resolution of Ambiguity, CUP, Cambridge, England.
7. Jorgensen, J. (1990) The psychological reality of word senses, Journal of Psycholinguistic Research, vol 19.
8. Kilgarriff, A. (1993) Dictionary word-sense distinctions: an enquiry into their nature, Computers and the Humanities, vol 26.
9. Knight, K. and Luk, S. (1994) Building a Large Knowledge Base for Machine Tanslation, Proceedings of the American Association for Artificial Intelligence Conference AAAI-94, pp. 185-109, Seattle, WA.
10. Mellish, C. (1983) Incremental semantic interpretation in a modular parsing system. In Karen Sparck-Jones and Yorick A. Wilks (eds.) Automatic Natural Language Parsing, Ellis Horwood/Wiley, Chichester/NYC.
11. Nirenburg, S. and Raskin., V. (1997) Ten choices for lexical semantics. Research Memorandum, Computing Research Laboratory, Las Cruces, NM.
12. Pedersen, T. and Bruce, R. (1997) Distinguishing Word Senses in Untagged Text, Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 197-207, Providence, RI.
13. Pustejovsky, J. (1995) The Generative Lexicon, MIT Press: Cambridge, MA.
14. Resnik, P. and Yarowsky, D. (1997) A Perspective on Word Sense Disambiguation Techniques and their Evaluation, Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics: What, why and how?", pp. 79-86, Washington, D.C.
15. Schank, R. and Abelson R.P. (1977). Scripts, Plans, Goals and Understanding, Hillsdale NJ: Lawrence Erlbaum.
16. Schutze, H. (1992) Dimensions of Meaning, Proceedings of Supercomputing '92, pp. 787-796, Minneapolis, MN.
17. Schutze, H. and Pederson, J. (1995) Information Retrieval based on Word Sense, Proc. Fourth Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, NV.

18. Small, S., Cottrell, G., and Tanenhaus, M. (Eds.) (1988) Lexical Ambiguity Resolution, Morgan Kaufmann: San Mateo, CA.
19. Sparck Jones, K. (1964/1986) Synonymy and Semantic Classification. Edinburgh UP: Edinburgh.
20. Wilks, Y. (1968) Argument and Proof. Cambridge University PhD thesis.
21. Wilks, Y. (1980) Frames, Semantics and Novelty. In D. Metzing (ed.), Frame Conceptions and Text Understanding, Berlin: de Gruyter.
22. Wilks, Y. (1997) Senses and Texts. Computers and the Humanities.
23. Wilks, Y. and Stevenson, M. (1998a) The Grammar of Sense: Using part-of-speech tags as a first step in semantic disambiguation, Journal of Natural Language Engineering, 4(1), pp. 1-9.
24. Wilks, Y. and Stevenson, M. (1998b) Optimising Combinations of Knowledge Sources for Word Sense Disambiguation, Proceedings of the 36th Meeting of the Association for Computational Linguistics (COLING-ACL-98), Montreal,
25. Wilks, Y. and Stevenson, M. (2001) Word sense disambiguation using multiple methods. Computational Linguistics.
26. Yarowsky, D. (1995) Unsupervised Word-Sense Disambiguation Rivaling Supervised Methods, Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95), pp. 189-196, Cambridge, MA

A Baseline Methodology for Word Sense Disambiguation

Ted Pedersen

University of Minnesota, Duluth, MN 55812, USA
tpederse@d.umn.edu
WWW home page: http://www.d.umn.edu/~tpederse

Abstract. This paper describes a methodology for supervised word sense disambiguation that relies on standard machine learning algorithms to induce classifiers from sense-tagged training examples where the context in which ambiguous words occur are represented by simple lexical features. This constitutes a baseline approach since it produces classifiers based on easy to identify features that result in accurate disambiguation across a variety of languages. This paper reviews several systems based on this methodology that participated in the Spanish and English lexical sample tasks of the SENSEVAL-2 comparative exercise among word sense disambiguation systems. These systems fared much better than standard baselines, and were within seven to ten percentage points of accuracy of the mostly highly ranked systems.

1 Introduction

Word sense disambiguation is the process of selecting the most appropriate meaning for a word, based on the context in which it occurs. We assume that a sense inventory or set of possible meanings is provided by a dictionary, so disambiguation occurs by choosing a meaning for a word from this finite set of possibilities.

Humans are able to determine the intended meanings of words based on the surrounding context, our understanding of language in general, and our knowledge of the real world. In fact, we usually arrive at the correct interpretation of a sentence without even considering the full set of possible meanings associated with a word. For example, in *He showed an interest in the new line of tailored suits*, a human reader immediately knows that *interest* refers to an appreciation, *line* to products, and *suits* to men's clothing. It is unlikely that a fluent speaker of English would consider alternative interpretations relating to interest rates, telephone lines, or playing cards. However, a computer program will have a difficult time making these kinds of distinctions, since it has much less knowledge of language and the world.

We take a *corpus–based* approach to this problem and learn a classifier from a corpus of sense–tagged sentences, where a human expert has manually annotated each occurrence of a word with the most appropriate sense for the given context. Such sense–tagged text is difficult to create in large quantities, but once available it provides strong evidence that allows a supervised learning algorithm

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 126–135, 2002.

to build a classifier that can recognize the patterns in the context surrounding an ambiguous word that are indicative of its sense. This classifier is then used to assign senses to that word when it is encountered again outside of the training examples, as would be the case when processing a held–out set of test instances.

For supervised learning we rely on *Naive Bayesian classifiers* and *decision trees*. These are widely used and relatively simple algorithms that have been applied in many different settings, and as such represent good choices for a baseline approach. Sense–tagged sentences are converted into a feature space that represents the context in which an ambiguous word occurs strictly in terms of *unigrams*, *bigrams*, and *co–occurrences*. Unigrams and bigrams are one and two word sequences that occur anywhere in the context with the ambiguous word, and co–occurrences are bigrams that include the word to be disambiguated. These are easy to identify features that are known to contribute to word sense disambiguation, and as such are a reasonable choice as a baseline set of features.

We have found this combination of machine learning algorithms and lexical features to result in surprisingly effective disambiguation in both Spanish and English, suggesting that this methodology is both robust and accurate. This represents a substantial improvement over standard baseline algorithms such as the majority classifier, which simply determines the most frequent sense of a word in the training data and applies that to every instance in the test data.

2 The Senseval-2 Exercise

The Senseval-2 exercise took place in May–July 2001, and brought together about 35 teams from around the world. There are two main tasks in Senseval; an all-words task where every content word in a corpus of text is to be disambiguated, and a lexical sample task where every occurrence of a particular set of words is to be disambiguated. Our systems, known collectively as the Duluth systems, participated in the English and Spanish lexical sample tasks.

The objective of Senseval is to provide a forum where word sense disambiguation systems can be evaluated in a fair and neutral fashion. This is achieved by carrying out a blind evaluation based on sense–tagged text specifically created for the exercise. In the lexical sample tasks, each team has access to sense–tagged training examples for two weeks, during which time they can build models or classifiers based on that data. After this two week period, teams have one week to sense–tag a set of test instances and return their results for scoring.

A lexical sample is created for a particular set of words, and provides multiple examples of each word in naturally occurring contexts that include the sentence in which the word occurs plus two or three surrounding sentences. Training examples are created by manually annotating each occurrence of the words in the lexical sample with a sense–tag that indicates which meaning from the sense inventory is most appropriate. In Senseval-2 the English sense inventory was defined by the lexical database WordNet, and the sense inventory for Spanish was defined by Euro–WordNet.

Most occurrences of a word are well defined by a single meaning and have one sense–tag. However, there are a few occurrences where multiple senses are equally appropriate, and these will have multiple sense–tags. In such cases each of these meanings is considered equally valid, so we generate a separate training example for each sense–tag. This leads to slightly more training examples than there are sense–tagged sentences. However, this only impacts classifier learning. Feature selection is based on the original sense–tagged sentences without regard to the number of possible senses of an occurrence.

The English lexical sample consists of 73 words, where there are 9,430 sense–tagged sentences which result in 9,536 training examples. There are 4,328 held–out test instances to be assigned senses. There are an average of nine senses per word in the test instances. The words in the lexical sample are listed below according to their part of speech, and are followed by the number of training examples and test instances.

Nouns: art (252, 98), authority (222, 92), bar (362, 151), bum (99, 45), chair (143, 69), channel (209, 73), child (135, 64), church (153, 64), circuit (182, 85), day (329, 145), detention (70, 32), dyke (84, 28), facility (121, 58), fatigue (89, 43), feeling (116, 51), grip (129, 51), hearth (71, 32), holiday (68, 31), lady (122, 53), material (150, 69), mouth (149, 60), nation (96, 37), nature (103, 46), post (176, 79), restraint (142, 45), sense (111, 53), spade (73, 33), stress (94, 39), yew (60, 28)

Verbs: begin (563, 280), call (143, 66), carry (134, 66), collaborate (57, 30), develop (135, 69), draw (83, 41), dress (122, 59), drift (64, 32), drive (85, 42), face (193, 93), ferret (2, 1), find (132, 68), keep (135, 67), leave (132, 66), live (131, 67), match (88, 42), play (129, 66), pull (122, 60), replace (86, 45), see (132, 69), serve (100, 51), strike (104, 54), train (190, 63), treat (91, 44), turn(132, 67), use (148, 76), wander (100, 50), wash (26, 12), work (122, 60)

Adjectives: blind (127, 55), colourless (72, 35), cool (127, 52), faithful (50, 23), fine (181, 70), fit (63, 29), free (196, 82), graceful (62, 29), green (212, 94), local (78, 38), natural (243, 103), oblique (64, 29), simple (135, 66), solemn (54, 25), vital (81, 38)

The Spanish lexical sample consists of 39 words. There are 4,480 sense–tagged sentences that result in 4,535 training examples. There are 2,225 test instances that have an average of five senses per word. The words in the lexical sample are listed below along with the number of training examples and test instances.

Nouns: autoridad (90, 34), bomba (76, 37), canal (115, 41), circuito (74, 49), corazón (121, 47), corona (79, 40), gracia (103, 61), grano (56, 22), hermano (84, 57), masa (91, 41), naturaleza (113, 56), operación (96, 47), órgano (131, 81), partido (102, 57), pasaje (71, 41), programa (98, 47), tabla (78, 41)

Verbs: actuar (100, 55), apoyar (137, 73), apuntar (142, 49), clavar (87, 44), conducir (96, 54), copiar (95, 53), coronar (170, 74), explotar (92, 41), saltar (101, 37), tocar (162, 74), tratar (124, 70), usar (112 56), vencer (120, 65)

Adjectives: brillante (169, 87), claro (138, 66), ciego (72, 42), local (88, 55), natural (79, 58), popular (457, 204), simple (160, 57), verde (78, 33), vital (178, 79)

3 Lexical Features

The word sense disambiguation literature provides ample evidence that many different kinds of features contribute to the resolution of word meaning (e.g., [3], [5]). These include part–of–speech, morphology, verb–object relationships, selectional restrictions, lexical features, etc. When used in combination it is often unclear to what degree each type of feature contributes to overall performance. It is also unclear to what extent adding new features allows for the disambiguation of previously unresolvable test instances. One of the long term objectives of our research is to determine which types of features are complementary and contribute to disambiguating increasing numbers of test instances as they are added to a representation of context. The methodology described here is a part of that effort, and is intended to measure the limits of lexical features.

Here the context in which an ambiguous word occurs is represented by some number of binary features that indicate whether or not particular unigrams, bigrams, or co–occurrences have occurred in the surrounding text. Our interest in simple lexical features, particularly co–occurrences, has been inspired by [1], which shows that humans determine the meaning of ambiguous words largely based on words that occur within one or two positions to the left and right. They have the added advantage of being easy to identify in text and therefore provide a portable and convenient foundation for baseline systems.

These features are identified using the Bigram Statistics Package (BSP) version 0.4. Each unigram, bigram, or co-occurrence identified in the training examples is treated as a binary feature that indicates whether or not it occurs in the context of the word being disambiguated. SenseTools version 0.1 converts training and test data into a feature vector representation, based on the output from BSP. This becomes the input to the Weka[10] suite of supervised learning algorithms, which induces a classifier from the training examples and applies sense–tags to a set of test instances. All of this is free software that is available from the following sites:

BSP, SenseTools: http://www.d.umn.edu/˜tpederse/code.html.

Weka: http://www.cs.waikato.ac.nz/˜ml

4 Machine Learning Algorithms

Supervised learning is the process of inducing a model to perform a task based on a set of examples where a human expert has manually indicated the appropriate outcome. Depending on the task, this might be a diagnosis, a classification, or a prediction. We cast word sense disambiguation as a classification problem, where a word is assigned the most likely sense based on the context in which it occurs.

While there are many supervised learning algorithms, we have settled upon two widely used approaches, decision trees and Naive Bayesian classifiers. Both have been used in a wide range of problems, including word sense disambiguation (e.g., [4], [9]).These are complementary approaches to supervised learning that differ in their *bias* and *variance* characteristics.

Decision tree learning is based on a general to specific search of the feature vector representation of the training examples in order to select a minimal set of features that efficiently partitions the feature space into classes of observations and assemble them into a tree. In our case, the observations are manually sense–tagged examples of an ambiguous word in context and the partitions correspond to the different possible senses. This process is somewhat unstable in that minor variations in the training examples can cause radically different trees to be learned. As a result, decision trees are said to be a low bias, high variance approach.

Each feature selected during the search process is represented by a node in the learned decision tree. Each node represents a choice point between a number of different possible values for a feature. Learning continues until all the training examples are accounted for by the decision tree. In general, such a tree will be overly specific to the training data and not generalize well to new examples. Therefore learning is followed by a pruning step where some nodes are eliminated or reorganized to produce a tree that can generalize to new circumstances.

Test instances are disambiguated by finding a path through the learned decision tree from the root to a leaf node that corresponds with the observed features. In effect an instance of an ambiguous word is disambiguated by passing it through a series of tests, where each test asks if a particular lexical feature occurs nearby. We use the Weka decision tree learner J48, which is a Java implementation of the C4.5 decision tree learner. We use the default parameter settings for pruning.

A Naive Bayesian classifier [2] is a probabilistic model that assigns the most likely sense to an ambiguous word, based on the context in which it occurs. It is based on a blanket assumption about the interactions among the features in a set of training examples that is generally not true in practice but still can result in an accurate classifier. The underlying model holds that all features are conditionally independent, given the sense of the word. In other words, features only directly affect the sense of the word and not each other.

Since the structure of the model is already assumed, there is no need to perform a search through the feature space as there is with a decision tree. As such the learning process only consists of estimating the probabilities of all the pairwise combinations of feature and sense values. Since it is not attempting to characterize relationships among features in the training data, this method is very robust and is not affected by small variations in the training data. As such it is said to be a high bias, low variance approach. We use the Weka implementation of the Naive Bayesian classifier with the default parameter settings.

5 System Descriptions

This section discusses the Duluth systems in the English and Spanish lexical sample tasks. We refer to them as system pairs since the only differences between the English and Spanish versions of a system are the tokenizers and stop–lists. In both languages tokens are made up of alphanumeric strings, and exclude punctuation. There is a stop–list for each language that is created by selecting five different sets of training examples, where each set is associated with a different word in the lexical sample and has approximately the same number of total words. The stop–list is made up of all words that occur ten or more times in each of the five sets of training examples. Stop–listed words are always excluded as unigram features, and any bigram that is made up of two stop–listed words is also excluded as a feature. Since co–occurrences always include the ambiguous word, they are not subjected to stop–listing.

All experimental results are presented in terms of fine-grained accuracy, which is calculated by dividing the number of correctly disambiguated test instances by the total number of test instances. Of the 20 English lexical sample systems that participated in SENSEVAL-2, the highest ranked achieved accuracy of 64% over the 4,328 test instances. The highest ranked of the 12 Spanish systems achieved accuracy of 68% on the 2,225 test instances. The most accurate Duluth system in English and Spanish ranked seventh and fourth, with accuracy of 59% and 61%, respectively.

There were eight Duluth systems in SENSEVAL-2, five of which are discussed here. In the following, the name of the English system appears first, followed by the Spanish system. The accuracy attained by each is shown in parenthesis.

Duluth1(53%)/Duluth6(58%) is an ensemble of three Naive Bayesian classifiers, where each is based on a different feature set representation of the training examples. The hope is that these different views of the training examples will result in classifiers that make complementary errors, and that their combined performance will be better than any of the individual classifiers.

Separate Naive Bayesian classifiers are learned from each representation of the training examples. Each classifier assigns probabilities to each of the possible senses of a test instance. We take a *weighted vote* by summing the probabilities of each possible sense and the one with the largest value is selected. In the event of ties multiple senses are assigned.

The first feature set is made up of bigrams that can occur anywhere in the context with the ambiguous word. To be selected as a feature, a bigram must occur two or more times in the training examples and have a log–likelihood ratio ≥ 6.635, which has an associated p–value of .01. The second feature set consists of unigrams that occur five or more times in the training data. The third feature set is made up of co-occurrence features that represent words that occur to the immediate left or right of the target word. In effect, these are bigrams that include the target word. They must also occur two or more times and have a log–likelihood ratio ≥ 2.706, which has an associated p–value of .10.

These systems are inspired by [6], which presents an ensemble of eighty-one Naive Bayesian classifiers based on varying sized windows of context to the left and right of the target word that define co-occurrence features. Here we only use a three member ensemble in order to preserve the portability and simplicity of a baseline approach.

Duluth2(54%)/Duluth7(60%) is a *bagged* decision tree that is learned from a sample of training examples that are represented in terms of the bigrams that occur two or more times and have a log–likelihood ratio ≥ 6.635.

Bagging is an ensemble technique that is achieved by drawing ten samples, with replacement, from the training examples. A decision tree is learned from each of these permutations of the training examples, and each of these trees becomes a member of the ensemble. A test instance is assigned a sense based on a majority vote among the ten decision trees. The goal of bagging is to smooth out the instability inherent in decision tree learning, and thereby lower the variance caused by minor variations in the training examples.

This bigram feature set is one of the three used in the Duluth1/Duluth6 systems. In that case every bigram meeting the criteria is included in the Naive Bayesian classifier. Here, the set of bigrams that meet these criteria become candidate features for the J48 decision tree learning algorithm, which first constructs a tree that characterizes the training examples exactly, and then prunes nodes away to avoid over–fitting and allow it to generalize to previously unseen test instances. Thus, the learned decision tree performs a second cycle of feature selection that removes some of the features that meet the criteria described above. As such the decision tree learner is based on a smaller number of features than the Naive Bayesian classifier.

This system pair is an extension of [7], which learns a decision tree where the representation of context consists of the top 100 bigrams according to the log–likelihood ratio. This earlier work does not use bagging, and just learns a single decision tree.

Duluth3(57%)/Duluth8(61%) is an ensemble of three bagged decision trees using the same features as Duluth1/Duluth6. A bagged decision tree is learned based on unigram features, another on bigram features, and a third on co–occurrences. The test instances are classified by each of the bagged decision trees, and a weighted vote is taken to assign senses to the test instances.

These are the most accurate of the Duluth systems for both English and Spanish. These are within 7% of the most accurate overall approaches for English (64%) and Spanish (68%).

One of the members of this ensemble is a bagged decision tree based on bigrams that is identical to the Duluth2/Duluth7 systems, which attains accuracy of 54% and 60%. Thus, the combination of the bigram decision tree, with two others based on unigrams and co–occurrences, improves accuracy by about 3% for English and 1% for Spanish. These minimal increases suggest that the members of the ensemble are largely redundant.

Duluth4(54%)/Duluth9(56%) is a Naive Bayesian classifier using a feature set of unigrams that occur five or more times in the English training examples. In the Spanish examples a unigram is a feature if it occurs two or more times. These features form the basis of the Naive Bayesian classifier, which will assign the most probable sense to a test instance, given the context in which it occurs.

This system pair is one of the three member classifiers that make up the ensemble approach of Duluth1/Duluth7, which consists of three Naive Bayesian classifiers, one based on unigrams, another on bigrams, and a third on co–occurrences. This ensemble is 1% more accurate for the English lexical sample than the single Naive Bayesian classifier based on unigrams, and 2% less accurate for the Spanish. This is one of the few cases where the performance of the English and Spanish systems diverged, although the difference in performance between the single Naive Bayesian classifier and the ensemble is relatively slight and suggests that each of these classifiers is largely redundant of the other.

DuluthB(51%)/DuluthY(52%) is a *decision stump* learned from a representation of the training examples that is based on bigrams and co–occurrences. Bigrams must occur two or more times and have a log–likelihood ratio ≥ 6.635, and co–occurrences must occur two or more times and have a log–likelihood ratio ≥ 2.706. A decision stump is simply a one–node decision tree where learning is stopped after the root node is found by identifying the single feature that is best able to discriminate among the senses. A decision stump will at worst reproduce the majority classifier, and may do better if the selected feature is particularly informative.

Decision stumps are the least accurate of the Duluth systems for both English and Spanish, but are more accurate than the majority classifier for English (48%) and Spanish (47%).

6 Discussion

The fact that a number of related systems are included in these experiments makes it possible to examine several hypotheses that motivate our overall research program in word sense disambiguation.

6.1 Features Matter Most

This hypothesis holds that variations in learning algorithms matter far less to disambiguation performance than do variations in the features used to represent the context in which an ambiguous word occurs. In other words, an informative feature set will result in accurate disambiguation when used with a wide range of learning algorithms, but there is no learning algorithm that can overcome the limitations of an uninformative or misleading set of features.

This point is clearly made when comparing the systems Duluth1/Duluth6 and Duluth3/Duluth8. The first pair learns three Naive Bayesian classifiers and the second learns three bagged decision trees. Both use the same feature set to represent the context in which ambiguous words occur. There is a 3% improvement in accuracy when using the decision trees. We believe this modest improvement when moving from a simple learning algorithm to a more complex one supports the hypothesis that significant improvements are more likely to be attained by refining the feature set rather than tweaking a supervised learning algorithm.

6.2 50/25/25 Rule

We hypothesize that in a set of test instances about half are fairly easy to disambiguate, another quarter is harder, and the last quarter is nearly impossible. In other words, almost any classifier induced from a sample of sense–tagged training examples will have a good chance of getting at least half of the test instances correct. As classifiers improve they will be able to get up to another quarter of the test instances correct, and that regardless of the approach there will remain a quarter that will be difficult to disambiguate. This is a variant of the 80/20 rule of time management, which holds that a small amount of the total effort accounts for most of the results.

Comparing the two highest ranking systems in the English lexical sample task, SMUls and JHU(R), provides evidence in support of this hypothesis. There are 2180 test instances (50%) that both systems disambiguate correctly. There are an additional 1183 instances (28%) where one of the two systems are correct, and 965 instances (22%) that neither system can resolve. If these two systems were optimally combined, their accuracy would be 78%. If the third-place system is also considered, there are 1939 instances (44.8%) that all three systems can disambiguate, and 816 (19%) that none could resolve.

When considering all eight of the Duluth systems that participated in the English lexical sample task, there are 1705 instances (39%) that all disambiguated correctly. There are 1299 instances (30%) that none can resolve. The accuracy of an optimally combined system would be 70%. The most accurate individual system is Duluth3 with 57% accuracy.

For the Spanish Duluth systems, there are 856 instances (38%) that all eight systems got correct. There are 478 instances (21%) that none of the systems got correct. This results in an optimally combined result of 79%. The most accurate Duluth system was Duluth8, with 1369 correct instances (62%). If the top ranked Spanish system (68%) and Duluth8 are compared, there are 1086 instances (49%) where both are correct, 737 instances (33%) where one or the other is correct, and 402 instances (18%) where neither system is correct.

This is intended as a rule of thumb, and suggests that a fairly substantial percentage of test instances can be resolved by almost any means, and that a hard core of test instances will be very difficult for any method to resolve.

6.3 Language Independence

We hypothesize that disambiguation via machine learning and lexical features is language independent. While English and Spanish are too closely related to draw general conclusions, the results are at least indicative. For both the English and Spanish tasks, the ensembles of bagged decision trees are the most accurate systems (Duluth3/Duluth8). The next most accurate systems in both languages are Duluth5/Duluth10, bagged decision trees based on bigram and co-occurrence features. The least accurate for both languages is the decision stump (DuluthB/DuluthY). In general system pairs perform at comparable levels of accuracy for both Spanish and English.

7 Conclusions

This paper presents a baseline methodology for word sense disambiguation that relies on simple lexical features and standard machine learning algorithms. This approach was evaluated as a part of the SENSEVAL-2 comparative exercise among word sense disambiguation systems, and was within seven to ten percentage points of accuracy of the most highly ranked systems.

Acknowledgments

This work has been partially supported by a National Science Foundation Faculty Early CAREER Development award (#0092784) and by a Grant–in–Aid of Research, Artistry and Scholarship from the Office of the Vice President for Research and the Dean of the Graduate School of the University of Minnesota.

We are grateful to the SENSEVAL-2 coordinators for making the data and results from this event available in the public domain. The Bigram Statistics Package and SenseTools have been implemented by Satanjeev Banerjee. A preliminary version of this paper appears in [8].

References

1. Y. Choueka and S. Lusignan. Disambiguation by short contexts. *Computers and the Humanities*, 19:147–157, 1985.
2. R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.
3. S. McRoy. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1):1–30, 1992.
4. R. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–91, May 1996.
5. H.T. Ng and H.B. Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, 1996.
6. T. Pedersen. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 63–69, Seattle, WA, May 2000.
7. T. Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86, Pittsburgh, July 2001.
8. T. Pedersen. Machine learning with lexical features: The Duluth approach to SENSEVAL-2. In *Proceedings of the* SENSEVAL-2 *Workshop*, Toulouse, July 2001.
9. T. Pedersen and R. Bruce. A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 604–609, Providence, RI, July 1997.
10. I. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan–Kaufmann, San Francisco, CA, 2000.

An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet

Satanjeev Banerjee and Ted Pedersen

University of Minnesota, Duluth, MN 55812, USA
{bane0025,tpederse}@d.umn.edu
http://www.d.umn.edu/{~bane0025,~tpederse}

Abstract. This paper presents an adaptation of Lesk's dictionary–based word sense disambiguation algorithm. Rather than using a standard dictionary as the source of glosses for our approach, the lexical database WordNet is employed. This provides a rich hierarchy of semantic relations that our algorithm can exploit. This method is evaluated using the English lexical sample data from the SENSEVAL-2 word sense disambiguation exercise, and attains an overall accuracy of 32%. This represents a significant improvement over the 16% and 23% accuracy attained by variations of the Lesk algorithm used as benchmarks during the SENSEVAL-2 comparative exercise among word sense disambiguation systems.

1 Introduction

Most words in natural languages are *polysemous*, that is they have multiple possible meanings or *senses*. For example, *interest* can mean a charge for borrowing money, or a sense of concern and curiosity. When using language humans rarely stop and consider which sense of a word is intended. For example, in *I have an interest in the arts*, a human reader immediately knows from the surrounding context that *interest* refers to an appreciation, and not a charge for borrowing money.

However, computer programs do not have the benefit of a human's vast experience of the world and language, so automatically determining the correct sense of a polysemous word is a difficult problem. This process is called *word sense disambiguation*, and has long been recognized as a significant component in language processing applications such as information retrieval, machine translation, speech recognition, etc.

In recent years corpus–based approaches to word sense disambiguation have become quite popular. In general these rely on the availability of manually created *sense–tagged* text, where a human has gone through a corpus of text, and labeled each occurrence of a word with a tag that refers to the definition of the word that the human considers most appropriate for that context. This sense–tagged text serves as training examples for a supervised learning algorithm that can induce a classifier that can then be used to assign a sense–tag to previously unseen occurrences of a word. The main difficulty of this approach is that

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 136–145, 2002.

sense–tagged text is expensive to create, and even once it exists the classifiers learned from it are only applicable to text written about similar subjects and for comparable audiences.

Approaches that do not depend on the existence of manually created training data are an appealing alternative. An idea that actually pre–dates most work in corpus–based approaches is to take advantage of the information available in machine readable dictionaries. The Lesk algorithm [3] is the prototypical approach, and is based on detecting shared vocabulary between the definitions of words. We adapt this algorithm to WordNet [2], which is a lexical database structured as a semantic network.

This paper continues with a description of the original Lesk algorithm and an overview of WordNet. This is followed by a detailed presentation of our algorithm, and a discussion of our experimental results.

2 The Lesk Algorithm

The original Lesk algorithm [3] disambiguates words in short phrases. The definition, or *gloss*, of each sense of a word in a phrase is compared to the glosses of every other word in the phrase. A word is assigned the sense whose gloss shares the largest number of words in common with the glosses of the other words. For example, in *time flies like an arrow*, the algorithm compares the glosses of *time* to all the glosses of *fly* and *arrow*. Next it compares the glosses of *fly* with those of *time* and *arrow*, and so on. The algorithm begins anew for each word and does not utilize the senses it previously assigned.

The original Lesk algorithm relies on glosses found in traditional dictionaries such as Oxford Advanced Learner's. We modify Lesk's basic approach to take advantage of the highly inter–connected set of relations among synonyms that WordNet offers. While Lesk's algorithm restricts its comparisons to the glosses of the words being disambiguated, our approach is able to compare the glosses of words that are related to the words to be disambiguated. This provides a richer source of information and improves overall disambiguation accuracy. We also introduce a novel scoring mechanism that weighs longer sequences of matches more heavily than single words.

3 About WordNet

While traditional dictionaries are arranged alphabetically, WordNet is arranged semantically, creating an electronic lexical database of nouns, verbs, adjectives, and adverbs. Synonymous words are grouped together to form synonym sets, or *synsets*. A word is polysemous if it occurs in several synsets, where each synset represents a possible sense of the word. For example *base* occurs in two noun synsets, {*base, alkali*} and {*basis, base, foundation, fundament, groundwork, cornerstone*}, and the verb synset {*establish, base, ground, found*}.

In WordNet version 1.7 there are 107,930 nouns arranged in 74,448 synsets, 10,860 verbs in 12,754 synsets, 21,365 adjectives in 18,523 synsets, and 4,583 adverbs in 3,612 synsets. Function words such as *for*, *the*, *and*, etc. are not defined in WordNet. Our algorithm only disambiguates words that belong to at least one synset, which we call *WordNet words.*

Each synset has an associated definition or gloss. This consists of a short entry explaining the meaning of the concept represented by the synset. The gloss of the synset {*base, alkali*} is "any of various water-soluble compounds capable of turning litmus blue and reacting with an acid to form a salt and water", while that associated with {*basis, base, foundation, fundament, groundwork, cornerstone*} is "lowest support of a structure". Each synset can also be referred to by a unique identifier, commonly known as a *sense–tag.*

Synsets are connected to each other through a variety of semantic relations. With few exceptions these relations do not cross part of speech boundaries, so synsets are only related to other synsets that belong to the same part of speech. Here we review only those relations that have entered into the experiments presented in this paper. A complete description of all the relations can be found in [2].

For nouns, two of the most important relations are *hyponymy* and *hypernymy.* If synset A is *a kind of* synset B, then A is the hyponym of B, and B is the hypernym of A. For example, {*bed*} is a hyponym of {*basis, base, foundation, fundament, groundwork, cornerstone*}, and conversely, {*basis, base, foundation, fundament, groundwork, cornerstone*} is the hypernym of {*bed*}. Another pair of related relations for nouns is that of *holonymy* and *meronymy.* Synset A is a meronym of synset B if *A is a part of B*. Conversely, B is a holonym of A if *B has A as a part.* Thus {*structure, construction*} is a meronym of {*basis, base, foundation, fundament, groundwork, cornerstone*}, and {*basis, base, foundation, fundament, groundwork, cornerstone*} is a holonym of {*structure, construction*}.

Verbs are related through the relations *hypernymy* and *troponymy.* Synset A is the hypernym of B, if *B is one way to A*; B is then the troponym of A. Thus, the verb synset {*station, post, base, send, place*} is the troponym of {*move, displace*} since to {*station, post, base, send, place*} is one way to {*move, displace*}.

One of the few relations available for adjectives is *attribute* that relates an adjective to a noun. For example, the attribute of {*beautiful*} is the noun {*beauty*}. This is an unusual relation, in that it crosses part of speech boundaries to connect an adjective synset with a noun synset.

4 The Adapted Lesk Algorithm

This algorithm takes as input an example or *instance* in which a single *target word* occurs, and it will output a WordNet sense for that target word based on information about the target word and a few immediately surrounding words that can be derived from WordNet.

Our experimental data is the English lexical sample from SENSEVAL-2, where each instance of a target word consists of the sentence in which it occurs, along with two or three surrounding sentences. However, our algorithm utilizes a much smaller window of context that surrounds the target word.

We define the *context* of the target word to be a window of n WordNet word tokens to the left and another n tokens to the right, for a total of $2n$ surrounding words. We include the target word in the context as well, giving a total context size of $2n + 1$ word tokens. Repeated occurrences of a WordNet word in the window are treated separately.

If the target word is near the beginning or end of the instance, we add additional WordNet words from the other direction. This is based on the suggestion of Lesk [3] that the quantity of data available to the algorithm is one of the biggest factors to influence the quality of disambiguation. We therefore attempt to provide roughly the same amount of data for every instance of every target word.

4.1 Definitions

Let the size of window of context, $2n+1$ be designated by N. Let the WordNet words in the window of context be designated as W_i, $1 \leq i \leq N$. If the number of WordNet words in the instance is less than $2n+1$, all of the WordNet words in the instance serve as the context.

Each word W_i has one or more possible senses, each of which is represented by a unique synset having a unique sense–tag. Let the number of sense–tags of the word W_i be represented by $|W_i|$. Hereafter we use sense–tag to refer to a sense of a word.

We evaluate each possible combination of sense–tag assignments for the words in the context window. There are $\prod_{i=1}^{N}|W_i|$ such combinations, each of which we refer to as a *candidate combination*.

A *combination score* is computed for each candidate combination. The target word is assigned the sense–tag of the candidate combination that attains the maximum score. While this combination also provides sense tags for the other words in the window of context, we view these as a side effect of the algorithm and do not attempt to evaluate how accurately they are disambiguated.

4.2 Processing

This algorithm compares glosses between each pair of words in the window of context. If there are N words in the window of context then there are $N(N-1)/2$ pairs of words to be compared. There are a series of *relation pairs* that identify which synset is to provide the gloss for each word in a pair during a comparison. For example, a relation pair might specify that the gloss of a synset of one word is to be compared with the gloss of a hypernym of the other word. The glosses to be compared are those associated with the senses given in the candidate combination that is currently being scored.

In our experiments, we compare glosses associated with the synset, hypernym, hyponym, holonym, meronym, troponym, and attribute of each word in the pair. If the part of speech of a word is known, as is the case for target words, then we restrict the relations and synsets to those associated with that part of speech. If the part of speech is not known, as is the case for the other words in the context, then we use relations and synsets associated with all the possible parts of speech. Since there are 7 possible relations, there are at most 49 possible relation pairs that must be considered for a particular pair of words. However, if we know the part of speech of the word, or if the word is only used in a subset of the possible parts of speech, then the number of relation pairs considered is less. The algorithm is not dependent on any particular relation pairs and can be run with as many or as few as seems appropriate.

When comparing two glosses, we define an *overlap* between them to be the longest sequence of one or more consecutive words that occurs in both glosses. Each overlap found between two glosses contributes a score equal to the square of the number of words in the overlap.

Two glosses can have more than one overlap where each overlap covers as many words as possible. For example, the sentences *he called for an end to the atrocities* and *after bringing an end to the atrocities, he called it a day* have the following overlaps: *an end to the atrocities* and *he called.* We stipulate that an overlap not be made up entirely of *non–content* words, that is pronouns, prepositions, articles and conjunctions. Thus if we have *of the* as an overlap, we would ignore it.

Once all the gloss comparisons have been made for every pair of words in the window based on every given relation pair, we add all the individual scores of the comparisons to arrive at the combination score for this particular candidate combination of sense–tags. This process repeats until all candidate combinations have been scored.

The candidate combination with the highest score is the winner, and the target word is assigned the sense given in that combination. In the event of a tie between two candidate combinations we choose the one that has the most familiar sense for the target word, as specified by WordNet.

5 Empirical Evaluation

We have evaluated this algorithm using the test data from the English lexical sample task used in the Senseval-2 comparative evaluation of word sense disambiguation systems. The 73 target words in this data are listed below. There are a total of 4,328 test instances, divided among 29 nouns, 29 verbs, and 15 adjectives. Each word is followed by the accuracy attained by our algorithm, the number of possible WordNet senses, and the number of test instances. Note that accuracy is defined to be the number of correctly disambiguated instances divided by the number of total test instances for a word.

Nouns: art (0.500, 4, 98), authority (0.337, 7, 92), bar (0.113, 17, 151), bum (0.178, 6, 45), chair (0.522, 6, 69), channel (0.096, 10, 73), child (0.500, 4, 64), church (0.453, 4, 64), circuit (0.247, 7, 85), day (0.172, 10, 145), detention (0.625, 2, 32), dyke (0.286, 3, 28), facility (0.293, 5, 58), fatigue (0.279, 6, 43), feeling (0.275, 6, 51), grip (0.078, 11, 51), hearth (0.562, 3, 32), holiday (0.710, 3, 31), lady (0.566, 3, 53), material (0.217, 11, 69), mouth (0.400, 11, 60), nation (0.730, 4, 37), nature (0.370, 5, 46), post (0.203, 20, 79), restraint (0.200, 6, 45), sense (0.377, 6, 53), spade (0.273, 4, 33), stress (0.256, 8, 39), yew (0.607, 2, 28)

Accuracy for nouns = 0.322, 564 of 1754 correct

Verbs: begin (0.475, 11, 280), call (0.091, 41, 66), carry (0.091, 40, 66), collaborate (0.900, 2, 30), develop (0.261, 21, 69), draw (0.049, 44, 41), dress (0.220, 19, 59), drift (0.062, 17, 32), drive (0.167, 33, 42), face (0.237, 23, 93), ferret (1.000, 5, 1), find (0.029, 18, 68), keep (0.164, 25, 67), leave (0.288, 17, 66), live (0.313, 19, 67), match (0.238, 18, 42), play (0.197, 53, 66), pull (0.033, 25, 60), replace (0.289, 4, 45), see (0.159, 26, 69), serve (0.118, 16, 51), strike (0.056, 26, 54), train (0.286, 17, 63), treat (0.409, 9, 44), turn (0.060, 38, 67), use (0.658, 13, 76), wander (0.100, 5, 50), wash (0.167, 19, 12), work (0.083, 34, 60)

Accuracy for verbs = 0.249, 450 of 1806 correct

Adjectives: blind (0.782, 10, 55), colourless (0.400, 2, 35), cool (0.403, 11, 52), faithful (0.783, 5, 23), fine (0.443, 15, 70), fit (0.448, 16, 29), free (0.378, 20, 82), graceful (0.793, 2, 29), green (0.404, 15, 94), local (0.289, 5, 38), natural (0.262, 13, 103), oblique (0.345, 3, 29), simple (0.500, 9, 66), solemn (0.920, 2, 25), vital (0.632, 4, 38)

Accuracy for adjectives = 0.469, 360 of 768 correct

Thus, overall accuracy is 31.7%, where 1374 of 4328 test instances are disambiguated correctly. In SENSEVAL-2 two variations of the Lesk algorithm were provided as benchmarks. The first counts the number of words in common between the instance in which the target word occurs and its gloss, where each word count is weighted by its inverse document frequency. Each gloss is considered a separate document in this approach. The gloss with the highest number of words in common with the instance in which the target word occurs represents the sense assigned to the target word. This approach achieved 16% overall accuracy. A second approach proceeded identically, except that it added example texts that WordNet provides to the glosses. This achieved accuracy of 23%. Since our approach does not use example texts, the most indicative comparison is with the first approach. Thus, by including an extended notion of which glosses to compare a target word's gloss with, we have doubled the accuracy from 16% to 32%. The fact that the example texts provided by WordNet improved the accuracy of these benchmark approaches suggests that we should consider using this information as well.

In addition, our approach compares favorably with other systems entered in SENSEVAL-2. Of the seven unsupervised systems that did not use any of the available training examples and only processed test data, the highest ranked achieved accuracy of 40%. There were four approaches that achieved accuracy of less than 30%.

6 Analysis of Results

In preliminary experiments we ignored the part of speech of the target word, and we included overlaps that consist entirely of non–content words. While the overall accuracy of this approach was only 12%, the accuracy for the nouns was 29%, which is only slightly less than that obtained when using the part of speech information for target words. However, significant reductions in accuracy were observed for adjectives (11%) and verbs (7%).

These results confirm the notion that WordNet is a particularly rich source of information about nouns, especially when considering the hypernym and hyponym relations. When compared to verbs and adjectives, there is simply more information available. When we ignored part of speech distinctions in the target word, those that can be used in multiple parts of speech such as *dress, blind,* etc., made gloss comparisons involving all their possible parts of speech. In doing so, they were likely overwhelmed by the sheer volume of noun information, and this resulted in poor accuracy when the target word was in fact an adjective or verb.

This algorithm very rarely encounters situations where it can not make a determination as to sense–tags. A candidate combination with no overlaps receives a score of zero. If every candidate combination associated with a particular target word gets a score of zero, then the algorithm assigns every word in the window with its most familiar sense, according to WordNet. However, there were only ten test instances for which our algorithm had to resort to this default. If there are two or more candidate combinations tied at the highest score, then we report the most familiar of these senses. Such ties are also rare, occurring only 57 times out of 4,328 test instances.

7 Discussion

There are numerous issues that arise in formulating and refining this algorithm. We discuss a few of those issues here, among them how to represent context, which relations should be the basis for comparisons, how to score matches, and how to deal with possible performance issues. The current approach is a first approximation of how to use a Lesk algorithm with WordNet, so certainly there is room for considerable variation and experimentation.

7.1 Context

Our choice of small context windows is motivated by Choueka and Lusignan [1], who found that human beings make disambiguation decisions based on very short windows of context that surround a target word, usually no more than two words to the left and two words to the right. While WordNet does not provide anywhere near the same level of knowledge about words that a human being has, it encodes at least a portion of such information through its definitional glosses and semantic relations between synsets. We believe this provides sufficient information to perform word sense disambiguation across a range of words and parts of speech.

For example, consider the sentence *I put money in the bank*. A human being asked to disambiguate *bank* knows that it is much more common to put money into a financial institution rather than a slope. WordNet supports the same inference if one observes that a *bank* "channels... money into lending activities" when it is a *financial institution* and not when it is a *slope*. The fact that *money* occurs in the definition of one sense and not in the other implies a strong connection between money and that sense of bank. Given that the words in a sentence usually have a flow of related meanings, it is very likely that successive words in a sentence will be related.

By identifying overlaps between the senses of one word and those of the next in a context window, we are trying to identify any such connection between a particular sense of one word and that of the next. Such connections are not accidental but are indicative of the senses in which these words are used.

7.2 Relations

This algorithm depends very much on the relation pairs that are employed. We have only experimented with synsets that are directly related to the words being compared, however, other more indirect relations may provide useful information. One example is the coordinate or sister relation, which consists of the hyponyms of the hypernym of a synset.

As was mentioned earlier, we have not used every relation provided in WordNet. Among those left out are *cause* and *entailment* for verbs and *similar to*, *participle of* and *pertainym of* for adjectives. There is also an *antonymy* relation that applies to all parts of speech, that relates a synset to another that represents its opposite in meaning. This is particularly intriguing in that it provides a source of negative information that will allow our algorithm to identify the sense of a word based on the absence of its antonymous sense in the window of context.

A single synset may be related to multiple synsets through a single relation. For example, when a relation pair includes hyponomy, we concatenate the glosses of all the hyponym synsets and treat them as one during matching. We do not distinguish between separate synsets, as long as they are all related to the synset through the same relation.

Our scoring mechanism also does not distinguish between matches amongst different relations; all relation–pairs are treated equally. Thus an n word match between two hypernyms gets precisely the same score as an n word match between a hyponym and a hypernym. As yet we have no reason to prefer matches between certain pairs of relations over others, and so the scoring is uniform. As we begin to better understand which relation pairs lead to more meaningful matches, we will likely adjust the scoring mechanism to reward more useful matches.

7.3 Scoring

One of the novel aspects of this approach is the fact that scores are based on the length of the match. By using the square of the number of words in the match as the score, we appeal to Zipf's Law which states that the frequency of an event is inversely proportional to the rank of that event, where the ranking is based on the frequencies of all events. This implies that most events occur only once, and only a few occur with greater frequency. The occurrence of individual words in a corpus of text holds to this distribution, and it also applies to the occurrence of multi-word sequences. As word sequences grow longer, it is increasingly rare to observe them multiple times in a corpus. Thus, if we find a multi-word match between two glosses, this is a remarkable event, and merits more than a linear increase in scoring. By squaring the length of the match, we give a higher score to a single n word sequence than to the combined score of those n words, if they were to occur in shorter sequences.

Partial word matches are discarded so as to rule out spurious matches. For example between *Every dog has its day* and *Don't make hasty decisions*, there exists an overlap *has*, which is not particularly useful. We also discard overlapping sequences that consist entirely of function words since these are also of questionable value and may skew results towards longer glosses that happen to contain more function words. However, sequences of content words that also contain function words are retained. This is to preserve longer sequences, as opposed to breaking them down into smaller sequences due to the presence of function words. In future we will consider *fuzzier matching* schemes, where stemming or measures of edit distance are employed to account for near matches. Sidorov and Gelbukh [4] present such an approach in a variant of the Lesk algorithm applied to a Spanish explanatory dictionary.

In scoring a combination of candidate senses, we compare all pairs of words in the context window. Thus, if we have a five word context window, a strong relationship between the words on the extreme ends can force a certain sense for the words in the middle of the window. A possible variation suggested by Lesk [3] is to weigh the score of a pair of words by the distance between them. Thus, one might give higher scores to words that appear more closely together in the window of context.

7.4 Performance

Each WordNet word usually has multiple sense–tags. As the window of context becomes larger, the number of possible combinations of candidate sense–tags grows rapidly. There are three immediate courses of action that we can take to alleviate this problem. The first is to part of speech tag all of the words in the window of context, and thereby restrict the range of their possible sense tags to those associated with the given part of speech. The second is to focus the algorithm strictly on the target word and eliminate all comparisons between pairs of glosses that do not involve the target word. The third is to restrict the consideration of possible senses to among the most familiar in WordNet.

8 Conclusions

This paper presents an adaptation of the Lesk algorithm for word sense disambiguation. While the original algorithm relies upon finding overlaps in the glosses of neighboring words, this extends these comparisons to include the glosses of words that are related to the words in the text being disambiguated. These relationships are defined by the lexical database WordNet. We have evaluated this approach on the English Senseval-2 lexical sample data and find that it attains overall accuracy of 32%, which doubles the accuracy of a more traditional Lesk approach. The authors have made their Perl implementation of this algorithm freely available on their web sites.

Acknowledgments

This work has been partially supported by a National Science Foundation Faculty Early CAREER Development award (#0092784). We would like to thank the Senseval-2 coordinators for putting their data in the public domain. We are grateful to Jason Rennie for making his Perl WordNet interface, QueryData, available, and to the WordNet team for their invaluable contributions to this research. And of course, we are indebted to Michael Lesk whose original work is the inspiration of this particular approach.

References

1. Y. Choueka and S. Lusignan. Disambiguation by short contexts. *Computers and the Humanities*, 19:147–157, 1985.
2. C. Fellbaum, editor. *WordNet: An electronic lexical database.* MIT Press, 1998.
3. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.
4. G. Sidorov and A. Gelbukh. Word sense disambiguation in a Spanish explanatory dictionary. In *Proceedings of TALN*, pages 398–402, Tours, France, 2001.

Feature Selection Analysis for Maximum Entropy-Based WSD[⋆]

Armando Suárez and Manuel Palomar

Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante,
Alicante, Spain
{armando,mpalomar}@dlsi.ua.es

Abstract. Supervised learning on a corpus-based Word Sense Disambiguation (WSD) system uses a previously classified set of linguistic contexts. In order to perform the training of the system, it is usual to define a set of functions that inform of any linguistic feature in each example. It is usual to look for the same kind of information for each word too, at least on words of the same part-of-speech.

In this paper, a study of feature selection in a supervised learning method of WSD based on corpus, *Maximum Entropy conditional probability models*, is presented. For a few words selected from the DSO corpus, the behaviour of several types of features has been analyzed in order to identify their contribution to gains in accuracy and to determine the influence of sense frequency in that corpus. This paper shows that not all words are better disambiguated with the same combination of features. Moreover, an improved definition of features in order to increase efficiency is presented as well.

1 Introduction

Word Sense Disambiguation is an open research field in Natural Language Processing (NLP). The task consists of assigning the correct sense to nouns, verbs, adjectives and adverbs and it is a hard problem that is receiving many efforts from the research community.

Currently, two main tendencies can be found in this research area: *knowledge-based* methods and *corpus-based* methods. The first ones rely on previously acquired linguistic knowledge, and the second ones use techniques from statistics and machine learning to induce models of language usage from large samples of text [7]. These last methods can perform supervised or unsupervised learning, that is, the corpus is previously tagged with correct answers or not.

Usually, supervised learning methods represent linguistic information in the form of features. Each feature informs of the occurrence of certain attribute in a context that contains a linguistic ambiguity. That context is the text surrounding

[⋆] This paper has been partially supported by the Spanish Government (CICYT) project number TIC2000-0664-C02-02.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 146–155, 2002.

this ambiguity and relevant to the disambiguation process. The features used can be of distinct nature: word collocations, part-of-speech labels, keywords, topics and domain information, and so on.

A WSD method using supervised learning tries to classify a context containing an ambiguous word in one of its possible senses by means of a classification function. This function is obtained after a training process on a sense tagged corpus. The information source for this training is the set of results of the features evaluation on each context, that is, each context has its vector of feature values.

This paper is focused on the definition and influence on evaluation results of different types of features in a supervised learning method of WSD. It is usual to train this kind of methods with the same kinds of information for all words of the same part-of-speech, underestimating which information is more relevant to each word. One the objectives of this paper is to show that each word needs a different set of features in the training of the method.

Another important issue of this paper is the definition of the feature-functions. WSD methods based on corpus suffer the sparse nature of data and each feature is activated a very few times. A new feature definition is proposed and it is empirically demonstrated that results have a minimum degradation while efficiency is highly improved.

The supervised learning WSD method used to do such analysis is based on Maximum Entropy probability models (ME). In the following, the ME framework will briefly shown. Next, the complete set of feature definitions used in this work will be detailed in the form of function templates. Next, evaluation results using several combinations of such types of features for a few words will be shown. With these results, the contribution of each kind of feature to the disambiguation process is analyzed. Finally, some conclusions and future and in progress work will be presented.

2 The Maximum Entropy Framework

Maximum Entropy (ME) modeling is a framework for integrating information from many heterogeneous information sources for classification [3]. ME probability models were successfully applied to some NLP tasks such as part-of-speech (POS) tagging or sentence boundary detection [8].

The WSD method used in this paper is based on conditional ME probability models [9]. It has been implemented a supervised learning method consisting of building word sense classifiers using a semantically tagged corpus. A classifier obtained by means of a ME technique consists of a set of parameters or coefficients estimated by an optimization procedure. Each coefficient is associated to one feature observed in the training data. The main purpose is to obtain the probability distribution that maximizes the entropy, that is, maximum ignorance is assumed and nothing apart of training data is considered. As advantages of the ME framework, knowledge-poor features applying and accuracy can be mentioned; The ME framework allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features [8].

Let us assume a set of contexts X and a set of classes C. The function $cl : X \rightarrow C$ chooses the class c with the highest conditional probability in the context x: $cl(x) = \arg\max_c p(c|x)$. Each feature is calculated by a function that is associated to a specific class c' and it has the form of (1), where $cp(x)$ is some observable characteristic in the context[1]. The conditional probability $p(c|x)$ is defined as (2) where α_i is the parameter or weights of the feature i, K the number of features defined, and $Z(x)$ a constant to ensure that the sum of all conditional probabilities for this context is equal to 1.

$$f(x,c) = \begin{cases} 1 \text{ if } c' = c \text{ and } cp(x) = \mathit{true} \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{2}$$

Next section shows the implementation of the ME framework in this work.

3 Feature Implementation

An important issue of this implementation of the ME framework is the form of the functions that calculate each feature. These functions are defined in the training phase and depend on the data in corpus. In WSD it is normal to use the information of words that are in a certain position related to the ambiguous word (e.g. the previous word w_{-1}). A usual definition of features has the form of (3) where $info(x,i)$ informs of a property that can be found at position i in contexts x that are previously classified as sense c'. In the following, this function form will be referenced as *template-word.*

$$f_{(c',a,i)}(x,c) = \begin{cases} 1 \text{ if } c' = c \text{ and } a = info(x,i) \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

This kind of feature-function template generates a function for each possible $(class, a, i)$, where $class$ is each possible sense of the ambiguous word and a is one of the elements in the set of all attribute values (e.g., words) observed at position i of one context at least. Obviously, even restricting the set of possible classes to only those ones in corpus, and using only those $(class, a, i)$ where a occurs with $class$ in some context, the number of features is very heavy.

Rather than the previous one, the template shown in (4) is used for the automatic definition of feature-functions before training the method. Instead of one function for each possible $(class, a, i)$, this kind of functions reduce the number of features to one per each $(class, i)$. In the following, this function form will be referenced as *template-set.*

[1] The ME approach is not limited to binary funtions, but the optimization procedure used for the estimation of the parameters, the *Generalized Iterative Scaling* procedure, uses this kind of features.

$$W_{(c',i)} = \{a \mid \exists x(x \in X, a = CP(x,i), sense(x) = c')\} \tag{4}$$

$$f_{(c',i)}(x,c) = \begin{cases} 1 \text{ if } c' = c \ \text{ and } \ CP(x,i) \in W_{(c',i)} \\ 0 \text{ otherwise} \end{cases}$$

Let us assume that CP is the name of a function that returns some information about the context x. From now on, when a set of features is being described, CP must be substituted by the correct function name, depending on the linguistic attribute to be identified. In general, this function returns a word, a lemma, or a POS label at a specific position in x. In this manner, in most of feature definitions, CP will be substituted by *lemma* or *word* or POS. Let us assume that these three functions return a lemma, a word and a POS-tag, respectively, at position i in a context x.

On the other hand, these functions are based on $W_{(c',i)}$-sets built before the training itself. Training contexts are preprocessed and a set for each possible (c', i) is built. These sets will contain all lemmas, words or POS-tags observed at position i of a context classified as sense c' in corpus.

The section 5 shows that the performance of the method is not penalized by this feature definition. Due to the nature of the disambiguation task, the times that a function generated by a *template-word* is activated are very low, and feature vectors have a large number of values 0. This new template reduces drastically the number of features with a minimum degradation of evaluation results.

The following section shows, finally, which features have been used in order to perform an analysis of its influence on WSD success.

4 Description of Features

The set of features defined for ME training is described below (figure 4) and it is based on the feature selection made in [6] and [1]. Features are automatically defined using a template-function and depend on the data in the training corpus. Each template builds a set of functions that are associated to a sense, a word, a set of words, a POS-tag and/or a position related to the word to classify (in the following, *target-word*).

0-Features

The first group of features, *0*-features, corresponds to *target-word* itself. For nouns and adjectives, the word changes with number and capitalization, and for verbs there are more possibilities depending on tense and number. These shape differences can be strongly related to a particular sense of the ambiguous word.

Before the method training, each different *target-word* is used to build one feature for each class using *template-word*, where CP is substituted by *word* and i by 0 (it is assumed that *target-word* defines the position zero in context).

S-Features

The *S*-features are those previously defined in [6] and [1], and supply information about words near *target-word* in context: $w_{-1}, w_{-2}, w_{-3}, w_{+1}, w_{+2}, w_{+3}$.

- *Template-word*
  - ***0*-features**: target-word.
  - ***S*-features**: words in positions $\pm1, \pm2, \pm3$.
  - ***Q*-features**: POS-tags of words in positions $\pm1, \pm2, \pm3$.
  - ***Km*-features**: lemmas of nouns in context in any position, occurring at least $m\%$ times with a sense.
- *Template-set*
  - ***L*-features**: lemmas of words in positions $\pm1, \pm2, \pm3$.
  - ***W*-features**: words in positions $\pm1, \pm2, \pm3$.
  - ***B*-features**: lemmas of collocations in positions $(-2,-1), (-1,+1), (+1,+2)$.
  - ***C*-features**: collocations in positions $(-2,-1), (-1,+1), (+1,+2)$.
  - ***P*-features**: POS-tags of words in positions $\pm1, \pm2, \pm3$.

Fig. 1. List of types of features

The template (3) is used by means of the substitution of CP by the function name *word* and i by each corresponding position. Therefore, for each word at position i, a function is defined for each possible sense of *target-word.*

Q-Features

Template-word is used again for Q-features. Now, the POS-tags of words at positions $q_{-3}, q_{-2}, q_{-1}, q_{+1}, q_{+2}, q_{+3}$ is the information allowed. CP is substituted by the function name POS and i by each position in order to define the corresponding functions.

Km-Features

This set of features is vaguely inspired by [6] and consists of a nouns selection done by means of frequency information of nouns co-occurring with a sense. For example, in a set of 100 examples of sense four of the noun "interest", if "bank" is found 10 times or more ($m = 10\%$) then a feature for each possible sense of "interest" is defined with "bank". These functions have the form of (5).

$$W = \{w \mid \exists x \exists c (x \in X,\ c \in C, w \in x, freq(w,c) > m, sense(x) = c)\} \qquad (5)$$

$$f_{(c',w)}(x,c) = \begin{cases} 1 \text{ if } c' = c \text{ and } w \in W \\ 0 \text{ otherwise} \end{cases}$$

Functions described above are those based on *template-word.* In the following, feature-functions using *template-set* will be described. As mentioned before, preprocessing the training corpus builds the $W_{(c',i)}$-sets on which these functions are based.

L-Features and *W*-Features

L-features, correspond to lemmas of words near *target-word.* According to their position related to the target word in each sentence, lemmas to be processed are:

$l_{-1}, l_{-2}, l_{-3}, l_{+1}, l_{+2}, l_{+3}$. *Template-set* is used substituting CP by the function name *lemma* and i by each selected position. The W-features are built by the same template, but using *word* (the shape of words).

In both cases, current implementation only permits content-words (nouns, verbs, adjectives and adverbs) to be able to activate these features.

B-Features and *C*-Features

The sets of features B and C are similar to previous L and W but refer to collocations of two lemmas or two words at positions $(-2, -1)$, $(-1, +1)$ and $(+1, +2)$. In both cases, at least one of the words in the collocation is a content-word. *Template-set* is used again by both kind of features with minor modifications.

P-Features

Finally, P-features are defined using POS-tags of words at positions related to *target-word*: $p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, p_{+3}$. POS is the name of the function used in *template-set*. All words, content and function words, are considered.

5 Feature Analysis

At this point, a study of the relevance of each feature is shown. Some polysemous nouns and verbs have been selected and evaluated using the DSO sense tagged English corpus [6]. This corpus is structured in files containing tagged examples of a word. Tags correspond to the a sense in WordNet 1.5 [5]. Examples were extracted from articles of the Brown Corpus and the Wall Street Journal.

Table 1 shows the best results obtained using a 10-fold cross-validation evaluation method. Several feature combinations have been established in order to find the best set of them for each selected word. For each word, number of distinct senses in the corpus, best feature selection, accuracy[2] and number of functions are shown.

In general, current ME implementation cannot properly classify a context when this last one has not enough information (e.g. there are no content words near *target-word*). These contexts obtain the same maximum probability value for several senses and cannot return a unique one. Nevertheless, all feature selections in table 1 obtain enough information from almost all examples in the evaluation folds.

Another important issue is the number of senses of each word and the distribution of them in corpus. In order to perform the ten tests on each word, some preprocessing on the corpus has been made. For each word file in DSO, all senses have been uniformly distributed in ten folds, that is, each file contains 1/10 examples of each sense, except the 10th fold that can contain the remaining examples. Those senses that have less than 10 examples in the original corpus file have been rejected and not processed.

[2] *accuracy* = number of correctly classified contexts divided by number of contexts.

Table 1. 10-fold cross-validation best results on DSO files

	senses	features	accuracy	functions
age,N	3	SQ	0.743	1482.5
art,N	4	0LWBCP	0.641	113.0
car,N	2	WSB	0.969	2758.9
child,N	2	LWBCQ	0.945	264.1
church,N	3	0LWSBCQ	0.654	1340.8
cost,N	2	SCQ	0.897	3030.7
fall,V	6	0LWBCK3	0.859	342.9
head,N	7	SQ	0.814	2317.8
interest,N	6	0LWSBCQ	0.683	4173.1
know,V	6	0LWSBCQ	0.488	3826.8
line,N	22	0LWBCK3	0.569	1942.0
set,V	11	0LWBCPK5	0.580	683.2
speak,V	5	SQ	0.762	1658.2
take,V	19	LWSBC	0.408	3119.9
work,N	6	LWBCPK5	0.518	207.7

Table 2. First best result of non-SQ-features selection selection

	features	functions	loss	diff-f
age,N	0LB	37	-2.08%	-1446
art,N	0LWBCP	113	0.00%	0
car,N	LWB	31	-0.29%	-2728
child,N	LWBCP	49	-12.29%	-215
church,N	LWBCPK10	83	-1.05%	-1258
cost,N	LWBCP	49	-0.49%	-2982
fall,V	0LWBCK3	343	0.00%	0
head,N	LWBCPK3	493	-2.76%	-1825
interest,N	LWBCK3	261	-2.18%	-3912
know,V	0WB	102	-1.94%	-3724
line,N	0LWBCK3	1942	0.00%	0
set,V	0LWBCPK5	683	0.00%	0
speak,V	0LWBCPK10	167	-2.67%	-1491
take,V	0LWBCPK5	1224	-0.43%	-1896
work,N	LWBCPK5	208	0.00%	0

Table 2 shows that evaluation results of the method are not penalized by *template-set* definitions. For each word, the first best feature selection without S or Q-features and a comparison with best results in table 1 are shown. Column *features* shows the feature selection; *functions* the number of functions generated by those features; *loss* the loss in accuracy related to results in table 1; and *diff-f* the reduction of the number of functions defined to do the learning.

Table 3 shows the results obtained for several selected combinations of features, in order to establish the gain in accuracy when new features are incorpo-

Table 3. Feature selection comparison

	LB	LWBC	LWBCP	0LB	LWBCQ	SQ	0LWSBCQ
age,N	0.704	-0.021	-0.007	0.008	0.014	0.039	0.037
art,N	0.529	0.013	0.071	0.107	0.066	0.036	0.088
car,N	0.963	0.002	-0.003	-0.002	-0.001	0.001	-0.001
child,N	0.795	-0.005	0.028	0.018	0.151	0.144	0.143
church,N	0.607	0.014	0.027	0.003	0.009	0.020	0.047
cost,N	0.869	-0.003	0.022	0.020	0.023	0.026	0.024
fall,V	0.836	0.005	-0.006	-0.006	-0.005	0.007	0.010
head,N	0.669	-0.012	0.092	0.011	0.128	0.145	0.143
interest,N	0.635	0.001	0.010	0.013	0.019	0.035	0.048
know,V	0.417	0.001	0.014	0.046	0.034	0.063	0.072
line,N	0.500	0.006	0.032	0.024	0.028	0.001	0.027
set,V	0.543	-0.003	0.015	0.027	0.021	0.006	0.033
speak,V	0.697	-0.012	-0.004	0.027	0.051	0.065	0.058
take,V	0.359	-0.007	-0.009	-0.017	-0.022	-0.004	-0.001
work,N	0.494	0.001	0.023	0.012	0.013	0.010	0.018
Averages	**gain**	**-0.13%**	**2.03%**	**1.94%**	**3.52%**	**3.96%**	**4.97%**
	nouns	-0.04%	2.95%	2.14%	4.50%	4.57%	5.74%
	verbs	-0.31%	0.19%	1.53%	1.56%	2.73%	3.43%

rated to the training. The column *LB* is the base-value from which the rest of columns shows the gain or loss in accuracy.

Except for the noun *child* where *template-word* functions work above a 12% better than the first *template-set* combination of features, the accuracy loss rounds 1.75% (0.99% without *child* loss) and several words have no loss because the best result does not use *template-word* functions. At the same time that computing time is drastically reduced, it is expected that this reduction of functions may be used in incorporating other linguistic features.

0-features are very useful for verbs due to tense and number variations of *target-word*. Nouns like *art* and *age* take benefit from this kind of features because they have senses strongly related to plural and capitalization (e.g. *Arts*).

Although such data is not shown in this paper, our experiments indicate that *L* and *W*-features contribute to disambiguate all senses of nouns and verbs with a high *precision* but a low *recall*. These features are useful for less frequent senses but the success rate is higher for more frequent ones, specially when they have much more examples than the first ones.

B and *C*-features have a minor impact on results and their improvement is almost negligible. Nevertheless, some senses are used in common expressions that are statistically significant. High *precision* and low *recall* is obtained again.

In general, *LB* combination of features is quite informative when used in combination with MFS assignment strategy: less frequent senses are better identified and the majority of not disambiguated contexts are MFS contexts. Therefore, the addition of more types of features improves *recall* and decreases *precision* until both have the same value. Obviously, this is true on corpora with a great difference between more and less frequent senses.

Q and P-features tend to favor more frequent senses and work to the disadvantage of the less frequent ones. However, in most cases, the improvement on more frequent senses success is reflected in accuracy too. As *0*-features, with this kind of features almost 100% of contexts are fully disambiguated by the ME method, and *precision* and *recall* make equal.

Km-features are the least used if table 1 is examined but their information of senses with low number of training examples is important. These features have the number of functions as a disadvantage, and can introduce "noise" into the model.

6 Discussion and Conclusions

A study of feature selection in a corpus-based WSD method, Maximum Entropy conditional probability models, has been presented. For a few words selected from the DSO corpus, several combinations of features have been analyzed in order to identify the best of them.

This study shows that not all words need the same information, probably due to the training corpus itself. A WSD system based on a supervised training must identify which features work better for each word. Moreover, certain sources of information make easier to classify more frequent senses but decreases the success on the others.

A new definition of the functions that calculate these features has been presented too. The number of features is not determinant to results: it has been shown that a redefinition of feature functions grouping the attributes, work as well as "classical" functions, for the ME method at least. This is an important issue in order to increase WSD efficiency.

Another important issue is the number of examples itself. The dependence of supervised learning methods on corpora is a real problem if we want to build a robust system [2]. A possible approach to this problem is the extraction of examples from internet using the available search engines in order to enrich corpora and to approximate the sense frequency to a realistic one. The topic and genre variations in the DSO corpus have not been examined either [4].

Finally, it seems assumed that more information sources are needed in order to improve current WSD disambiguation systems. Deeper sentence analysis data, domain and topic information and, probably, a cooperation between several methods, even knowledge-based methods, are promising sources of features.

References

1. Gerard Escudero, Lluis Màrquez, and German Rigau. Boosting applied to word sense disambiguation. In *Proceedings of the 12th Conference on Machine Learning ECML2000*, Barcelona, Spain, 2000.
2. Gerard Escudero, Lluis Màrquez, and German Rigau. On the portability and tuning of supervised word sense disambiguation systems. In Schütze and Su [10].
3. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, 1999.

4. David Martínez and Eneko Agirre. One sense per collocation and genre/topic variations. In Schütze and Su [10].
5. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Five Papers on WordNet. *Special Issue of the International journal of lexicography*, 3(4), 1993.
6. Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word senses: An exemplar-based approach. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, San Francisco, 1996. Morgan Kaufmann Publishers.
7. Ted Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In ACL, editor, *Proceedings of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, PA, USA, 2001.
8. Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.
9. Maximiliano Saiz-Noeda, Armando Suárez, and Manuel Palomar. Semantic pattern learning through maximum entropy-based wsd technique. In *Proceedings of CoNLL-2001*, pages 23–29. Toulouse, France, 2001.
10. Hinrich Schütze and Keh-Yih Su, editors. *Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora 2000*, Hong Kong, China, 2000.

Combining Supervised-Unsupervised Methods for Word Sense Disambiguation*

Andrés Montoyo, Armando Suárez, and Manuel Palomar

Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante,
Alicante, Spain
{montoyo,armando,mpalomar}@dlsi.ua.es

Abstract. This paper presents a method to combine two unsupervised methods (Specification Marks, Conceptual Density) and one supervised (Maximum Entropy) for the automatic resolution of lexical ambiguity of nouns in English texts. The main objective is to improved the accuracy of knowledge-based methods with statistical information supplied by the corpus-based method. We explore a way of combining the classification results of the three methods: "voting" is the way we have chosen to combine the three methods in one unique decision.

These three methods have been applied both individually as in a combined way to disambiguate a set of polysemous words. Our results show that a combination of different knowledge-based methods and the addition of statistical information from a corpus-based method might eventually lead to improve accuracy of first ones.

1 Introduction

In this paper we concentrate on the resolution of the lexical ambiguity that arises when a given word has several different meanings. This specific task is commonly referred to as Word Sense Disambiguation (WSD). In general terms, WSD involves assigning a definition to a given word, in either a text or a discourse, that endows it with a meaning that distinguishes it from all of the other possible meanings that the word might have in other contexts.

Currently, two main tendencies can be found in this research area: *knowledge-based* methods and *corpus-based* methods.

The first group of methods rely on previously acquired linguistic knowledge, and work disambiguating of words by matching the context in which they appear with information from an external knowledge source. To accomplish this task, the two knowledge-based methods (Specification Marks Method [7,9] and Conceptual Density [1,2]) used in this paper, chose WordNet as it combines the features of both dictionaries and thesauruses, and also includes other links among words by means of several semantic relations, (Hyponymy, hypernymy,

* This paper has been partially supported by the Spanish Government (CICYT) project number TIC2000-0664-C02-02.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 156–164, 2002.

meronymy, etc). In other words, WordNet provides definitions for the different senses that a given word might have (as a dictionary does) and defines groups of synonymous words by means of "Synsets", which represent distinct lexical concepts, and organises them into a conceptual hierarchy (as a thesaurus does).

The second one use techniques from statistics and machine learning to induce models of language usage from large samples of text [11]. These last methods can perform supervised or unsupervised learning, that is, the corpus is previously tagged with correct answers or not.

Usually, supervised learning methods represents linguistic information in the form of features. Each feature informs of the occurrence of certain attribute in a context that contains a linguistic ambiguity. That context is the text surrounding this ambiguitiy and relevant to the disambiguation process. The features used can be of distinct nature: word collocations, part-of-speech labels, keywords, topics and domain information, etc.

A WSD method using supervised learning tries to classify a context containing an ambiguous word or compound word in one of its possible senses by means of a classification function. This function is obtained after a training process on a sense tagged corpus. The information source for this training is the set of results of the features evaluation on each context, that is, each context has its vector of feature values. The supervised learning WSD method (Maximum Entropy) used in this paper to do such analysis is based on Maximum Entropy probability models (ME) [13].

This paper is organized as follows. After this short introduction, section 2 shows the methods we have applied. Section 3 describes the test sets and shows the results. With this results, the contribution of each method to the disambiguation process is analyzed. Finally, some conclusions and future and in progress work will be presented.

2 Methods WSD for Combining

2.1 Specification Marks Framework

The method we present here [8,7] consists basically of the automatic sense-disambiguating of nouns that appear within the context of a sentence and whose different possible senses are quite related. Its context is the group of words that co-occur with it in the sentence and their relationship to the noun to be disambiguated. The disambiguation is resolved with the use of the WordNet lexical knowledge base.

The intuition underlying this approach is that the more similar two words are, the more informative the most specific concept that subsumes them both will be. In other words, their lowest upper bound in the taxonomy. (A "concept" here, corresponds to a Specification Mark (SM)). In other words, the more information two concepts share in common, the more similar they obviously are, and the information commonly shared by two concepts is indicated by the concept that subsumes them in the taxonomy.

The input for the WSD module will be the group of words $W = \{W_1, ..., W_n\}$. Each word wi is sought in WordNet, each one has an associated set $S_i = \{S_{i1}, ..., S_{in}\}$ of possible senses. Furthermore, each sense has a set of concepts in the IS-A taxonomy (hypernymy/Hyponymy relations). First, the concept that is common to all the senses of all the words that form the context is sought. We call this concept the Initial Specification Mark (ISM), and if it does not immediately resolve the ambiguity of the word, we descend from one level to another through WordNets hierarchy, assigning new Specification Marks. The number of concepts that contain the subhierarchy will then be counted for each Specification Mark. The sense that corresponds to the Specification Mark with highest number of words will then be chosen as the sense disambiguation of the noun in question, within its given context.

At this point, we should like to point out that after having evaluated the method, we subsequently discovered that it could be improved with a set of heuristics, providing even better results in disambiguation. The set of heuristics are Heuristic of Hypernym, Heuristic of Definition, Heuristic of Common Specification Mark, Heuristic of Gloss Hypernym, Heuristic of Hyponym and Heuristic of Gloss Hyponym. Detailed explanation of the method and heuristics can be found in [9], while its application to NLP tasks are addressed in [6,10].

2.2 Conceptual Density Framework

Conceptual distance tries to provide a basis for determining closeness in meaning among words, taking as reference a structured hierarchical net. The measure of conceptual distance among concepts we are looking for should be sensitive to:

- the length of the shortest path that connects the concepts involved.
- the depth in the hierarchy: concepts in a deeper part of the hierarchy should be ranked closer.
- the density of concepts in the hierarchy: concepts in a dense part of the hierarchy are relatively closer than those in a more sparse region.
- the measure should be independent of the number of concepts we are measuring.

We are working with the Agirre-Rigau Conceptual Density formula [2] shown in the formula 1, which compares areas of subhierarchies.

$$CD(c, m) = \frac{\sum_{i=0}^{m-1} nhyp^{i^{0.20}}}{descendants_c} \tag{1}$$

The numerator expresses the expected area for a subhierarchy containing m senses of the words to be disambiguated, while the divisor is the actual area, and is given by the formula 2:

$$descendants_c = \sum_{i=0}^{h-1} nhyp^i \tag{2}$$

2.3 Maximum Entropy Framework

Maximum Entropy(ME) modeling is a framework for integrating information from many heterogeneous information sources for classification [4]. ME probability models were successfully applied to some NLP tasks such as part-of-speech(POS) tagging or sentence boundary detection [12].

The WSD method used in this paper is based on conditional ME probability models [13]. It implements a supervised learning method consisting of building word sense classifiers through training on a semantically tagged corpus. A classifier obtained by means of a ME technique consists of a set of parameters or coefficients estimated by means of an optimization procedure. Each coefficient is associated to one feature observed in the training data. The main purpose is to obtain the probability distribution that maximizes the entropy, that is, maximum ignorance is assumed and nothing apart of training data is considered. As advantages of the ME framework, knowledge-poor features applying and accuracy can be mentioned; The ME framework allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features [12].

Let us assume a set of contexts X and a set of classes C. The function $cl : X \rightarrow C$ chooses the class c with the highest conditional probability in the context x: $cl(x) = \arg\max_c p(c|x)$. Each feature is calculated by a function that is associated to a specific class c' and it has the form (3), where $cp(x)$ is some observable characteristic in the context[1]. The conditional probability $p(c|x)$ is defined as (4) where α_i is the parameter or weights of the feature i, K the number of features defined, and $Z(x)$ a constant to ensure that the sum of all conditional probabilities for this context is equal to 1.

$$f(x,c) = \begin{cases} 1 \text{ if } c' = c \text{ and } cp(x) = true \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{4}$$

3 Experiments and Results

It is to prove the effectiveness of the three applied methods in an individual way and in a combined way.

The main objective of these experiments is to check the effectiveness of the three methods, applied in an individual or combined way, on oneself group of examples. The individual evaluation to each method has been conducted on the SemCor collection [5], a set of 171 documents where all content words are annotated with the most appropiate WordNet sense. However, the evaluation in a combined way has been carried out on 18 documents of SemCor. In order to evaluate each previously described method and their combination, we selected

[1] The ME approach is not limited to binary funtions, but the optimization procedure used for the estimation of the parameters, the *Generalized Iterative Scaling* procedure, needs this kind of features.

Table 1. Results of Specification Marks Method in SemCor

nombre	#	P	R	A
account	21	**0,048**	0,048	1,000
age	86	**0,523**	0,523	1,000
art	64	**0,333**	0,328	0,984
car	65	**0,734**	0,723	0,985
child	180	**0,622**	0,594	0,956
church	107	**0,539**	0,514	0,953
cost	76	**0,289**	0,289	1,000
duty	23	**0,348**	0,348	1,000
head	168	**0,204**	0,190	0,935
interest	126	**0,444**	0,444	1,000
line	118	**0,209**	0,203	0,975
member	68	**0,515**	0,515	1,000
people	244	**0,531**	0,520	0,980
term	45	**0,156**	0,156	1,000
test	34	**0,088**	0,088	1,000
work	190	**0,255**	0,253	0,989
TOTAL	1615	**0,404**	0,395	0,978

a set of nouns at random: account, age, art, car, child, church, cost, duty, head, interest, line, member, people, term, test, and work.

3.1 Experiments on Specification Marks

In this experiment, all the sentences were selected when some of the previously selected nouns appeared in the whole corpus Semcor. For each one of these sentences the nouns were obtained, forming the context of the word to be disambiguated. This context is introduced to the method of WSD, and it returns the sense corresponding of WordNet automatically for each one of the nouns. An important advantage of the method we present here consists basically of the automatic sense-disambiguating of nouns that appear within the context of a sentence. Therefore, it does not require any sort of training process, no hand-coding of lexical entries, nor the hand-tagging of texts. However, an inconvenience found in the experiments carried out with the Semcor is that the method relies on the semantics relations (Hypernymy/Hyponymy) and the hierarchical organization of WordNet used for disambiguate the sense of the words. For this reason, when the method of ME is applied on the selected nouns, there are words that have a percentage of desambiguacin so low. As it is shown in the table 1, i.e., the word "test" obtains a low percentage of disambiguation, because the other nouns of the context are not related semantically by WordNet.

3.2 Experiments on Conceptual Density

In this experiment, all the sentences were selected when some of the previously selected nouns appeared in the whole corpus Semcor. For each one of these sen-

Table 2. Results of Conceptual Density in SemCor

nombre	P	R	A
account	**0,000**	0,000	1,000
age	**0,333**	0,333	1,000
art	**0,121**	0,088	0,733
car	**1,000**	1,000	1,000
child	**0,352**	0,352	1,000
church	**0,500**	0,464	0,928
cost	**1,000**	1,000	1,000
duty	**0,500**	0,500	1,000
head	**0,000**	0,000	1,000
interest	**0,277**	0,263	0,947
line	**0,000**	0,000	1,000
member	**0,166**	0,166	1,000
people	**0,454**	0,396	0,873
term	**0,250**	0,250	1,000
test	**0,333**	0,333	1,000
work	**1,000**	0,500	0,500
TOTAL	**0,393**	0,353	0,936

tences the nouns were obtained, forming the context of the word to disambiguate. This context is introduced to the Conceptual Density Method, and it computes the Conceptual Density of each concept in WordNet according to the senses it contains in its subhierarchy. It selects the concept with highest Conceptual Density and selects the senses below it as the correct senses for the respective words. Besides completely disambiguating a word or failing to do so, in some cases the disambiguation algorithm returns several possible senses for a word. In this experiment we considered these partial outcomes as failure to disambiguate. In the table 2 is shown the results of each words.

3.3 Experiments on Maximum Entropy

Some evaluation results over a few terms of the aforementioned corpus are presented in Table 3. The system was trained with features that inform of content words in the sentence context (w_{-1}, w_{-2}, w_{-3}, w_{+1}, w_{+2}, w_{+3}), collocations ((w_{-2}, w_{-1}), (w_{-1}, w_{+1}), (w_{+1}, w_{+2}), (w_{-3},w_{-2},w_{-1}), (w_{-2}, w_{-1},w_{+1}), (w_{-1},w_{+1},w_{+2}), (w_{+1}, w_{+2},w_{+3})), and POS tags (p_{-1}, p_{-2}, p_{-3}, p_{+1}, p_{+2}, p_{+3}).

For each word, the training set is divided in 10 folds, 9 for training and 1 for evaluation; ten tests were accomplished using a different fold for evaluation in each one (10-fold cross-validation). The accuracy results are the average accuracy on the ten tests for a word.

Some low results can be explained by the corpus itself. There has not been made any selection of articles and fiction and non-fiction ones had been processed. Moreover, the number of examples of the selected words is very low too.

Table 3. Results of Maximum Entropy Method in SemCor

noun	#	P	R	A
account	2,7	**0,285**	0,263	0,872
age	10,3	**0,313**	0,143	0,438
art	7,3	**0,596**	0,575	0,966
car	6,9	**0,959**	0,959	1,000
child	19,1	**0,957**	0,169	0,189
church	12,7	**0,558**	0,543	0,967
cost	8,4	**0,883**	0,851	0,962
duty	2,5	**0,778**	0,685	0,870
head	16,6	**0,600**	0,582	0,961
interest	13,7	**0,485**	0,454	0,932
line	12,2	**0,070**	0,067	0,946
member	7,3	**0,874**	0,874	1,000
people	27,1	**0,626**	0,359	0,530
term	5,2	**0,445**	0,430	0,951
test	3,6	**0,258**	0,252	0,938
work	20,3	**0,405**	0,392	0,962
TOTAL		**0,586**	0,473	0,805

Table 4. Results comparison

method	precision	recall	attempted
SM	0.361	0.330	0.914
CD	0.358	0.327	0.891
ME	0.638	0.614	0.963
Voting	0.514	0.345	0.670
QVoting	0.517	0.517	1.000

3.4 Experiments on Voting

Two experiments had been done: *voting* and *"quality" voting*. The first one consists on considering only those contexts where at least two methods classify it as the same sense. The second one consists on assigning a "quality" vote to ME method, that is, if none of the method agrees with other, then the response of ME is the sense in which the context is classified.

In order to obtain the results shown in table 4, 18 articles of Semcor had been selected. All methods work on this set classifying the selected words previously mentioned. Context by context, classification results of every context are compared and they take a vote on each context to decide its sense.

4 Discussion

The main objective of this work is to enforce the knowledge methods and raise their accuracy but maintaining their virtues: no corpus dependence. In order to get this, two strategies had been defined: adding more knowledge-based methods and adding statistical information too.

Voting is the kind of cooperation chosen and, for those contexts which doesn't reach the enough number of votes, statistical information of a corpus-based method is supplied to resolve the ambiguity, finally.

ME, the corpus-based method, obtains better results than the knowledge-based methods, SM and CD, when applied on the evaluation set of articles, but, we have no security on what happens when the domain changes [3].

Due to this, we consider a good result the gain in precision obtained when voting is applied. The low recall of pure voting is resolve when the ME method uses its quality vote. It is usual, in circumstances like that described here, to assign the most frequent sense (in the corpus or sense one in WordNet) but this is a statistical information too. Therefore, more elaborated statistical information has been preferred; moreover, ME also applies MFS rule when the context has no enough information.

These results are indicative of a promising approach: the combination of several WSD methods in order to improve accuracy. More complex cooperation formulas can be explored too.

5 Conclusions

A study of cooperation between different WSD methods has been shown. Two knowledge-based methods, Specification Marks and Conceptual Density, and a corpus-based method, Maximum Entropy probability models, had been used in a voting strategy of sense classification.

Two voting methods had been performed. The first one only considers those context in which at least two methods agree in the classification sense. The second one, for those contexts in which there is not a minimum agreement, the ME method decides which is the sense of them.

The analysis of the results presented in this paper shows that the knowledge-based methods can obtain a considerably gain in accuracy when used jointly and combining statistical information of a corpus-based method. Approximately a 15% of precision gain is achieved in both voting methods and the number classified contexts rise to 100% when corpus-based method uses its quality vote.

As future and in progress work, more WSD methods are being studied and more complex cooperation strategies are being developed.

References

1. Eneko Agirre and German Rigau. A proposal for Word Sense Disambiguation using Conceptual Distance. In *Proceedings of the International Conference "Recent Advances in Natural Language Processing" (RANLP 95)*, 1995.
2. Eneko Agirre and German Rigau. Word Sense Disambiguation using Conceptual Density. In *Proceedings of the 16th International Conference on Computational Linguistic (COLING 96*, Copenhagen, Denmark, 1996.

3. Gerard Escudero, Lluis Màrquez, and German Rigau. On the portability and tuning of supervised word sense disambiguation systems. In Hinrich Schütze and Keh-Yih Su, editors, *Proceedings of the Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong, China, 2000.
4. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
5. G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.
6. Andrés Montoyo, Manuel Palomar, and German Rigau. Wordnet enrichment with classification systems. In *Proceedings of WordNet and Other Lexical Resources: Applications, Extensions and Customisations Workshop. The Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, pages 101–106. Carnegie Mellon University. Pittsburgh, PA, USA, 2001.
7. Andrés Montoyo and Manuel Palomar. Word Sense Disambiguation with Specification Marks in Unrestricted Texts. In *Proceedings of 11th International Workshop on Database and Expert Systems Applications (DEXA 2000). 11th International Workshop on Database and Expert Systems Applications*, pages 103–107, Greenwich, London, UK, September 2000. IEEE Computer Society.
8. Andrés Montoyo and Manuel Palomar. WSD Algorithm Applied to a NLP System . In Mokrane Bouzeghoub, Zoubida Kedad, and Elisabeth M tais, editors, *Proceedings of 5th International conference on Applications of Natural Language to Information Systems (NLDB-2000). Natural Language Processing and Information Systems*, Lecture Notes in Computer Science, pages 54–65, Versailles, France, June 2000. Springer-Verlag.
9. Andrés Montoyo and Manuel Palomar. Specification Marks for Word Sense Disambiguation: New Development. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 182–191, Mexico City, February 2001. Springer-Verlag.
10. M. Palomar, M. Saiz-Noeda, R. Muñoz, A. Suárez, P. Martínez-Barco, and A. Montoyo. PHORA: A NLP aystem for Spanish. In A. Gelbukh, editor, *Proceedings of 2nd International conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001). Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 126–139, Mexico City, February 2001. Springer-Verlag.
11. Ted Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In ACL, editor, *Proceedings of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, PA, USA, 2001.
12. Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.
13. Maximiliano Saiz-Noeda, Armando Suárez, and Manuel Palomar. Semantic pattern learning through maximum entropy-based wsd technique. In *Proceedings of CoNLL-2001*, pages 23–29. Toulouse, France, 2001.

A Proposal for WSD Using Semantic Similarity

Susana Soler and Andrés Montoyo

Research Group of Language Processing and Information Systems,
Department of Software and Computing Systems,
University of Alicante, Alicante, Spain
{susana,montoyo}@dlsi.ua.es

Abstract. The aim of this paper is to describe a new method for the automatic resolution of lexical ambiguity of verbs in English texts, based on the idea of semantic similarity between nouns using WordNet.

1 An Outline of Our Approach

The method of WSD proposed in this paper is based on knowledge and consists basically of sense-disambiguating of the verb that appear in an English sentence.

A simple sentence or question can usually be briefly described by an action and an object [1]. For example the main idea from the sentence "*He eats bananas*" can be described by the action-object pair *"eat-banana"*. Our method determine which senses of these two words are more similar between themselves.

For this task we use the concept of semantic similarity [2] between nouns based on WordNet [3] hierarchy. In WordNet, the gloss of a verb synset provides a noun-context for that verb, i.e. the possible nouns occurring in the context of that particular verb [1]. The glosses are used here in the same way a corpus is used.

Our method takes into consideration the verb-noun pair extracted from the sentence. This verb-noun pair is the input for the algorithm. The output will be the sense tagged verb-noun pair, so we assign the sense of the verb. The algorithm is described as follows:

Step 1. Determine all the possible senses for the verb and the noun by using WordNet. Let us denote them by $<v_1, v_2, \ldots, v_k>$ and $<n_1, n_2, \ldots, n_m>$

Step 2. For each sense of verb v_h and all senses of noun $<n_1, n_2, \ldots, n_m>$:

2.1. Extract all the glosses from the sub-hierarchy including v_h. The sub-hierarchy including a verb v_h is determined as follows: consider the hypernym h_h of the verb v_h and consider the hierarchy having h_h as top [1].

2.2. Determine the nouns from these glosses. These constitute the noun-context of the verb. Determine all the possible senses for all these nouns. Let us denote them by $<x_1, x_2, \ldots, x_n>$.

2.3. Then we obtain the similarity matrix (Sm) using the semantic similarity, where each element is defined as follows:

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 165–167, 2002.

$$Sm(i, j) = sim (x_i, n_j)$$

For determining the semantic similarity ($sim(x_i, n_j)$) between each sense of the nouns extracted from the gloss of verb and each sense of the input noun, we use the formula followed:

$$sim (x_i, n_j) = 1 - sd (x_i, n_j)^2$$

$$\underline{sd} (x_i, n_j) = \frac{1}{2} (\frac{D1 - D}{D1} + \frac{D2 - D}{D2}),$$

where $sim (x_i, n_j)$ is the semantic similarity between two concepts defined by their WordNet synsets x_i and $n_{j;}$ $sd (x_i, n_j)$ is the semantic distance for nouns. D1 is the depth of synset x_i, D2 is the depth of synset n_j, and D is the depth of their nearest common ancestor in the WordNet hierarchy.

2.4. Determine the total similarity between the sense *h* of verb (v_h) and all the senses of input noun $<n_1, n_2, \ldots, n_m>$. For each n_j:

$$Ts(h, j) = \sum_{i=1}^{n} sim (x_i, n_j),$$

where n is the number of nouns extracted from the gloss of the sense *h* of the verb.

Step 3. To resume all similarity matrixes (Sm) obtained in step2 for each sense of verb, we make now the total similarity matrix (Tsm) composed by total similarity (Ts) for each sense of verb and each sense of noun. Each element of this matrix is defined as follows:

$$Tsm (i, j) = Ts (i, j).$$

Step 4. The most similar sense combination scores the highest value in the total similarity matrix (Tsm). So the output of the algorithm is the pair verb-noun (v_i-n_j) that contains this value in the matrix. Therefore the sense of the verb is chosen and given as the solution.

Consider as an example of a verb-noun pair the phrase *rewrite-article* extracted from the sentence ”*She rewrites the article once again*”. The verb *rewrite* has two senses and the noun *article* has four senses in WordNet version 1.5.

From the sense1 of verb *rewrite* we extract the nouns from its gloss. Then we have <student, thesis, week>. We obtain the semantic similarity matrix (Sm1).

rewrite1	article1	article2	article3	Article4
student1	0.31	0.37	0	0
student2	0.45	0.40	0	0
thesis1	0.67	0	0.70	0.40
thesis2	0.72	0	0.94	0.44
week1	0.29	0	0.30	0.30
week2	0.29	0	0.30	0.30
week2	0.26	0	0.27	0.27
Ts1	**2.99**	**0.77**	**2.51**	**1.71**

From the sense2 of verb *rewrite* we extract the nouns from its gloss: <purpose, play, schools, work, poem, novels>. We would obtain the following total similarity (Ts).

rewrite2	article1	article2	article3	article4
Ts2	**2.84**	**0.83**	**2.45**	**1.46**

We obtain the total similarity matrix (Tsm):

Tsm	article1	article2	article3	article4
Rewrite1	**2.99**	0.77	2.51	1.71
Rewrite2	2.84	0.83	2.45	1.46

The most similar sense combination is the sense one of the noun *article* and the sense one of the verb *rewrite*. So the output of the algorithm is the pair verb-noun: *rewrite1-article1* that contains the highest value in the matrix. The sense one of the verb *rewrite* is chosen as the solution.

2 Conclusion and Further Work

In this paper, we have presented a method for WSD that is based on semantic similarity between nouns using WordNet. Although this method has been presented as stand-alone, it is our belief that our method could be combined with other methods or could be a new heuristic of another method. In further work we intend to modify the method by adding more lexical categories for disambiguating adjectives and adverbs using the gloss of a noun synset. Finally, we pretend to test this method on sentences taken from Semcor.

References

1. Mihalcea R. and Moldovan D. (1999) *A Method for word sense disambiguation of unrestricted text*. Proc. 37th Annual Meeting of the ACL 152-158, Maryland, Usa.
2. Stetina J., Kurohashi S. and Nagao M. (1998) *General word sense disambiguation method based on full sentencial context*. In Usage of WordNet in Natural Language Processing. COLING-ACL Workshop, Montreal, Canada.
3. Miller G.A. (1990) *WordNet: An on-line lexical database*. International Journal of Lexicography, 3(4): 235-312.

A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method

Ruslan Mitkov, Richard Evans, and Constantin Orasan

School of Humanities, Languages and Social Sciences,
University of Wolverhampton,
Stafford Street,
Wolverhampton,
WV1 1SB.
UK
{r.mitkov,r.j.evans,c.orasan}@wlv.ac.uk
http://www.wlv.ac.uk/sles/compling/

Abstract. This paper describes a new, advanced and completely revamped version of Mitkov's knowledge-poor approach to pronoun resolution [21]. In contrast to most anaphora resolution approaches, the new system, referred to as MARS, operates in fully automatic mode. It benefits from purpose-built programs for identifying occurrences of non-nominal anaphora (including pleonastic pronouns) and for recognition of animacy, and employs genetic algorithms to achieve optimal performance. The paper features extensive evaluation and discusses important evaluation issues in anaphora resolution.

1 The Original Approach

Mitkov's approach to anaphora resolution [21] avoids complex syntactic, semantic and discourse analysis relying on a list of preferences known as antecedent indicators. The approach operates as follows: it works on texts first processed by a part-of-speech tagger and a noun phrase (NP) extractor, locates NPs which precede the anaphor within a distance of 2 sentences, checks them for gender and number agreement with the anaphor and then applies indicators to the remaining candidates that assign positive or negative scores to them (-1, 0, 1 or 2). The NP with the highest composite score is proposed as antecedent[1].

The antecedent indicators[2] can act either in a boosting or impeding capacity. The boosting indicators apply a positive score to an NP, reflecting a positive likelihood that it is the antecedent of the current pronoun. In contrast, the impeding ones apply a negative score to an NP, reflecting a lack of confidence that

[1] The approach only handles pronominal anaphors whose antecedents are NPs.

[2] The original indicators are named *First NPs* (FNP), *Indefinite NPs* (INDEF), *Indicating Verbs* (IV), *Lexical Reiteration* (REI), *Section Heading Preference* (SH), *Collocation Match* (CM), *Prepositional Noun Phrases* (PNP), *Immediate Reference* (IR), *Sequential Instructions* (SI), *Referential Distance* (RD), and *Term Preference* (TP).

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 168–186, 2002.

it is the antecedent of the current pronoun. Most of the indicators are genre-independent and related to coherence phenomena (such as salience and distance) or to structural matches, whereas others are genre-specific[3]. For a complete and detailed description see [21]. As an illustration, the indicator, *Immediate Reference* (IR) acts in a genre-specific manner and predicts that an NP appearing in a construction of the form "...(You) V_1 NP ... *con* (you) V_2 it (*con* (you) V_3 it)", where *con* ϵ {*and/or/before/after...*} will be the antecedent of a given pronoun. This preference is highly genre-specific and occurs frequently in imperative constructions such as "To turn on the printer, press *the Power button* and hold *it* down for a moment" or "Unwrap *the paper*, form *it* and align *it*, then load *it* into the drawer." This indicator, together with *collocation match* and *prepositional noun phrases* was most successful in pointing to the correct antecedent[4] of a given pronoun. In fact, initial results showed that the NP awarded a score by *immediate reference* always emerged as the correct antecedent.

The evaluation of Mitkov's knowledge-poor approach which was carried out by running the algorithm on post-edited outputs from the POS tagger and NP extractor, showed a success rate of 89.7% on a collection of texts, including the user guide referred to in Section 3 as PSW.

2 MARS: A Re-implemented and Improved Fully Automatic Version

Our project addresses the most crucial type of anaphora to NLP applications - that of identity-of-reference nominal anaphora, which can be regarded as the class of single-document identity coreference. This most frequently occurring class of anaphora has been researched and covered most extensively, and is the best understood within the field[5]. The current implementation of MARS is limited to pronoun resolution.

2.1 Fully Automatic Anaphora Resolution

MARS is a re-implemented version of Mitkov's robust, knowledge-poor approach which uses the FDG-parser [30] as its main pre-processing tool. MARS operates

[3] Typical of the genre of user guides.

[4] The confidence is computed in terms of decision power, which is a measure of the influence of each indicator on the final decision, its ability to 'impose' its preference in line with, or contrary to the preference of the remaining indicators. The decision power values partially served as a guide in proposing the numerical scores for each indicator. For a definition of this measure see [22].

[5] *Nominal anaphora* arises when a referring expression - pronoun, definite noun phrase, or proper name, has a non-pronominal noun phrase as antecedent. MARS does not handle identity-of-sense anaphora where the anaphor and the antecedent do not correspond to the same referent in the real world but to ones of a similar description as in the example "The man_i who gave his_i paycheck_j to his_i wife was wiser than the man_k who gave it_j to his_k mistress."

in a fully automatic mode, in contrast to the vast majority of approaches which rely on some kind of pre-editing of the text which is fed to the anaphora resolution algorithm[6] or which have only been manually simulated. As an illustration, Hobbs's naïve approach [17] was not implemented in its original version. In [7], [8], [1], and [19] pleonastic pronouns are removed manually[7], whereas in [21] and [12] the outputs of the PoS tagger and the NP extractor/partial parser are post-edited in a similar way to [20] where the output of the Slot Unification Grammar parser is corrected manually. Finally, [13] and [31] make use of annotated corpora and thus those approaches do not perform any pre-processing.

The development of MARS and also the re-implementation of fully automatic versions of Baldwin's as well as Kennedy and Boguraev's approaches for comparative purposes in another project [2], showed that fully automatic anaphora resolution is more difficult than previous work has suggested[8]. In the real-world fully automatic resolution must deal with a number of hard pre-processing problems such as morphological analysis/POS tagging, named entity recognition, NP gender identification, unknown word recognition, NP extraction, parsing, identification of pleonastic pronouns, selectional constraints, etc. Each one of these tasks introduces error and thus contributes to a reduction of the success rate of the anaphora resolution system; the accuracy of tasks such as robust parsing and identification of pleonastic pronouns is way below 100%[9]. For instance, many errors will be caused by the failure of systems to recognise pleonastic pronouns - and their consequent attempt to resolve them as anaphors.

2.2 Differences between MARS and the Original Approach

The initial implementation of MARS followed Mitkov's original approach more closely, the main differences being (i) the addition of three new indicators and (ii) a change in the way some of the indicators are implemented or computed due to the available pre-processing tools. In its most recent version, however, MARS uses a program for automatically recognising instances of non-nominal

[6] This statement refers to anaphora resolution systems and not to the coreference resolution systems implemented for MUC-6 and MUC-7.

[7] In addition, [8] undertook additional pre-editing such as removing sentences for which the parser failed to produce a reasonable parse, cases where the antecedent was not an NP etc.; [19] manually removed 30 occurrences of pleonastic pronouns (which could not be recognised by their pleonastic recogniser) as well as 6 occurrences of *it* which referred to a VP or prepositional constituent.

[8] By fully automatic anaphora resolution we mean that there is no human intervention at any stage: such intervention is sometimes large-scale such as manual simulation of the approach and sometimes smaller-scale as in the cases where the evaluation samples are stripped of pleonastic pronouns or anaphors referring to constituents other than NPs.

[9] The best accuracy reported in robust parsing of unrestricted texts is around the 86% mark; the accuracy of identification of non-nominal pronouns is under the 80% mark though [27] reported 92% for identification of pleonastic *it*.

pronominal anaphors and pleonastic pronouns[10], it incorporates two new syntax filters, and a program for automatic gender identification. Each of these new components is described in sections 2.2.1-2.2.4 below.

2.2.1 New Indicators

The three new indicators that were included in MARS are:

Boost Pronoun (BP): As NPs, pronouns are permitted to enter the sets of competing candidates for other pronouns. The motivation for considering pronominal candidates is twofold. Firstly, pronominalised forms represent additional mentions of entities and therefore increase their topicality. Secondly, the NP corresponding to an antecedent may be beyond the range of the algorithm, explicitly appearing only prior to the two sentences preceding the one in which the pronoun appears. Pronoun candidates may thus serve as a stepping-stone between the current pronoun and its more distant nominal antecedent. Of course, it is not helpful in any application for the system to report that the antecedent of a pronoun *it* is another pronoun *it*. When a pronoun is selected as the antecedent, the system has access to that pronoun's own antecedent in a fully transitive fashion so that a NP is always returned as the antecedent of a pronoun, even when this is accessed via one or more pronouns. Given that pronominal mentions of entities may reflect the salience of their antecedents, pronouns are awarded a bonus of +1.

Syntactic Parallelism (SP): The pre-processing software (FDG-Parser) used by MARS also provides the syntactic role of the NP complements of the verbs. This indicator increases the chances that a NP with the same syntactic role as the current pronoun will be its antecedent by awarding it a boosting score of +1.

Frequent Candidates (FC): This indicator was motivated by our observations during annotation of coreference that texts frequently contain a narrow "spine" of references, with perhaps less than three entities being referred to most frequently by pronouns throughout the course of the document. This indicator awards a boosting score (+1) to the three NPs that occur most frequently in the sets of competing candidates of all pronouns in the text (for a definition of 'set of competing candidates' see Section 2.3).

Five of the original indicators are computed in a different manner by MARS. In the case of the indicator *lexical reiteration*, in addition to counting the number of explicit occurrences of an NP, MARS also counted pronouns previously resolved to that NP. The conditions for boosting them remain the same.

Collocation Match (CM) was originally implemented to boost candidates found in the same paragraph as the pronoun, preceding or following a verb identical or morphologically related to a verb that the pronoun precedes or follows. CM was modified so that in the first step, for every appearance of a verb

[10] Examples of pleonastic *it* include non-referential instances as in 'It is important...', 'It is requested that...', 'It is high time that...' Examples of the pronoun *it* that exhibit non-nominal anaphora are the cases where the antecedent is not an NP but a clause or whole sentence.

in the document, the immediately preceding and immediately following heads (PHEAD and FHEAD respectively) of NP arguments are written to a file. In the case of prepositions, the immediately following NP argument is written. An extract from the resulting file is shown below:

```
VERB replace PHEAD you FHEAD it
VERB replace PHEAD battery FHEAD cover
VERB replace PHEAD printer FHEAD cartridge
VERB replace FHEAD cartridge
VERB replace PHEAD You FHEAD cartridge
VERB replace FHEAD battery
VERB replace PHEAD battery FHEAD it
VERB replace PHEAD You FHEAD battery
VERB replace PHEAD problem FHEAD battery
VERB replace PHEAD you FHEAD battery
VERB replace PHEAD this FHEAD cartridge
VERB replace PHEAD Ink FHEAD Cartridge
VERB replace FHEAD Cartridge
VERB replace FHEAD Ink
```

MARS then consults this data file when executing CM. When resolving the pronoun *it* in sentence 4 of the illustrative paragraph,

> (1) Do not touch the battery terminals with metal objects such as paper clips or keychains. (2) Doing so can cause burns or start a fire. (3) Carry batteries only within the printer or within their original packaging. (4) Leave *the battery* inside the printer until you need to charge or replace *it*.

the NP *the battery* is awarded a boosting score of +2 because the pronoun is in the FHEAD position with respect to the lemma of the verb *replace* and the lemma of the head of *the battery* also appears in the FHEAD position with respect to that verb in the database. Thus, the indicator applies on the basis of information taken from the whole document, rather than information only found in the paragraph.

We are currently investigating the generalisation of CM using semantic information from the WordNet ontology. The method under investigation involves post-processing the data file produced by CM so that each entry is replaced by the most general senses (unique beginners) in WordNet of its elements. It was assumed that patterns appearing with significant frequency in the post-processed file could be used in a more generalised version of CM in which predicates with pronoun arguments and competing candidates are associated with their unique beginners (which we will denote by *Pred-UB* and *Cand-UB* respectively). The data file is then consulted to see if the patterns *Cand-UB* - *Pred-UB* or *Pred-UB* - *Cand-UB* have a significant presence. Candidates involved in those patterns in the data file that have a significant frequency are awarded a boosting score.

Our experiments in using WordNet to generalise the CM indicator have not yielded an improvement in the system, and have diminished MARS's performance overall. There are three reasons for this. Firstly, we have not yet incorporated a word sense disambiguator into our system, though work is underway in that regard with reference to the method proposed in [29]. Instead, we associate each word with the first sense returned in the list by WordNet. Secondly, many of the senses appearing in the somewhat specialised domain of technical manuals are not present in the WordNet ontology. It would require the use of a more specialised ontology to obtain optimum performance from the system. Thirdly, we have taken the mean frequency of appearance of a pattern in the datafile as the threshold level of significance. It may be possible to improve performance by using more sophisticated methods such as TF.IDF for patterns with respect to all the texts at our disposal.

First NPs has been renamed *obliqueness* (OBL). Following centering theory [15], where grammatical function is used as an indicator of discourse salience, MARS now awards subject NPs a score of +2, objects a score of +1, indirect objects no bonus, and NPs for which the FDG parser is unable to identify a function a penalising score of -1[11].

A clause splitter is not yet incorporated into MARS, so a simplified version of the *referential distance* indicator is implemented, with distance being calculated only in terms of sentences rather than clauses and sentences.

Regarding the *term preference* indicator, in the first implementation of MARS, significant terms were obtained by identifying the words in the text with the ten highest TF.IDF scores. Candidates containing any of these words were awarded the boosting score. In the current implementation, it is the ten NPs that appear with greatest frequency in the document that are considered significant. All candidates matching one of these most frequent NPs are awarded the boosting score.

2.2.2 Classification of *It*

MARS includes a program that automatically classifies instances of the pronoun *it* as pleonastic, examples of non-nominal anaphora, or nominal anaphora [10].

The method was developed by associating each instance of it in a 368830 word corpus with a vector of feature values. 35 feature-value pairs are used, the values being computed automatically by our software. Each feature belongs to one of six different types. *Type 1 features* carry information about the position of the instance in the text. *Type 2 features* describe the number of elements in the surrounding text, such as complementisers and prepositions, which are indicative of the pronoun's class. *Type 3 features* display the lemmas of elements such as verbs and adjectives in the same sentence as the instance. *Type 4 features* show the parts of speech of the tokens surrounding the instance. *Type 5 features* indicate the presence or otherwise of particular sequences of elements, such as *adjective* + NP or *complementiser* + NP, following the instance. *Type 6 features* indicate the proximity of suggestive material such as *-ing* forms of verbs or complementisers, following the instance in the text. The 3171 resultant vectors were then manually classified as belonging to one of the following classes: *nominal anaphoric*; *clause anaphoric*; *proaction*, *cataphoric*; *discourse topic*; *pleonastic*; or *idiomatic/stereotypic*. This manually annotated set of instances constitutes the training file.

[11] Note that the FDG parser proposes grammatical functions for most words. The POS tagger used in the original version was not able to identify syntactic functions and first NPs were used as approximations of subjects.

The classification system works by rendering new feature-value vectors for previously unseen instances of *it* and using TiMBL [6] to classify them with respect to the instances in the training file. The overall classification rate was reported to be 78.74% using ten-fold cross-validation. Table 1 gives more details on the accuracy of this classification over the texts processed in the current study.

2.2.3 Syntactic Constraints

The following constraints proposed by Kennedy and Boguraev [19] that act as knowledge-poor approximations of Lappin and Leass's [20] syntax filters, were also implemented in MARS's latest version: *A pronoun cannot corefer with a co-argument*, *a pronoun cannot co-refer with a non-pronominal constituent which it both commands and precedes*, and *a pronoun cannot corefer with a constituent which contains it.* These constraints are applied before activating the antecedent indicators and after the gender and number agreement tests.

2.2.4 Identification of Animate Entities

Evans and Orasan [9] presented a robust method for *identifying animate entities* in unrestricted texts, using a combination of statistics from WordNet [11] and heuristic rules.

Here, seven unique beginners from WordNet were taken to contain senses that in the case of nouns, usually refer to animate entities, and in the case of verbs, usually take animate subjects. For the NPs in a text, their heads were scrutinised in order to count the number of animate/inanimate senses that they can be associated with. In the case of subject NPs, their predicates were scrutinised in a similar fashion. The information concerning the number of an entity's animate/inanimate senses was then used when classifying the entity as being either animate or inanimate. The heuristic rules examined the specific form of the NPs in the text, reporting whether or not they contained suggestive complementisers such as *who*, or whether they were in fact pronouns whose gender could be determined in a trivial way. Once each NP was associated with all of this information, a simple rule-based method was used to classify the NP as animate or inanimate.

Overall, the method was shown to be a useful step towards enforcing gender agreement between pronouns and potential antecedents. The method worked adequately over texts containing a relatively high number of animate entities (+5.13% success rate in anaphora resolution), but it was ineffective over texts with relatively few animate entities as a result of the incorrect elimination of valid antecedents (-9.21% success rate on the technical document referred to in Section 3 as PSW).

In subsequent work, Orasan and Evans [26] refined the method for gender identification. In the original method, the unique beginners in WordNet were manually classified as *animate* or *inanimate* in line with the crude expectation that all their hyponyms were likely to refer to animate or inanimate entities. This approach was flawed in that the classification of a unique beginner is not a very reliable indicator of the classifiction of all of its hyponyms. Addressing this problem, the new effort used files from the sense-annotated SEMCOR corpus. Head nouns and verbs in those files were then manually annotated as either animate or inanimate depending upon their use in the texts. Chi-squared was then used to classify the hypernyms of the senses whose animacy was known. More specific senses were then taken to share the classification of the hypernyms. Machine learning was coupled with an approach similar to that described in [9] in order to make an automatic classification of NPs in unseen texts. The method described in [26] obtained an accuracy of around 97% in identifying animate entities.

Despite the greater accuracy of this method, we found that it still hinders MARS's performance in the domain of technical manuals, as was the case for the earlier work. Although, with respect to the PSW text, the error rate dropped from 9.21% to 1.33%, application of the method still induces deterioration in system performance in the domain of technical manuals. There are two reasons for this. Firstly, the technical domain refers to specialised senses that cannot be found in WordNet. Secondly, for those senses that are found, they are usually used with a highly specialised meaning. In many cases there is strong evidence from WordNet that nouns such as *computer* or *printer* are normally used to refer to animate entities when in fact they are only used with inanimate senses in computer technical manuals. It may be possible to improve the performance of the system by first performing word sense disambiguation (WSD) in order to limit the number of animate senses that particular nouns are permitted to have with respect to documents from particular domains. Work is currently underway to implement the method for WSD described in [29].

Due to these problems, our methods for identification of animate entities have not been incorporated when running MARS over the technical documents described in Section 3. Instead, gender agreement was only enforced using a gazetteer of first names.

2.3 The Algorithm

MARS operates in five phases. In *phase 1*, the text to be processed is parsed syntactically, using Conexor's FDG Parser [30] which returns the parts of speech, morphological lemmas, syntactic functions, grammatical number, and most crucially, dependency relations between tokens in the text which facilitates complex noun phrase (NP) extraction.

In *phase 2*, anaphoric pronouns are identified and non-anaphoric and non-nominal instances of *it* are filtered using the machine learning method described in [10]. In its current implementation, MARS is only intended to resolve third person pronouns and possessives of singular and plural number that demonstrate identity-of-reference nominal anaphora.

In *phase 3*, for each pronoun identified as anaphoric, potential antecedents (candidates), are extracted from the NPs in the heading of the section in which the pronoun appears, and from NPs in the text preceding the pronoun up to the limit of either three sentence boundaries or one paragraph boundary, whichever contains the smallest amount of text. Once identified, these candidates are subjected to further morphological and syntactic tests. Extracted candidates are expected to obey a number of constraints if they are to enter the *set of competing candidates*, i.e. the candidates that are to be considered further. Firstly, competing candidates are required to agree with the pronoun with respect to number and gender, as was the case in the original version of MARS. Secondly, they must obey the syntactic constraints described in Section 2.2.3.

In *phase 4*, preferential and impeding factors (a total of 14) are applied to the sets of competing candidates. On application, each factor applies a numerical score to each candidate, reflecting the extent of the system's confidence about whether the candidate is the antecedent of the current pronoun.

Finally, in *phase 5*, the candidate with the highest composite score is selected as the antecedent of the pronoun. Ties are resolved by selecting the most recent highest scoring candidate.

2.4 Using Genetic Algorithms to Search for Optimal Performance

The scores of the antecedent indicators as proposed in Mitkov's original method were derived on the basis of empirical observations, taking their decision power into consideration, and have never been regarded as definite or optimal. By changing the scores applied by the antecedent indicators, it is possible to obtain better success rates.

Given that the score of a competing candidate is computed by adding the scores applied by each of the indicators, the algorithm can be represented as a function with 14 parameters, each one representing an antecedent indicator

$$score_k = \sum_{i=1}^{i=14} x_{k_i} \tag{1}$$

where $score_k$ is the composite score assigned to the candidate k, and x_{k_i} is the score assigned to the candidate k by the indicator i. The goal of a search method would be to find the set of indicator scores for which the composite score is maximum for the antecedents and lower for the rest of candidates. This would lead to a high success rate.

Genetic algorithms (GA) seemed the most appropriate way of finding the optimal solution. First proposed by Holland [18], GA mimic reproduction and selection of natural populations to find the solution that maximises a function, called fitness. The GA maintains a population of candidate solutions to the fitness function represented in the form of chromosomes. For our problem, each chromosome, representing a set of indicator scores, is a string of 34 real numbers; each value representing the outcome of an indicator application. The alphabet used to represent chromosomes is the set of real numbers. As a fitness function we used the number of anaphors correctly resolved by the system when a candidate solution's indicator scores are applied by the algorithm. Therefore, maximisation of the fitness function leads to an increase in the success rate.

The main use of the GA is to find the upper limits of a method based on numerical preferences. In this case, the algorithm does not try to find a general set of scores that could be useful over general texts. Instead, it searches the solution space for a set which maximises the fitness function (success rate) for a certain text. This value represents the maximum success rate that the given preference-based algorithm can obtain for that text. A secondary usage of the GA is as an optimisation method. In this case, the set of indicators which maximises the success rate for a particular file is applied by the algorithm when processing different files. The results of such cross-evaluation are presented in Section 3 and discussed in Section 4.

3 Evaluation

MARS was evaluated on eight different files, from the domains of computer hardware and software technical manuals, featuring 247,401 words and 2,263 anaphoric pronouns (Table 1). Each text was annotated coreferentially in accordance with the methodology reported in [23]. Applied over this corpus, MARS obtained an average success rate of 59.35%. Success rate is defined as the ratio of the number of anaphoric pronouns that MARS resolves correctly to the number of anaphoric pronouns in the text. We do not take the number of pronouns that the system attempts to resolve as the denominator because this would mean that a system that only attempted to resolve pronouns with a single candidate could obtain unfairly high levels of performance.

Each technical manual is identified by an abbreviation in column 1 of Table 1. Column 2 shows the size of the text in words, column 3 displays the number of anaphoric pronouns[12], column 4 shows the number of pronouns in the text that are instances of non-nominal anaphora or pleonastic *it*. Column 5 shows the accuracy with which the system is able to classify instances of the pronoun *it*. The reader will note that these figures are markedly improved over those reported in [10]. This is explained by the fact that in that paper, the system was tested over texts from many different genres, which included free narrative and direct speech. In the domain of technical manuals, instances of *it* are found in far more constrained and predicable linguistic contexts, resulting in greater reliability on the part of the machine learning method. Of the anaphoric pronouns, 1709 were intrasentential anaphors and 554 - intersentential. In 238 cases the antecedents were not on the list of candidates due to pre-processing errors.

The overall success rate of MARS was 59.35% (1343/2263). After using GA [25], the success rate rose to 61.55% (1393/2263). Table 2 gives details on the evaluation of MARS - covering the standard version and the version in which the GA was used to obtain the set of scores leading to optimal performance. As a result of errors at the level of NP extraction, and therefore possible omission of antecedents, the success rate of MARS cannot reach 100%. In the MAX column, the theoretical maximum success rate that MARS can obtain as a result of pre-processing errors is indicated. The column *Sct* represents the maximum possible

[12] More accurately, pronouns that demonstrate nominal identity-of-reference anaphora.

Table 1. The characteristics of the texts used for evaluation

Text	#Words	#Anaphoric pronouns	#Non-nominal anaphoric/pleonastic *it*	Classification accuracy for *it*
ACC	9753	157	22	81.54%
BEO	7456	70	22	83.02%
CDR	10453	83	7	92.86%
GIMP	155923	1468	313	83.42%
MAC	15131	149	16	89.65%
PSW	6475	75	3	94.91%
SCAN	39328	213	22	95.32%
WIN	2882	48	3	97.06%
Total	247401	2263	408	85.54%

success rate when a pronoun is considered correctly resolved only if the whole NP representing its antecedent is selected as such, in its entirety. As can be seen, this figure does not exceed 92%. Given the preprocessing errors, inevitable in an automatic system, we considered a pronoun correctly resolved if only part of a pronoun's antecedent was identified and that part included the head of the NP (as proposed in MUC-7 [16]). When this partial matching is considered, the maximum success rate can reach the values presented in the column *Ptl.* Two baseline models, presented in the *Baseline* column, were evaluated, one in which the most recent candidate was selected as the antecedent (*Rcnt*) and one in which a candidate was selected at random (*Rand*) - both after agreement restrictions had been applied.

The column *Old* displays the performance of a fully automatic implementation of the algorithm proposed in [21]. We should emphasise that it follows the method briefly discussed in Section 1 without including any additional components such as new or modified indicators or recognition of pleonastic pronouns. The values in this column are noticeably lower than those obtained for any of the subsequent systems.

We evaluated MARS in four different configurations: Default (*Dflt*), in which the system described in Section 2.3 is run in its entirety; *no it filter*, where the system is run without attempting to identify pleonastic/non-nominal instances of *it*; *no num/gend agr*, where the system is run without applying number and gender agreement constraints between pronouns and competing candidates, and *no syn constr*, where no syntactic constraints are enforced between pronouns and intrasentential candidates. Of course, more combinations are possible, but due to space and time constraints, we did not evaluate them. By comparing these columns with the *dflt* column, for example, it is possible to see that, overall, MARS gains around 30% in performance as a result of enforcing number and gender agreement between pronouns and competing candidates. For each configuration and each text, we obtained MARS's success rate, displayed in the column *Standard.* Additionally, we used the GA described in Section 2.4 to find the upper limit of MARS's performance when the optimal set of indicator scores is applied, displayed in the column *Upper bound.* In this case, the GA was used

Table 2. Success rates for different versions of MARS

Files	Old (2000)	MARS								MAX		Baseline	
		Standard				Upper bound							
		Dflt	no *it* filter	no num /gend agr	no syn constr	Dflt	no *it* filter	no num /gend agr	no syn constr	Sct	Ptl	Rcnt	Rand
ACC	33.33	51.59	52.87	35.67	49.04	55.41	55.41	43.31	43.31	73.88	96.18	28.02	26.75
BEO	35.48	60.00	60.00	45.71	60.00	67.14	64.28	50.00	67.14	81.43	95.71	35.71	22.86
CDR	53.84	67.47	68.67	51.81	67.47	75.90	74.69	54.22	74.69	78.31	95.18	36.14	43.37
GIMP	-	57.15	60.42	17.57	57.63	57.83	60.83	18.94	57.22	79.70	91.69	37.80	30.72
MAC	53.93	71.81	69.79	60.40	71.14	75.84	77.85	67.11	76.51	83.89	96.64	51.68	44.97
PSW	64.55	82.67	84.00	80.00	82.67	86.67	90.67	80.00	89.33	92.00	97.33	49.33	45.33
SCAN	-	61.50	62.44	46.48	60.56	63.85	64.79	51.64	63.85	79.81	87.32	32.39	30.52
WIN	33.32	52.08	62.50	39.58	52.08	68.75	66.67	60.42	68.75	81.25	87.50	37.50	18.75
TOTAL	45.81	59.35	61.82	29.03	59.35	61.55	63.68	32.04	60.41	80.03	92.27	37.78	31.82

as a search algorithm and not as a general optimisation method. It allowed us to explore the limitations of this knowledge poor pronoun resolution system.

The optimal indicator scores obtained after applying the GA to a specific text were applied when running the algorithm on different texts, in order to make a blind test and to ascertain the general usefulness of genetic optimisation. The results of the cross-evaluation were quite disappointing.

Table 3. The results of cross-evaluation

Inds/ Texts	ACC	BEO	CDR	MAC	PSW	WIN	SCAN	GIMP
ACC	55.41	47.77	47.13	45.22	42.67	45.86	44.59	51.59
BEO	48.57	67.14	52.86	45.72	51.43	60.00	58.57	65.71
CDR	60.24	71.08	75.90	48.19	57.83	57.83	62.65	71.08
MAC	61.74	64.43	63.76	75.84	63.09	61.74	65.77	65.77
PSW	81.33	73.33	72.00	77.33	86.67	74.67	74.67	78.67
WIN	41.67	47.92	52.08	47.92	43.75	68.75	52.08	52.08
SCAN	50.23	55.87	54.46	54.93	47.42	54.93	63.85	53.05
GIMP	51.43	55.04	51.91	53.06	49.80	51.77	50.89	57.83

In most cases the success rates obtained were lower that the ones obtained by the *Standard* version of MARS. The application of the GA will be discussed further in Section 4.

3.1 The Influence of Indicators

Relative importance is a measure showing how much the system's performance is degraded when an indicator is removed from the algorithm [24][13]. We computed

[13] Similar to the measure used in [20].

Table 4. Standard relative importance

W/O	ACC	BEO	CDR	MAC	PSW	WIN	SCAN	GIMP	TOTAL
INDEF	-0.64%	-1.43%	0%	-2.01%	+3.95%	0%	-1.88%	+0.14%	-0.18%
OBL	+7.01%	+11.43%	+6.02%	-2.01%	-1.31%	-10.42%	+7.98%	+4.90%	+0.62%
IV	0%	0%	0%	0%	0%	0%	0%	+0.14%	+neg%
REI	-2.55%	-2.86%	+2.41%	+2.01%	-1.31%	-10.42%	-1.41%	+0.27%	-0.26%
SH	-0.64%	+2.86%	+2.41%	+0.67%	0%	-6.25%	+0.94%	+0.82%	+0.66%
PNP	0%	0%	0%	-3.35%	0%	0%	-0.47%	+0.48%	+neg%
CM	+1.27%	0%	0%	+0.67%	+2.63%	+2.08%	+3.29%	+0.82%	+1.10%
IR	0%	0%	-1.20%	+2.01%	0%	0%	+0.47%	+0.14%	+0.22%
SI	0%	0%	0%	0%	0%	0%	0%	0%	0%
RD	+3.18%	+5.71%	+1.20%	+1.34%	+2.63%	+12.50%	+3.29%	+5.31%	+4.64%
TP	0%	0%	-2.40%	-0.67%	0%	+2.08%	0%	-0.61%	-0.49%
BP	+3.82%	0%	+2.40%	-0.67%	0%	0%	+0.47%	+0.54%	+0.71%
SP	+1.27%	0%	+1.20%	-1.34%	-1.31%	+2.08%	+2.35%	+1.02%	+0.93%
FC	+0.64%	0%	0%	-0.67%	0%	+2.08%	0%	-0.34%	-0.18%

this measure for each indicator and each file, before and after the GA was applied. In some cases, there were negative values for relative importance reflecting the fact that in some isolated cases, depending on the particular characteristics of the text, removing one of the indicators actually improved MARS's performance. The relative importance of each indicator is displayed in Table 4 (before the GA is applied) and Table 5 (following application of the GA). Our findings are discussed in Sections 3.1.1 and 3.1.2.

Interestingly, after we had made the assessment of the importance of each indicator, and deactivated those with no importance or negative importance so that only the positively important were in effect, overall, MARS performed slightly worse than when all indicators were active (success rate of 59.21 vs. 59.35).

3.1.1 Original Indicators Our examination of the relative importance of each indicator with respect to each file showed that for the *Standard* version of MARS, the most important of the original indicators was SH in most of the cases. Due to the differences in the current implementation of RD, and its original statement, the importance of that indicator is discussed in 3.1.2. On the texts used for evaluation, the relative importance of SI and INDEF is negligible. The rest of the indicators have a moderate influence. A similar observation can be made for the version of the algorithm after the GA was applied, though the difference in importance between indicators is somewhat reduced. SH, PNP, and IR are the most important of the original indicators after application of the GA.

3.1.2 New/Modified indicators. With respect to the new and modified indicators presented for the first time in this paper, we noted the following. RD, even without access to information on a sentence's internal structure, is the most important of the modified indicators, followed by CM. Although of variable importance over different texts, overall, OBL and SP make a positive contribution

Table 5. Relative importance after the GA was applied

W/O	ACC	BEO	CDR	MAC	PSW	WIN	SCAN	GIMP	TOTAL
INDEF	0%	0%	0%	0%	+1.33%	-2.08%	+1.88%	-0.27%	0%
OBL	+2.55%	-1.43%	+1.20%	-1.34%	+1.33%	+6.25%	+2.82%	+1.97%	+1.81%
IV	0%	-1.43%	0%	-2.01%	0%	0%	+0.47%	+0.82%	+0.40%
REI	0%	-1.43%	-1.20%	+1.34%	0%	0%	0%	-0.34%	-0.22%
SH	+1.27%	0%	+3.61%	-1.34%	0%	-2.08%	0%	+0.27%	+0.26%
PNP	-1.27%	0%	-2.40%	-0.67%	+1.33%	-2.08%	+1.41%	+0.14%	0%
CM	+1.27%	-1.43%	0%	-1.34%	+1.33%	0%	+1.88%	+2.11%	+1.55%
IR	+0.64%	-1.43%	-1.20%	0%	0%	0%	+0.94%	+1.29%	+0.88%
SI	-0.64%	-1.43%	0%	0%	0%	0%	+1.41%	+0.48%	+0.35%
RD	+1.27%	+10.00%	+2.40%	+4.03%	+5.33%	+8.33%	+5.63%	+6.33%	+5.74%
TP	+1.27%	0%	0%	-0.67%	0%	0%	+1.88%	+1.02%	+0.88%
BP	+1.27%	-2.86%	-1.20%	+0.67%	0%	-2.08%	+1.41%	-0.20%	-neg%
SP	-1.27%	0%	+1.20%	+0.67%	0%	0%	+3.29%	-0.14%	+0.22%
FC	-0.64%	-2.86%	0%	0%	+1.33%	-2.08%	+1.88%	-2.11%	-1.31%

in both the *Standard* and *Upper bound* versions of MARS. On the other hand, REI has negative importance. We can account for this because the pronoun resolution process is itself imprecise and the fact that REI counts pronouns resolved by MARS to NPs as additional mentions of those NPs will make it somewhat inaccurate. Perhaps for similar reasons, the importance of BP was variable, having positive importance in the *Standard* version and negligibly negative importance in the *Upper bound* version. The importance of TP was negative in the *Standard* version of MARS but positive in the *Upper bound* version. It is very probable that the implementation of this indicator can be improved by using better algorithms to identify the significant terms in the texts. Of variably negative and positive importance when applied over different texts, the FC indicator was of negative importance overall, despite the observations and justification for this indicator presented in Section 2.2.1.

3.2 The Influence of an Automatic Classification of *it*

The reader will note, by comparison of columns 3 and 4 in Table 2, that MARS's performance is slightly better, in terms of success rate, when no attempt is made at recognition of pleonastic/non-nominal *it*. Overall, as a result of classifying *it*, the success rate drops by more than 2%. This is due to inaccuracies in the classification module with some anaphoric instances of *it* being incorrectly filtered. In light of this, one may conclude that the pronoun classification module should be eliminated. However, we argue that the reader is drawn to this conclusion by inadequacies in the definition of success rate. In Section 4, we argue that success rate cannot capture the positive contribution made by the classification module and a new measure of performance is proposed.

3.3 The Influence of Syntactic Constraints

In Table 2, the column *no syn constr* shows MARS's performance when the syntactic constraints described in Section 2.2.3 are not applied between pronouns and their competing candidates. Comparison of the *Dflt* columns with these shows the scale of the contribution to the system made by syntactic and agreement constraints. The contribution made by the syntactic constraints (around +2% success rate overall for the *Upper bound* version of MARS and no contribution in the *Standard* version) is not as great as may be expected. This is due to their reliance on an accurate global parse of sentences, which was not always obtained for the texts that we processed.

4 Discussion

The evaluation results give rise to a number of interesting conclusions that can be made with regard to the approach presented and with regard, more generally to anaphora resolution.

To start with, a close look at the MAX columns in Table 2 clearly shows the limits of fully automatic anaphora resolution, based on a given pre-processing tool, with candidates extracted from a range of two sentences from the pronoun. Systems depend on the efficiency of the pre-processing tools which analyse the input before feeding it to the resolution algorithm. Inaccurate pre-processing can lead to a considerable drop in the performance of the system, however accurate an anaphora resolution algorithm may be. The accuracy of today's pre-processing is still unsatisfactory from the point of view of anaphora resolution. Whereas POS taggers are fairly reliable, full or partial parsers are not. Named entity recognition is still a challenge, with the development of a product name recogniser being a vital task for a number of genres. While recent progress in areas such as identification of pleonastic pronouns [10], identification of non-anaphoric definite descriptions [3]; [32] and recognition of animacy [9] have been reported, these tasks and other vital pre-processing tasks such as gender recognition and term recognition, have a long way to go. For instance, the best accuracy reported in robust parsing of unrestricted texts is around the 86% mark [5]; the accuracy of identification of non-nominal pronouns normally does not exceed 80%[14] [10]; though the accuracy of identification of NP gender has reached 97% [26]. Other tasks may be more accurate but are still far from perfect. The state of the art of NP chunking which does not include NPs with post-modifiers, is 90-93% in terms of recall and precision. The best-performing named entity taggers achieve an accuracy of about 96% when trained and tested on news about a specific topic, and about 93% when trained on news about one topic and tested on news about another [14]. Finally, comparison of MARS which employs arguably one of the best shallow parsers for English with Mitkov's original approach which operated on correctly pre-processed texts, shows a drop of up to 25% of the success rate!

[14] However, Paice and Husk [27] reported 92% for identification of pleonastic *it*.

The results also show that the reported success rate is reduced if we consider resolution correct only if the full NP representing the antecedent is identified and if similarly to MUC-7 [16], the task is not simplified to tracking down only a part of the full NP as long as that part contains the head.

The use of the GA allowed us to gain an insight into the limits of this preference-based anaphora resolution method. It was shown that by choosing the right set of indicator scores, it is possible to improve the success rate of the system by up to 3% over all files tested. However, at this stage, we cannot find a method which can determine the optimal set of scores for unseen texts. Cross-evaluation showed that the optimal scores derived by the GA for a text are specific to it and attempts to use them when processing different texts led to low success rates. This result can be explained by over-fitting on the part of the GA with respect to the characteristics of a particular text. Further research on this topic is necessary in order to design a generally applicable optimisation method.

We should note that MARS employs a knowledge-poor algorithm: we do not have any access to real-world knowledge, or even to any semantic knowledge. MARS does not employ full parsing either and works from the output of a POS tagger enhanced with syntactic roles (in most cases) and functional dependency relations. Recent research [28] shows that approaches operating without any semantic knowledge (e.g. in the form of selectional restrictions) usually do not achieve a success rate higher than 75%. In light of this, we find MARS's success rate on a number of files to be encouraging.

The evaluation carried out raises another important issue. We have adopted the measure of success rate since it has been shown [24]; [4] that recall and precision are not always suitable for anaphora resolution. The current definition of success rate as the number of successfully resolved pronouns divided by the total number of pronouns (as marked by humans), however, does not capture cases where the program incorrectly tries to resolve instances of non-nominal anaphora. For programs handling nominal anaphora, we feel it is important to be able to judge the efficiency of the program in terms of removing instances of non-nominal anaphora and not incorrectly attempting to resolve these instances to NPs. Therefore, we believe that a measure which reflects this efficacy would be appropriate.

If an anaphora resolution system is presented with a set P of pronouns, where the subset A are instances of nominal anaphora and subset N are not nominally anaphoric, it may be useful to assess that system using a measure that captures the correctness of its response to all P pronouns. Ideally, such a system will attempt to resolve the set A and filter out the set N. If the system correctly resolves A' of the nominally anaphoric pronouns and correctly filters, N' of the non-nominally anaphoric ones, it can be evaluated using the single ratio, which we call *Resolution Etiquette*, $RE = 100*(N'+A')/P$. This measure captures the contribution made to the system by both recognition modules for non nominal and pleonastic pronouns and the anaphora resolution module itself, in a way that our previous measure, *success rate* (SR), did not. This measure is intended

Table 6. Evaluation of different configurations of MARS using SR and RE

File	Default		no *it* filter	
	SR	RE	SR	RE
ACC	51.59	49.17	52.87	45.86
BEO	60.00	60.21	60.00	45.16
CDR	67.47	67.03	68.67	62.64
GIMP	57.15	56.03	60.42	49.75
MAC	71.81	70.30	69.79	63.03
PSW	82.67	81.01	84.00	79.75
SCAN	61.50	62.13	62.44	56.60
WIN	52.08	54.90	62.50	58.82
TOTAL	59.35	58.21	61.82	52.24

to describe a system's ability to "behave appropriately" in response to a set of pronouns. Table 6 compares the *success rate* and *resolution etiquette* scores obtained by MARS when run with and without the recognition component for non-nominal and non-anaphoric pronouns.

It should be pointed out that a direct comparison between SR and RE is not appropriate. The purpose of Table 6 is not to compare them, but to show the ability of RE to capture the contribution made by the pronoun classification module.

When the pronoun classification module is deactivated, we notice an increase in SR. This is caused because the pronoun classification module incorrectly filters some nominal-anaphors. By definition, SR can only capture errors made by the classification module; its successful filtration of non-nominal anaphora/pleonastic pronouns is ignored by that measure. In contrast, the measure RE is much reduced when the pronoun classification module is deactivated. Even though the module incorrectly filters some nominally anaphoric pronouns, this side effect is outweighed by the correct filtration of non-nominal and pleonastic pronouns. Deactivating the module reduces MARS's ability to respond appropriately to the pronouns it is presented with, making it less useful in further NLP applications such as MT, information retrieval, information extraction, or document summarisation. The RE measure reflects this whereas SR does not. However, we appreciate that as this is a new measure, a comparison of MARS with other systems, using this measure, is not possible.

5 Conclusion

A new, advanced, and fully automatic version of Mitkov's knowledge-poor approach to pronominal anaphora resolution has been proposed in this paper. We have argued that there is a big difference between previously proposed anaphora resolution methods that were tested over small texts, in which most of the preprocessing steps were post-edited, and fully automatic systems which have to deal with messy data, and errors. The new method has been thoroughly evaluated with respect to 8 technical manuals. By means of a GA, the practical limitations of the system have been revealed. As a result of the insights gained during the evaluation phase, a new measure that is argued to better reflect the performance of fully automatic anaphora resolution systems has been proposed.

References

1. Aone, C. and Bennett, S. W. (1995) Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the ACL (ACL'95)*, pp. 122-129. ACL.
2. Barbu, C. and Mitkov, R. (2000) Evaluation environment for anaphora resolution. In *Proceedings of the International Conference on Machine Translation and Multilingual Applications (MT2000)*, 18-1-18-8. Exeter, UK.
3. Bean, D. L. and Riloff, E. (1999) Corpus-based Identification of Non-anaphoric Noun Phrases. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL'99).* pp. 373-380. ACL.
4. Byron, D. (2001) A proposal for consistent evaluation of pronoun resolution algorithms. *Computational Linguistics.* Forthcoming. MIT Press.
5. Collins, M. (1997) Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, Madrid.
6. Daelemans, W. (1999) *TiMBL: Tilburg Memory Based Learner version 2 Reference Guide*, ILK Technical Report - ILK 99-01, Tilburg University, The Netherlands
7. Dagan, I. and Itai, A. (1990) Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, Vol. III, 1-3, Helsinki, Finland.
8. Dagan, I. and Itai, A. (1991) A statistical filter for resolving pronoun references. In Y.A. Feldman and A. Bruckstein (Eds) *Artificial Intelligence and Computer Vision*, pp. 125-135. Elsevier Science Publishers B.V. (North-Holland).
9. Evans R. and Orasan, C. (2000) Improving anaphora resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC 2000).* pp. 154-162. Lancaster, UK.
10. Evans, R. (2001) Applying Machine Learning Toward an Automatic Classification of It. *Journal of Literary and Linguistic Computing.* 16(1) pp. 45-57. Oxford University Press.
11. Fellbaum, C. (ed) (1998) *WordNet An Eletronic Lexical Database.* MIT Press
12. Ferrández, A., Palomar, M., and Moreno, L. (1998) Anaphora resolution in unrestricted texts with partial parsing. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference*, pp. 385-391. Montreal, Canada.
13. Ge, N., Hale, J. and Charniak, E. (1998) A statistical approach to anaphora resolution. In *Proceedings of the Workshop on Very Large Corpora*, pp. 161-170. Montreal, Canada.
14. Grishman, R. (Forthcoming) Information Extraction. In R. Mitkov (Ed.), *Oxford Handbook of Computational Linguistics.* Oxford University Press, forthcoming.
15. Grosz, B. J., Joshi, A., and Weinstein, S. (1995) Centering: a framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2), pp. 44-50. MIT Press.

16. Hirschman, L. and Chinchor, N. (1997) *MUC-7 Coreference Task Definition* at *http://www.itl.nist.gov/iaui/*894.02*/related_projects/muc/proceedings/co_task.html*
17. Hobbs, J. R. (1978) Resolving pronoun references. *Lingua*, 44, pp. 339-352.
18. Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, University of Mitchigan Press, US.
19. Kennedy, C. and Boguraev, B. (1996) Anaphora for everyone: pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pp. 113-118. Copenhagen, Denmark.
20. Lappin, S. and Leass, H.J. (1994) An Algorithm for Pronominal Anaphora Resolution, in *Computational Linguistics* Volume 20, Number 4
21. Mitkov, R. (1998) Robust pronoun resolution with limited knowledge. In *Proceedings of the 18.th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference*, pp. 869-875. Montreal, Canada.
22. Mitkov, R. (2000) Towards more comprehensive evaluation in anaphora resolution. In Proceedings of *The Second International Conference on Language Resources and Evaluation, volume III*, pp. 1309-1314, Athens, Greece. ELRA.
23. Mitkov, R., Evans, R., Orasan, C., Barbu, C., Jones, L., and Sotirova, V. (2000) Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pp. 49-58. Lancaster, UK.
24. Mitkov, R. (2000) Towards a more consistent and comprehensive evaluation of anaphora resolution algorithms and systems. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pp. 96-107. Lancaster, UK.
25. Orasan C., Evans, R. and Mitkov, R. (2000) Enhancing Preference-Based Anaphora Resolution with Genetic Algorithms. In *Proceedings of NLP'2000*, Patras, Greece. pp. 185-195.
26. Orasan, C. and Evans, R. (2001) Learning to identify animate references. In *Proceedings of the Workshop Computational Natural Language Learning 2001 (CoNLL-2001).* ACL. Toulouse.
27. Paice, C.D. And Husk, G.D. (1987) Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun 'it,' in *Computer Speech and Language*, 2 pp. 109-132, Academic Press, US.
28. Palomar, M., Moreno, L., Peral, J., Munoz, R., Ferrandez, A., Martinez-Barco, P., and Saiz-Noeda, M. (2001) An algorithm for anaphora resolution in Spanish texts. Forthcoming.
29. Resnik, P. (1995) Disambiguating noun groupings with respect to WordNet senses. In *Proceedings of the Third Workshop on Very Large Corpora.* ACL. New Jersey. pp. 54-68.
30. Tapanainen, P. and Järvinen, T. (1997) A Non-Projective Dependency Parser, in The Proceedings of The *5th Conference of Applied Natural Language Processing*, pages 64-71, ACL, US.
31. Tetreault, J. R. 1999. Analysis of Syntax-Based Pronoun Resolution Methods. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pp. 602 - 605, ACL, University College Maryland. US.
32. Vieira, R. and Poesio, M. (2000) An Empirically-Based System for Processing Definite Descriptions. *Computational Linguistics*, v. 26, n.4.

Pronominal Anaphora Generation in an English-Spanish MT Approach*

Jesús Peral and Antonio Ferrández

Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante. Alicante, Spain
{jperal,antonio}@dlsi.ua.es

Abstract. This paper presents the pronominal anaphora generation module in a Machine Translation (MT) system. The MT interlingua approach –AGIR (*Anaphora Generation with an Interlingua Representation*)– allows the generation of anaphoric expressions into the target language from the interlingua representation of the source text.
AGIR uses different kinds of knowledge (lexical, syntactic, morphological and semantic information) to solve the Natural Language Processing (NLP) problems of the source text. Subsequently, an interlingua representation of the whole text is obtained that allows the correct generation of anaphoric expressions. In this paper we have evaluated the generation of English and Spanish (including zero pronouns) third person personal pronouns into the target language. We have obtained the following results: a precision of 80.39% and 84.77% in the generation of Spanish and English pronominal anaphora respectively.

1 Introduction

One of the main problems of many commercial and experimental Machine Translation (MT) systems is that they do not carry out a correct pronominal anaphora generation. Solving the anaphora and extracting the antecedent are key issues in a correct generation into the target language. Unfortunately, the majority of MT systems do not deal with anaphora resolution and their successful operation usually does not go beyond the sentence level. This paper presents a complete approach that allows pronoun resolution and generation into the target language.

Our approach works on unrestricted texts unlike other systems, the KANT interlingua system [8], the Météo system [3], the Candide system [2], etc. that are designed for well-defined domains. Although full parsing of these texts could be applied, we have used partial parsing of the texts due to the unavoidable incompleteness of the grammar. This is a main difference with the majority of the interlingua systems such as the DLT system based on a modification of Esperanto [15], the Rosetta system which is experimenting with Montague semantics as the basis for an interlingua [1], the KANT system, etc. as they use full parsing of the text.

* This paper has been partially supported by the Spanish Government (CICYT) project number TIC2000-0664-C02-02.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 187–196, 2002.

After the parsing and solving pronominal anaphora, an interlingua representation of the whole text is obtained. From this interlingua representation, the generation of anaphora (including intersentential anaphora), the detection of coreference chains of the whole text and the generation of Spanish zero-pronouns into English have been carried out, issues that are hardly considered by other systems. Furthermore, this approach can be used for other different applications, e.g. Information Retrieval, Summarization, etc.

The paper is organized as follows: in section 2 and section 3 the Analysis and Generation modules of our approach are presented. In section 4, the Generation module has been evaluated in order to measure the efficiency of our proposal. Finally, the conclusions of this work will be presented.

2 AGIR's Analysis Module

The AGIR (*Anaphora Generation with an Interlingua Representation*) system architecture is based on the general architecture of a MT system that uses an interlingua strategy. Translation is carried out in two stages: from the source language to the interlingua, and from the interlingua into the target language. Modules for analysis are independent from modules for generation.

In AGIR, the analysis is carried out by means of SUPAR (*Slot Unification Parser for Anaphora resolution*) system [4]. SUPAR is a computational system focused on anaphora resolution. It can deal with several kinds of anaphora, such as pronominal anaphora, one-anaphora, surface-count anaphora and definite descriptions. In this paper, we focus on pronominal anaphora resolution and generation into the target language.

A grammar defined by means of the grammatical formalism SUG (*Slot Unification Grammar*) is used as input of SUPAR. A translator that transforms SUG rules into Prolog clauses has been developed. This translator will provide a Prolog program that will parse each sentence. SUPAR allows to carry out either a full or a partial parsing of the text, with the same parser and grammar. Here, partial parsing techniques have been used due to the unavoidable incompleteness of the grammar and the use of unrestricted texts (corpora) as inputs.

The first stage of the analysis module is the lexical and morphological analysis of the input text. Due to the use of unrestricted texts as input, the system obtains the lexical and morphological information of the text's lexical units from the output of a part-of-speech (POS) tagger. The word, as it appears in the corpus, its lemma and its POS tag (with morphological information) is supplied for each lexical unit in the corpus.

The next stage is the parsing of the text (it includes the lexical and morphological information extracted in the previous stage). The corpus is split into sentences before applying the parsing. The output of the parsing stage will be the Slot Structure (SS) that stores the necessary information[1] for Natural Lan-

[1] The SS stores for each constituent the following information: constituent name (NP, PP, etc.), semantic and morphological information, discourse marker (identifier of the entity or discourse object) and the SS of its subconstituents.

guage Processing (NLP) problem resolution. This SS will be the input for the following stage in which NLP problems (anaphora, extraposition, ellipsis, etc.) will be treated and solved.

After the anaphora resolution stage, a new Slot Structure (SS') is obtained. In this new structure the correct antecedent (chosen from the possible candidates) for each anaphoric expression will be stored together with its morphological and semantic information. The new structure SS' will be the input for the last stage of the Analysis module in which the interlingua representation will be obtained.

2.1 Interlingua Representation in AGIR

As said before, this stage takes the SS of the sentence after applying the anaphora resolution module as input. SUPAR generates one SS for each sentence from the whole text and it solves intrasentential and intersentential anaphora. Then, AGIR generates the interlingua representation of the whole text. This is the main difference between AGIR and the rest of MT systems that carry out a processing of the input text sentence by sentence. The interlingua representation will allow the correct generation of the intrasentential and intersentential pronominal anaphora into the target language. Moreover, AGIR allows the identification of coreference chains of the text and their subsequent generation into the target language.

The interlingua representation of the input text is based on the clause as main unit of this representation. Once the text has been split into clauses, AGIR uses a complex feature structure for each clause. It is composed of semantic roles and features extracted from the SS of the clause. Semantic roles that have been used in this approach are the following: ACTION, AGENT, THEME and MODIFIER that correspond to verb, subject, object and prepositional phrases of the clause respectively. The notation we have used is based on the representation used in KANT interlingua.

Once the semantic roles have been identified, the interlingua representation will store the clauses with their features, the different entities that have appeared in the text and the relations between them (such as anaphoric relations). This representation will be the input for the generation module. More details about the interlingua representation in AGIR have been presented in [12,10].

3 AGIR's Generation Module

The Generation module takes the interlingua representation of the source text as input. The output of this module is the target text, that is, the representation of the source text's meaning with words of the target language. In this paper we are only describing the generation of third person personal pronouns into the target language, so we have only focused on the differences between the Spanish and English languages in the generation of the pronoun. These differences are what we have named discrepancies (a detailed study of Spanish-English-Spanish discrepancies is shown in [12,13,10]).

3.1 Syntactic Generation

Elliptical Zero-Subject Constructions (Zero-Pronouns). The Spanish language allows to omit the pronominal subject of the sentences. These omitted pronouns are usually named zero pronouns. While in other languages, zero pronouns may appear in either the subject's or the object's grammatical position, (e.g. Japanese), in Spanish texts, zero pronouns only appear in the position of the subject. In [11,5] the processing of Spanish zero pronouns in AGIR is presented. Basically, in order to generate Spanish zero pronouns into English, they must first be located in the text (ellipsis detection), and then resolved (anaphora resolution). At the ellipsis detection stage, information about the zero pronoun (e.g. person, gender, and number) must first be obtained from the verb of the clause and then used to identify the antecedent of the pronoun (resolution stage).

Pleonastic Pronouns. Sometimes pronouns can be used in a non-referential construction, that is, appear due to some requirement in the grammar of the language. These pronouns are usually named pleonastic. In AGIR, the pleonastic use of pronoun *it* has been detected before the anaphora resolution stage and thereby will not be resolved. These pronouns will appear marked like pleonastics in the interlingua representation, they will not have antecedent and they will not be generated into Spanish. In order to detect pleonastic *it* pronouns in AGIR, a set of rules, based on pattern recognition, that allows the identification of this kind of pronouns is constructed. These rules are based on the study developed by other authors [7,9] that faced with this problem in a similar way.

3.2 Morphological Generation

Number Discrepancies. This problem is generated by the discrepancy between words of different languages that express the same concept. These words can be referred to a singular pronoun in the source language and to a plural pronoun in the target language. In order to take into account number discrepancies in the generation of the pronoun into English or Spanish a set of morphological (number) rules is constructed. The left-hand side of the number rule contains the interlingua representation of the pronoun and the right-hand side contains the pronoun in the target language.

Gender Discrepancies. Gender discrepancies came from the existing morphological differences between different languages. For instance, English has less morphological information than Spanish. The English plural personal pronoun *they* can be translated into the Spanish pronouns *ellos* (masculine) or *ellas* (feminine), the singular personal pronoun *it* can be translated into *él/éste* (masculine) or *ella/ésta* (feminine), etc. In order to take into account gender discrepancies in the generation of the pronoun into English or Spanish a set of morphological (gender) rules is constructed.

4 Evaluation of Generation Module

In this section the Generation module of AGIR has been evaluated. To do so, two experiments have been accomplished: (a) in the first one, the generation of English pronouns into Spanish has been evaluated; (b) in the second one, the generation of Spanish pronouns into English has been evaluated.

As said before, the generation module takes the interlingua representation as input. In this representation, pleonastic *it* pronouns have been detected (with a **P**recision[2] of 88.75%), Spanish zero pronouns have been detected (89.20% **P**) and resolved (81.38% **P**), and anaphoric third person personal pronouns have been resolved in English and Spanish (80.25% **P** and 82.19% **P** respectively).

Once the interlingua representation has been obtained, the method proposed for pronominal anaphora generation into the target language is based on the treatment of number and gender discrepancies.

4.1 Pronominal Anaphora Generation into Spanish

In this experiment the generation of English third person personal pronouns into the Spanish ones has been evaluated.

We have tested the method on both literary and manual texts. In the first instance, we used a portion of the SemCor collection (presented in [6]) that contains a set of 11 documents (23,788 words) where all content words are annotated with the most appropriate WordNet sense. SemCor corpus contains literary texts about different topics (laws, sports, religion, nature, etc.) and by different authors. In the second instance, the method was tested on a portion of MTI[3] corpus that contains 7 documents (101,843 words). MTI corpus contains Computer Science manuals about different topics (commercial programs, word processing, devices, etc.).

We randomly selected a subset of the SemCor corpus (three documents –6,473 words–) and another subset of the MTI corpus (two documents –24,264 words–) as training corpus. The training corpus was used for improving the number and gender rules. The remaining fragments of the corpus were reserved for test data.

To apply the number and gender rules it is necessary to know the semantic type and the grammatical gender of the anaphor's antecedent. In the SemCor corpus the WordNet sense has been used to identify the antecedent's semantic type. In the MTI corpus, due to the lack of semantic information, a set of heuristics has been used to determine the antecedent's semantic type.

[2] By **P**recision we mean the number of pronouns successfully resolved divided by the total number of pronouns resolved in the text. A detailed study of the evaluation of the different tasks carried out in order to obtain the interlingua representation in AGIR can be found in [14].

[3] This corpus has been provided by the Computational Linguistics Research Group of the School of Humanities, Languages and Social Studies –University of Wolverhampton, England–. The corpus is anaphorically annotated indicating the anaphors and their correct antecedents.

With regard to the information about the antecedent's gender, an English-Spanish electronic dictionary has been used due to the POS tag do not provide gender and number information. The dictionary has been incorporated into the system as a database and it provides for each English word: its translation into Spanish, and its gender and number in Spanish.

With this morphological and semantic information the number and gender rules have been applied. We conducted a blind test over the entire test corpus and the obtained results appear in table 1.

Table 1. Generation of pronominal anaphora into Spanish. Evaluation phase

Corpus		Subject	Complement		Correct	Total	P (%)
		AGENT	THEME	MODIF.			
SEMCOR	a02	21	5	1	23	27	85,19
	a11	10	5	0	14	15	93,33
	a13	17	2	3	21	22	95,45
	a14	40	10	1	48	51	94,12
	a15	32	5	4	34	41	82,93
	d02	14	2	3	18	19	94,74
	d03	13	0	1	12	14	85,71
	d04	50	6	9	59	65	90,77
	SEMCOR TOTAL	**197**	**35**	**22**	**229**	**254**	**90,16**
MTI	CDROM	38	24	7	47	69	68,12
	PSW	24	36	2	52	62	83,87
	WINDOWS	16	19	2	30	37	81,08
	SCANWORX	95	87	11	142	193	73,58
	GIMP	66	33	10	82	109	75,23
	MTI TOTAL	**239**	**199**	**32**	**353**	**470**	**75,11**
	TOTAL	**436**	**234**	**54**	**582**	**724**	**80,39**

The evaluation of this task was automatically carried out after the anaphoric annotation of each pronoun. This annotation includes information about the antecedent and the translation into the target language of the anaphor.

Table 1 shows the anaphoric pronouns of each document classified by semantic roles: AGENT, THEME and MODIFIER. The last three columns represent the number of pronouns successfully resolved, the total number of pronouns resolved and the obtained **P**recision, respectively. For instance, the a02 document of the SemCor corpus contains 21 pronouns with semantic role of AGENT, 5 pronouns with semantic role of THEME and 1 pronoun with semantic role of MODIFIER. The **P**recision obtained in this document was of 85.19% (23/27).

Discussion. In the generation of English third person personal pronouns into the Spanish ones an overall **P**recision of 80.39% (582/724) has been obtained. Specifically, 90.16% **P** and 75.11% **P** were obtained in SemCor and MTI corpus respectively. From these results we have extracted the following conclusions:

- In SemCor corpus all the instances of the English pronouns *he*, *she*, *him* and *her* have been correctly generated into Spanish. It is justified by two reasons:
  - The semantic roles of these pronouns have been correctly identified in all the cases.
  - These pronouns contain the necessary grammatical information (gender an number) that allows the correct generation into Spanish, independently of the antecedent proposed as solution by the AGIR system.

  The errors in the generation of pronouns *it*, *they* and *them* have been originated by different causes:
  - Mistakes in the anaphora resolution stage, i.e., the antecedent proposed by the system is not the correct one (44.44% of the global mistakes).This causes an incorrect generation into Spanish mainly due to the proposed antecedent and the correct one have different grammatical gender.
  - Mistakes in the identification of the semantic role of the pronouns that cause the application of an incorrect morphological rule (44.44%). These mistakes are mainly originated by an incorrect process of clause splitting.
  - Mistakes originated by the English-Spanish dictionary (11.12%). Two circumstances can occur: (a) the word does not appear in the dictionary; and (b) the word's gender in the dictionary is different to the real word's gender due to the word has different meanings.
- In MTI corpus, nearly all the pronouns are instances of the pronouns *it*, *they* and *them* (96.25% of the total pronouns). The errors in the generation of these pronouns are originated by the same causes than in SemCor corpus but with different percentages:
  - Mistakes in the anaphora resolution stage (22.86% of the mistakes).
  - Mistakes in the identification of the pronouns' semantic role (62.86%).
  - Mistakes originated by the English-Spanish dictionary (14.28%).

4.2 Pronominal Anaphora Generation into English

In this experiment the generation of Spanish third person personal pronouns (including zero pronouns) into the English ones has been evaluated.

We have tested the method on literary texts. We used a portion of the Lexesp[4] corpus that contains a set of 31 documents (38,999 words). Lexesp corpus contains literary texts about different topics (politics, sports, etc.) from different genres and by different authors.

[4] The Lexesp corpus belongs to the project of the same name carried out by the Psychology Department of the University of Oviedo and developed by the Computational Linguistics Group of the University of Barcelona, with the collaboration of the Language Processing Group of the Catalonia University of Technology, Spain.

We randomly selected a subset of the Lexesp corpus (three documents –6,457 words–) as training corpus. The remaining fragments of the corpus were reserved for test data.

To apply the number and gender rules it is necessary to know the semantic type and the grammatical gender of the anaphor's antecedent. In the Lexesp corpus, due to the lack of semantic information, a set of heuristics has been used to determine the antecedent's semantic type. On the other hand, the information about the antecedent's gender is provided by the POS tag of the antecedent's head. We conducted a blind test over the entire test corpus and the obtained results appear in table 2.

Table 2. Generation of pronominal anaphora into English. Evaluation phase

Corpus		Subject	Complement		Correct	Total	P (%)
		AGENT	THEME	MODIF.			
LEXESP	txt1	19	3	1	21	23	91,30
	txt2	35	7	1	33	43	76,74
	txt3	21	4	1	19	26	73,08
	txt4	13	4	1	15	18	83,33
	txt5	13	4	1	14	18	77,78
	txt6	17	1	0	16	18	88,89
	txt7	22	3	4	28	29	96,55
	txt8	10	0	0	9	10	90
	txt9	9	3	1	8	13	61,54
	txt10	17	2	1	19	20	95
	txt11	7	0	1	7	8	87,5
	txt12	25	4	0	29	29	100
	txt13	16	0	0	12	16	75
	txt14	11	0	0	10	11	90,91
	txt15	16	3	5	18	24	75
	txt16	11	1	2	13	14	92,86
	txt17	14	1	0	11	15	73,33
	txt18	9	4	0	10	13	76,92
	txt19	7	0	1	7	8	87,5
	txt20	17	3	1	13	21	61,90
	txt21	4	2	0	6	6	100
	txt22	12	1	2	15	15	100
	txt23	15	4	2	19	21	90,48
	txt24	21	7	2	25	30	83,33
	txt25	92	11	5	100	108	92,59
	txt26	132	16	11	129	159	81,13
	txt27	24	6	1	27	31	87,10
	txt28	21	5	2	24	28	85,71
	TOTAL	**630**	**99**	**46**	**657**	**775**	**84,77**

Discussion. In the generation of Spanish third person personal pronouns into the English ones an overall **P**recision of 84.77% (657/775) has been obtained. From these results we have extracted the following conclusions:

- All the instances of the Spanish plural pronouns (*ellos*, *ellas*, *les*, *los*, *las* and zero pronouns in plural) have been correctly generated into English. It is justified by two reasons:
  - The semantic roles of these pronouns have been correctly identified in all the cases.
  - The equivalent English pronouns (*they* and *them*) lack gender information, i.e., are valid for masculine and feminine, then the antecedent's gender does not influence the generation of these pronouns.
- The errors occurred in the generation of the Spanish singular pronouns (*él*, *ella*, *le*, *lo*, *la* and zero pronouns in singular). They have been originated by different causes:
  - Mistakes in the anaphora resolution stage (79.66% of the global mistakes). This causes an incorrect generation into Spanish mainly due to the proposed antecedent and the correct one have different grammatical gender. Sometimes, both have the same gender but they have different semantic type.
  - Mistakes in the application of the heuristic used to identify the antecedent's semantic type (20.34%). This fact involves the application of an incorrect morphological rule.

Conclusion

In this paper a complete MT approach to solve and generate pronominal anaphora in the Spanish and English languages is presented. The approach works on unrestricted texts to which partial parsing techniques have been applied. After the parsing and solving pronominal anaphora, an interlingua representation (based on semantic roles and features) of the whole text is obtained. The representation of the whole text is one of the main advantages of our system due to several problems, that are hardly solved by the majority of MT systems, can be treated and solved. These problems are the generation of intersentential anaphora, the detection of coreference chains and the generation of Spanish zero-pronouns into English. The generation of English and Spanish personal pronouns (including zero pronouns) into the target language has been evaluated obtaining a Precision of 80.39% and 84.77% respectively.

References

1. L. Appelo and J. Landsbergen. The machine translation project Rosetta. In T.C. Gerhardt, editor, *I. International Conference on the State of the Art in Machine Translation in America, Asia and Europe: Proceedings of IAI-MT86, IAI/EUROTRA-D*, pages 34–51, Saarbrucken (Germany), 1986.

2. A. Berger, P. Brown, S.D. Pietra, V.D. Pietra, J. Gillett, J. Lafferty, R.L. Mercer, H. Printz, and L. Ures. The Candide system for Machine Translation. In *Proceedings of the ARPA Workshop on Speech and Natural Language*, pages 157–163, Morgan Kaufman Publishers, 1994.
3. J. Chandioux. MÉTÉO: un système opérationnel pour la traduction automatique des bulletins météreologiques destinés au grand public. *META*, 21:127–133, 1976.
4. A. Ferrández, M. Palomar, and L. Moreno. An empirical approach to Spanish anaphora resolution. *Machine Translation*, 14(3/4):191–216, 1999.
5. A. Ferrández and J. Peral. A computational approach to zero-pronouns in Spanish. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'2000)*, pages 166–172, Hong Kong (China), 2000.
6. S. Landes, C. Leacock, and R. Tengi. Building semantic concordances. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 199–216. MIT Press, Cambridge, Mass, 1998.
7. S. Lappin and H.J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.
8. T. Mitamura, E. Nyberg, and J. Carbonell. An efficient interlingua translation system for multi-lingual document production. In *Proceedings of Machine Translation Summit III*, Washington, DC (USA), 1991.
9. C.D. Paice and G.D. Husk. Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun "it". *Computer Speech and Language*, 2:109–132, 1987.
10. J. Peral and A. Ferrández. An application of the Interlingua System ISS for Spanish-English pronominal anaphora generation. In *Proceedings of the Third AMTA/SIG-IL Workshop on Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP (ANLP/NAACL'2000)*, pages 42–51, Seattle, Washington (USA), 2000.
11. J. Peral and A. Ferrández. Generation of Spanish zero-pronouns into English. In D.N. Christodoulakis, editor, *Natural Language Processing - NLP'2000*, volume 1835 of *Lecture Notes in Artificial Intelligence*, pages 252–260, Patras (Greece), 2000. Springer-Verlag.
12. J. Peral, M. Palomar, and A. Ferrández. Coreference-oriented Interlingual Slot Structure and Machine Translation. In *Proceedings of the ACL Workshop Coreference and its Applications*, pages 69–76, College Park, Maryland (USA), 1999.
13. J. Peral. Proposal of an English-Spanish interlingual mechanism focused on pronominal anaphora resolution and generation in Machine Translation systems. In *Proceedings of the Student Session of the 11th European Summer School in Logic, Language and Information (ESSLLI'99)*, pages 169–179, Utrecht (Holland), 1999.
14. J. Peral. *Resolución y generación de la anáfora pronominal en español e inglés en un sistema interlingua de Traducción Automática.* PhD thesis, University of Alicante, 2001.
15. A.P.M. Witkam. *Distributed language translation: feasibility study of multilingual facility for videotex information networks.* BSO, Utrecht, 1983.

Using LSA for Pronominal Anaphora Resolution

Beata Klebanov[1] and Peter Wiemer-Hastings[2]

[1] Hebrew University[⋆]
beata@cs.huji.ac.il, http://www.cs.huji.ac.il/~beata
[2] DePaul University
pwh@cti.depaul.edu, http://reed.cs.depaul.edu/peterwh

Abstract. Until now, the contribution of world knowledge to the process of pronominal anaphora resolution has not received a thorough computational investigation, mostly due to the lack of a large scale implemented model of world knowledge. This paper proposes Latent Semantic Analysis (LSA) as such a model: word meaning representation it constructs can be used to rank potential antecedents according to how well they fit in the pronoun's context. The initial results of incorporating LSA into a pronominal anaphora resolution algorithm are encouraging.

The importance of world knowledge in pronominal anaphora resolution has been recognized since the early days of computational approaches to the problem. In his groundbreaking paper that proposes a syntax-based resolution algorithm [4], J. Hobbs suggests that his algorithm could be enhanced by adding "simple selectional restrictions, like dates don't move, large fixed objects don't move, etc." However, he does not propose a way to acquire this knowledge automatically.

Attempts have been made to model selectional restrictions[1] by estimating from a corpus the probability of the appearance of a word w as an object of a verb v [7], [2]. The higher the probability, the better candidate w is for resolving a pronoun in the object position of v. However, statistical methods that rely only on explicit examples of the sought-after cases suffer from the problem of sparse data – not everything that could be the object of a certain verb will actually be encountered in this guise in the training data. For example, it might happen that the word *apple* appeared as an object of *eat* in a certain corpus, whereas *pear* failed to do so. Should we conclude that *pear* is as likely an object of *eat* as any word that did not appear in this position in the corpus? No, we would like the algorithm to know that since almost anything that can be said about apples can also be said about pears, an occurrence of *eat apples* should teach us that *eat pears* is just as possible. More generally, we would like the algorithm to be informed about relationships between concepts, such that we could use direct evidence more effectively.

To tackle this problem, we propose to use Latent Semantic Analysis [1], [6] – a system with a proven ability to handle synonymy and semantic relatedness in general – as a model of world knowledge. Being trained on large amounts

[⋆] The work was carried out when both authors were at the University of Edinburgh

[1] Restrictions that verbs place on their objects, eg. *eat* requires an edible object

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 197–199, 2002.

of plain text segmented into documents[2], LSA keeps track of occurrences of words within documents to construct a high-dimensional space where words that appear in similar contexts are close to each other (cosine is used as the distance metric). Similar patterns of contextual occurrences are very characteristic of closely related words (pear/apple).

The algorithm reported here uses the Penn Treebank Wall Street Journal corpus [8]. LSA was trained on sections 02-60. Sections 00-01 are annotated with anaphoric links, i.e. referring expressions that realize an entity mentioned more than once in a discourse are marked with the same unique reference number designating this entity. These two sections were used for development (the first 43 discourses) and for testing (the remaining 91) of the resolution algorithm.

The task is to find an antecedent for every 3rd person singular non-reflexive pronoun that is marked as object-referential. A candidate antecedent is, roughly (see [5] for the exact definition), every NP that is encountered in the previous sentence within the same discourse or precedes the pronoun in the current sentence. Note that for pronouns whose only referent is ≥ 2 sentences back and for cataphora[3], the correct resolution would not be in the list of candidates. This restriction leads to a recall loss of 4.9% on both development and test data.

For every pronoun, the algorithm collects all the candidate antecedents and excludes those that fail person, number or binding constraints. The baseline algorithm picks at random one of the remaining candidates. Our algorithm uses LSA to assign scores to all the remaining candidates and proposes as the antecedent the one that scored highest. The resolution is correct if the proposed antecedent has the same reference number as the pronoun.

To assign scores, every candidate is LSA-compared to a query, which is the string dominated by the syntactic tree node corresponding to the governing category of the pronoun[4]. Note that for pronouns in verb-dependent positions (subject/object/indirect object), this generalizes the idea of selectional restrictions by using not just the verb itself to make the fitness judgment, but also its other dependents. Indeed, the latter are important, especially for non-restricting verbs, like *give* – almost anything can be given, from homework to medicine. However, if we know that the giver is a doctor, medicine would be a more likely resolution; if the giver is a teacher, homework is probably more fitting. For possessive pronouns, the query consists of the possessed entity[5].

Consider the following short discourse:

Example 1. The new medicine has been released to the market a week ago. The doctor gave it to the boy.

The new medicine, the market, a week are all legal candidates to resolve *it.* For each one of them, we ask LSA to compare the candidate with the query *the*

[2] LSA's context unit, which is set to a paragraph of text in our application

[3] cases where the only referent follows the pronoun in the sentence

[4] In GB, the governing category of a node α is the minimal phrasal domain that contains α and its governor and has a specifier [3]. The needed syntactic structure is retrieved from the corpus markup – see [5] for a detailed discussion

[5] assuming the predicate structure of the verb "to have": *his dog = he has a dog*

doctor gave to the boy. LSA's answer is the cosine between the two vectors in its space. We expect *medicine* to be closer to the query (thus corresponding to a larger cosine value and a higher score) than *the market* or *a week*, since *doctor* is indicative of the context within which *medicine* also tends to appear. Table 1 presents precision figures obtained from running the algorithm on the test data.

Table 1. Performance of the algorithm with LSA vs. baseline

Algorithm	Resolved correctly (out of 1058)
Baseline	33.8%(358)
LSA	42.9%(454)

The results show that LSA knows something that is not contained within the constraints we implemented – number and person agreement and binding (one can view the constraints as possessing relevant knowledge since they manage to rule out some inappropriate antecedents). Thus, using LSA to help resolve anaphora is a promising strategy.

It is yet to be established just how much of what we call "world knowledge" LSA possesses. It also remains to be seen exactly what is the role of this knowledge in the process of resolution: is it redundant or complementary to other kinds of information traditionally used in anaphora resolution algorithms, eg. the knowledge of the grammatical role of the candidate antecedent, or the distance between the pronoun and the antecedent? These are the main directions for future work.

References

1. Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by Latent Semantic Analysis. In *Journal of the Society for Information Science*, **41(6)** (1990) 391–407
2. Ge, N., Hale, J., Charniak, E.: A statistical Approach to Anaphora Resolution. In *Proceedings of the 6th Workshop on Very Large Corpora*, Montreal, Canada (1998) 161–170
3. Haegeman, L.: Introduction to Government and Binding Theory. 2nd edition, Cambridge, Mass: Blackwell Publishers (1994)
4. Hobbs, J.: Resolving Pronoun References. *Lingua* **44** (1977) 311–338
5. Klebanov, B.: Using Latent Semantic Analysis for Pronominal Anaphora Resolution. *MSc Dissertation, Division of Informatics, University of Edinburgh* (2001). Also available from http://www.cs.huji.ac.il/~beata
6. Landauer, T., Foltz, P., Laham, D.: Introduction to Latent Semantic Analysis. In *Discourse Processes* **25** (1998) 259–284
7. Lappin, S., and Leass, H.: An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics* **20(4)** (1994) 535–561
8. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* **19(2)** (1993) 313–330

The Spanish Auxiliary Verb System in HPSG

Ivan Meza and Luis Pineda

Dept. Computer Science, IIMAS, UNAM
Cicuito interior s/n, Ciudad Universitaria, Coyoacán, 04510, México D.F.
ivanvladimir@cic1.iimas.unam.mx
luis@leibniz.iimas.unam.mx

Abstract. This paper presents an analysis of the Spanish auxiliary verbs from a syntactic point of view instead a semantic analysis that is proposed by the traditional Spanish grammars. The syntactic context of the Spanish auxiliary verbs is clarified with the definition of five properties, which allow us to determinate if a verb behave as an auxiliary verb or not. The subject raising verb type defined in the formalism HPSG is used in order to capture the behavior of Spanish auxiliary verbs. We conclude with the analysis of a typical auxiliary verb, *poder*, as a case of study.

1 Introduction

The definition and implementation of a robust grammar for Spanish requires a solid analysis of the Spansih auxiliary verb system. Although the verb system for English has been widely studied and implemented in computational grammars since Chomsky´s original analysis in Syntactic Structures, where the crucial role of the study of auxiliary verbs for the development of English grammar is pointed out [1], to our knowledge, no such rigorous computational analysis for the Spanish auxiliaries is available to this date. Even traditional Spanish grammars lack a definite analysis of the grammatical phenomena, as several criteria and classifications, not always consistent, can be found in the literature.

In particular, the need for a useful implementation of this system comes from the DIME project (Diálogos Multimodales Inteligentes en Español) currently developed at IIMAS, UNAM, which has as its main objective the construction of a conversational multimodal agent with spoken input and output facilities in Spanish for helping the interactive solution of simple design task [2]. In particular, the DIME prototype will be able to assist human-users in simple kitchen design tasks. One of the main objectives of this project is the development of a Spanish grammar and parser able to cope with the language of a corpus of Spanish dialogs in this domain that was compiled within the context of this project [3]. For the development of the grammar the HPSG grammatical formalism [4] with its associated environment development was adopted [5].

The need for a full analysis of auxiliary verbs emerged immediately when the first sentences of the DIME corpus were analyzed. Sentence (1) is a typical construction in the corpus:

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 200–209, 2002.

(1) ¿Me puedes mostrar el catálogo de muebles?
Can you show me the catalog of furniture?

This sentence shows a number of syntactic phenomena that are characteristic of Spanish syntax, which are alien to English: omitted subject, clitic constructions and the "periphrastic conjugation". Also, the sentence is ambiguous because the verb *poder* can be interpreted as signaling ability or possibility: in the latter case it is an auxiliary verb but in the former it is not, as will be explained below in this paper. In addition, (1) is an interrogative form in which the subject-verb inversion of English questions does not take place; furthermore, the word-order for the declarative, interrogative and imperative forms of this kind of sentences can be the same. Each of these syntactic phenomena needs a detailed analysis; however, central to all of them is the analysis of the complex verbal construction and, in particular, of the auxiliary verb construction, which is the skeleton upon which all other phenomena are supported.

2 Spanish Auxiliary Verbs

Intuitively, an auxiliary verb in Spanish is a verb that has lost its original lexical meaning and has acquired a grammatical function or meaning in specific syntactic contexts. Consider the examples in (2) taken from Gili Gaya [6 pp. 108]:

(2) a. Tener que escribir
Have to write
b. Estar escribiendo
To be writing
c. Ir a escribir
Going to write

In all three, there is a reference to the action of writing, but the initial verbs marks a specific mode in which the writing action should be interpreted. In (2.a), *tener* in the context of *tener que* has changed its original meaning of possession to obligation, and *tener que escribir* refers to the obligation of writing. In (2.b), *estar* has lost its meaning of location and *estar escribiendo* alludes to the duration of the act of writing, and in (2.c), the verb *ir* has lost its meaning of physical transfer and marks the incoative notion of the act of writing, and the whole construction means to start to write.

The grammaticalization of the verbs is not a phenomenon particular to Spanish, as can be seen from the translations, but it does give rise to some kind of ambiguities that are very peculiar to Spanish. Auxiliaries in English are fully grammaticallized: when a verb becomes an auxiliary, it preserves this function most of the time; however, in Spanish this is not the case, as shown in (3). In (3.a), *debo* preserves its original meaning of obligation of paying a debt, but to express the same idea in English requires the use of the verb *owe* which is not an auxiliary. On the other hand, in the verbal complex *debo ir* in (3.b), *debo* is an auxiliary marking a general kind of obligation, and its expression in English requires the auxiliary verb *must*. In this regard, English is better behaved than Spanish.

(3) a. Debo cien pesos
I owe one hundred pesos
b. Debo ir al banco
I must go to the bank

Verbal complex constructions are known as periphrastic conjugation (*conjugación perifrásica*) in which the first verb is conjugated (*forma personal*) and the rest of the verbs in the complex are in a non-finite form (*formas no personales*) which are the infinitive, the gerund and the participle, as can be seen in Table 1.

Table 1. Different non-finite forms taken by auxiliars

a.	Iba a decir *He/she was going to tell*	Infinitive
b.	Estaba comiendo *He/she/it was eating*	Gerund
c.	He caminado *I have walked*	Participle

In Spanish, all auxiliaries appear in periphrastic constructions but there are periphrastic constructions in which the conjugated verbs are not auxiliaries; for this reason it is not trivial to state when a verb has such a function. Even traditional grammars of Spanish have different criteria. Gili Gaya [6], for instance, adopts the strongest position and defines an auxiliary as a verb that has lost his meaning and has taken a new one, as illustrated in (3). However, most Spanish grammars adopt a less restricted position, and propose a hierarchy of auxiliaries: while *ser*, *estar* and *haber* are classified as full auxiliaries, all modal verbs function as semi-auxiliaries [7]. Nevertheless, the indecision prevails, and the full set of auxiliares is not well-defined.

To clarify the distinction, it can be noticed that changes of meaning are accompanied with changes in syntactic behavior (e.g., in (3.b) the verb *debo* is no longer a transitive). Accordingly, we adopt the view that to understand the behavior of auxiliaries not only semantic criteria but also syntactic properties must be taken into account.

2.1 Behavior of Periphrastic Constructions

For the study of periphrastic constructions we focus on the behavior of agents and patients. The agent is individual who executes the action named by the verb and the patient is the one how receives such an action; in general, the agent corresponds to syntactic subject and the patient to the direct object in active sentences. In Spanish periphrastic constructions agents and verbs are related through the following syntactic properties:

1. The conjugation of the auxiliary verb contains the syntactic subject, which is normally omitted; however, auxiliaries do not require agents, as the main semantic import of the periphrasis is marked by the verb in a non-finite form. For this reason, the syntactic subject marked by the conjugation helps to identify the agent of the non-finite verb. In (4), for instance, the verb *voy* (1st-sing) marking the incoative action helps to identify that the agent of *comer* is me (*yo*).

   (4) (Yo) voy a comer
   I am going to eat

2. It is possible to construct periphrasis with verbs that do not need an agent. These verbs are known as impersonal (*impersonales*) and they represent agentless actions like raining and snowing. In addition, impersonal constructions can also be formed dropping the agent, either because it is not known or just because it is not important. In (5.a), *va* (3rd-sing) marks the subject of the sentence, but *llover* requires no agent, and the information provided by the conjugation is simply not used. In (5.b), *van* (3rd-pl) marks that someone (perhaps more than one) are going to knock the door, but we don't know how is it. Again, the information provided by the subject is not used, because the agent needs not to be determined.

   (5) a. Va a llover
   It is going to rain
   b. Van a tocar la puerta
   Someone is going to knock the door

3. It is not possible construct an interrogative sentence asking for the direct object of the periphrases using the auxiliary verb only. For instance, in *Voy a estudiar matemáticas* the direct object of whole periphrasis (i.e., *matemáticas*) is an argument of *estudiar*; this is the case because the auxiliary verb *voy a* has no semantic import, but only that the action will take place in the future. Question (5.b) ask for the direct object of *estudiar* and is well-formed, but (5.a) is not well-formed because it asks for the direct object of the auxiliary, which does not exist.

   (6) a. *¿Qué (tú) vas a?
   **What are you going to?*
   b. ¿Qué (tú) vas a estudiar?
   What are you going to study?

4. It is not possible construct an interrogative sentence asking for the action named by the non-finite verb using the auxiliary verb only; to form this kind of questions a wildcard verb is required. The action of the sentence *Voy a estudiar* can be inquired with question (6.b) where the wild-card verb *hacer* substitutes the action *estudiar*; however, question (6.a) is not well formed.

   (7) a. *¿Qué vas a?
   **What are you going to?*
   b. ¿Qué vas a hacer?
   **What are you going to do?*

5. It is not possible construct periphrasis with the auxiliary in passive voice; if the periphrasis in *voy a entregar la carta* is expressed as a passive construction, the participle needs to be non-finite verb.

   (8) *La carta es ida a entregar por mi
   **The letter is gone deliver by me*
   (9) La carta va a ser entregada por mi
   The letter is going to be delivered by me

These five properties define the auxiliary verbs in Spanish. This analysis contrasts with most traditional accounts of Spanish grammar where the sole presence of a periphrastic construction is normally taken to signal auxiliary verbs. As there are periphrastic construction in which these five properties do not hold, our analysis permits to distinguish finite verbs occurring in periphrastic construction which do function as auxiliaries, from verbs also occurring in this kind of construction which, nevertheless, are not auxiliaries. Sentences (10) and (11) are examples of this latter phenomenon.

(10) Quiero comer una manzana
I want to eat an apple
(11) Tengo estudiada la materia
I have studied the subject

Gili Gaya classifies auxiliaries according to the non-finite verb occurring in the periphrasis as follows [6 pp. 100]:

- With infinitives: *ir a, pasar a, echar a, venir a, volver a, haber de, haber que, tener que, deber de, llegar a, acabar de* and *alcanzar a*;
- With gerunds: *estar, ir, venir, seguir* and *andar*;
- With participles: *llevar, tener, traer, quedar, dejar, estar* and *ser.*

The auxiliary verb *haber* used in all tenses of the Spanish conjugation is taken as fix desinence and its given an independent treatment.

We have tested all five properties in the set and only sixteen passed the test. In particular *echar a*, *pasar a* and *haber que*, and also all five that are followed by participle, do not count as auxiliaries in our criteria. On the other hand, the verb *haber* in the conjugation of composite tenses do counts as a normal auxiliary verb.

As all five properties also occur in English, our analysis also shows that despite traditional perception, the auxiliary verbal systems of both of these languages are quite similar. However, English auxiliaries have, in addition, the so-called NICE properties (negation, inversion, contraction and ellipsis) [4 pp. 302] making their behavior more systematic than the corresponding Spanish constructions. On the other hand, unlike auxiliary verbs in English, which seem to be fully grammaticalized, the Spanish verbs that function as auxiliaries, can preserve its original function, even in periphrastic constructions, producing a number of ambiguities that need to be addressed. Next, we turn to the formal analysis of periphrasis in Spanish both when auxiliaries are involved, and also when they are not.

3 Auxiliary Verbs in HPSG

In HPSG all lexemes are related in a lattice of types [4]. In particular, verb-lexemes (*verb-lxm*) have the so-called subject raising verb (*srv-lxm*) and subject control verbs (*scv-lxm*) as subtypes. This distinction can be found original in Chomsky's Extended Transformational Grammar [4 pp. 280], and it is widely used in HPSG and related grammatical formalisms. Auxiliary verbs (*auxv-lxm*) are also subtypes of the type *srv-lxm*. In particular, objects of type *auxv-lxm* are objects of type *srv-lxm* that also have the NICE properties. Here, we will claim that auxiliary verbs in Spanish are *srv-lxm*. But in addition, we will claim that the same verbs can also occur as *scv-lxm* and even as transitive or intransitive verbs. In particular, the verb *poder* occurs at least in three different types, with different syntactic properties, and different semantic import.

3.1 Subject-Raising and Subject-Control Verbs

The type *srv-lxm* is defined in HPSG as a attribute-value matrix (AVM) in (12), where the symbol *srv-lxm* is the type identifier, and the argument structure (ARG-ST) has two arguments, where the first argument must also be the specifier of the second. The principal characteristic of this AVM is that nothing is specified about the agent of sentences in which the head of a verbal phrase is of this type (*srv-lxm*).

(12)

$$\left[\begin{array}{ll} \textit{srv-lxm} & \\ \textit{ARG-ST} & \left\langle \#1, \left[\begin{array}{ll} \textit{phrase} & \\ \textit{SPR} & \langle \#1 \rangle \end{array}\right] \right\rangle \end{array}\right]$$

In the phrase *poder mostrar*, for instance, *poder* is a verb of type *srv-lxm* and its second argument has *mostrar* as its head. Consequently, according to (12), the first argument of *poder*, the subject of the sentence, is also the specifier (the first argument) of *mostrar*. As there are no further constraints in this specification, the subject of the sentence is the agent of the action denoted by *mostrar* (the one who is doing the showing), regardless whether the agent is mentioned explictly, or it is absent, as it is the case in impersonal constructions. Here, the verb *poder* is an auxiliary verb marking the possibility of the act of showing.

Next, we consider the subject-control verbs. The type *srv-lxm* is defined with the following AVM:

(13)

$$\left[\begin{array}{ll} \textit{scv-lxm} & \\ \textit{ARG-ST} & \left\langle NP_i, \left[\begin{array}{ll} \textit{phrase} & \\ \textit{FORM} & \text{inf} \\ \textit{SPR} & \langle NP_i \rangle \end{array}\right] \right\rangle \end{array}\right]$$

This structure is similar to the definition of *srv-lxm*, but it forces a restriction in the type of the specifier. If *poder*, in *poder mostrar*, is of this latter type, its first argument is marked as a noun-phrase that is also the first argument of *mostrar*. In this case, the subject of *poder* becomes the agent of *mostrar*, and this agent must necessarily be there. The meaning of *poder* in this interpretation is that an agent has the capability of showing or is able to show something. English prefers the form *be able* for this function.

3.2 Definition of Auxiliary Verbs in Spanish

Unlike English where auxiliary verbs have the NICE properties, and the type *auxv-lxm* can be defined as a subtype of *srv-lxm*, in Spanish there no such distinction and we identify the type of auxiliaries with the *srv-lxm* type, as follows:

Definition (1): auxiliary verbs in Spanish are of type *srv-lxm*.

4 *Poder*

Next, we present an analysis of the verb *poder* in the three different contexts as shown in (14.a) to (14.c), as follows:

(14) a. Puedes con las matemáticas
You are capable in mathematics
b. Puedes mostrar el catálogo
You are able to show the catalog
c. Puedes mostrar el catálogo
It is possible that you show the catalog

In the sentence (14.a) the verb *poder* means ability; this is perhaps its original lexical meaning. In this context, *poder* is a prepositional intransitive verb (*piv-lxm*). In (14.b) it has the meaning of capability, and in (14.c) functions as an auxiliary and signals the possibility of showing. Next, we show the corresponding lexical entries in HPSG in (15).

(15) a. *poder* as a *piv-lxm*

$$\left\langle poder,\ \begin{bmatrix} piv-lxm & \\ ARG-ST & \left\langle [\,],\ \begin{bmatrix} HEAD & \begin{bmatrix} Phrase & \\ FORM & \text{Prep} \\ P-OBJ & [NP_j] \end{bmatrix} \end{bmatrix}_k \right\rangle \\ SEM & \begin{bmatrix} INDEX & i \\ MODE & prop \\ REST & \left\langle \begin{bmatrix} RELN & be\_able \\ SIT & i \\ CAPABLE & j \\ CAPACITY & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

b. *Poder* as a *scv-lxm*

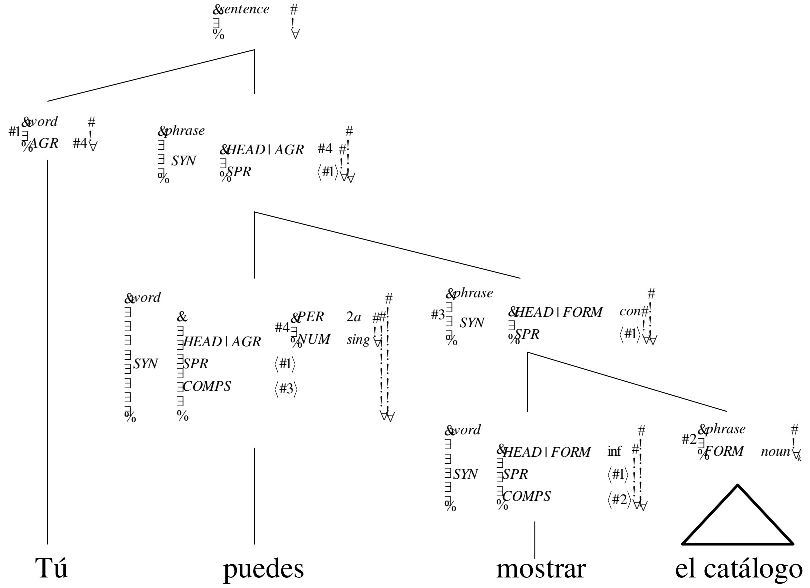
$$\left\langle poder,\ \begin{bmatrix} scv-lxm & \\ ARG-ST & \left\langle [\,]_j, [FORM \quad \text{inf}\,]_k \right\rangle \\ SEM & \begin{bmatrix} INDEX & i \\ MODE & prop \\ REST & \left\langle \begin{bmatrix} RELN & be\_able \\ SIT & i \\ CAPABLE & j \\ CAPACITY & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

c. *Poder* as a *srv-lxm*

$$\left\langle poder,\ \begin{bmatrix} srv-lxm & \\ ARG-ST & \left\langle [\,], [FORM \quad \text{inf}\,]_j \right\rangle \\ SEM & \begin{bmatrix} INDEX & i \\ MODE & prop \\ REST & \begin{bmatrix} RELN & possibility \\ SIT & i \\ POSSIBLE & j \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

As can be seen in the argument structure of (15.a) and (15.b), the difference between these lexical entries is that while the intransitive verb takes a prepositional phrase as a complement, the subject control verb takes a non-finite form as its complement. On the other hand, the semantics in both of these entries is the same, and states the relation between an individual and something that he or she is capable of. (15.c) differs in the syntax as was explained above. The semantics is also different as the relation marked is one of possibility, and the agent needs not to be present. Next, we show the syntactic structure of (15.c).

(16)


sentence
#1 word AGR #4
phrase SYN HEAD | AGR #4 SPR ⟨#1⟩
word SYN HEAD | AGR #4 PER 2a NUM sing SPR ⟨#1⟩ COMPS ⟨#3⟩
#3 phrase SYN HEAD | FORM con SPR ⟨#1⟩
word SYN HEAD | FORM inf SPR ⟨#1⟩ COMPS ⟨#2⟩
#2 phrase FORM noun
Tú
puedes
mostrar
el catálogo

This example shows how the first structure that is formed is the verbal phrase *mostrar el catalogo,* when the verb *mostrar* takes his complements in the application of the Head Complement Rule. After, this structure unifies with the first argument of the COMPS of the verb *poder*, the Head Complement Rule is applied again and the verbal phrase *puedes mostrar el catálogo* is obtained. Finally, the SPR of *mostrar* unifies with the SPR of the verb *poder*; this last phrase can take his subject *tú*(2nd-sing), which helps to define the agent of the second verb. The syntactic structure of (14.a) and (14.b) is obtained by a similar process. The final semantics for the sentences in (14.c) are:

(17) *Poder* meaning possibility:

$$\begin{bmatrix} \mathit{INDEX} & \mathit{i} \\ \mathit{MODE} & \mathit{prop} \\ \mathit{RESRT} & \left\langle \begin{matrix} \begin{bmatrix} \mathit{RELN} & \mathit{possibilit\ y} \\ \mathit{SIT} & \mathit{i} \\ \mathit{POSSIBLE} & \mathit{k} \end{bmatrix} & \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & \mathit{u} \\ \mathit{NAME} & \mathit{hearer} \\ \mathit{NAMED} & \mathit{j} \end{bmatrix} \\ \begin{bmatrix} \mathit{RELN} & \mathit{show} \\ \mathit{SIT} & \mathit{k} \\ \mathit{SHOWER} & \mathit{j} \\ \mathit{OBJECT} & \mathit{l} \end{bmatrix} & \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & \mathit{v} \\ \mathit{NAME} & \mathit{catalog} \\ \mathit{NAMED} & \mathit{l} \end{bmatrix} \end{matrix} \right\rangle \end{bmatrix}$$

This semantic refers to the situation *i* that signals the possibility of an another situation; here the possible situation is mark by index *k* which corresponds to the relation of showing, which needs two arguments: one who shows that is the agent of showing act and the object showed; the shower corresponds to the hearer because the syntactic subject of the sentence was *tú* (2nd-sing), and the object is the catalog. Notice that the value that corresponds to the agent is only used by the relation of show and it does not apear in the relation of possibility.

Next, we show the final semantic of *poder* for (14.a) and (14.c):

(18) a. *Poder* meaning ability

$$\begin{bmatrix} \mathit{INDEX} & i \\ \mathit{MODE} & \mathit{prop} \\ \mathit{RESRT} & \left\langle \begin{matrix} \begin{bmatrix} \mathit{RELN} & \mathit{be\_able} \\ \mathit{SIT} & i \\ \mathit{CAPABLE} & j \\ \mathit{CAPACITY} & k \end{bmatrix} & \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & u \\ \mathit{NAME} & \mathit{math} \\ \mathit{NAMED} & k \end{bmatrix} \\ \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & v \\ \mathit{NAME} & \mathit{listener} \\ \mathit{NAMED} & j \end{bmatrix} & \end{matrix} \right\rangle \end{bmatrix}$$

b. *Poder* meaning capability

$$\begin{bmatrix} \mathit{INDEX} & i \\ \mathit{MODE} & \mathit{prop} \\ \mathit{RESRT} & \left\langle \begin{matrix} \begin{bmatrix} \mathit{RELN} & \mathit{be\_able} \\ \mathit{SIT} & i \\ \mathit{CAPABLE} & j \\ \mathit{CAPACITY} & k \end{bmatrix} & \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & u \\ \mathit{NAME} & \mathit{hearer} \\ \mathit{NAMED} & j \end{bmatrix} \\ \begin{bmatrix} \mathit{RELN} & \mathit{show} \\ \mathit{SIT} & k \\ \mathit{SHOWER} & j \\ \mathit{OBJECT} & l \end{bmatrix} & \begin{bmatrix} \mathit{RELN} & \mathit{name} \\ \mathit{SIT} & v \\ \mathit{NAME} & \mathit{catalog} \\ \mathit{NAMED} & l \end{bmatrix} \end{matrix} \right\rangle \end{bmatrix}$$

Structure (18.a) and (18.b) are similar as the firts argument in their restriction list is the same, but while (18.a) states that the agent *j* has the capability *k* (mathematics), (18.b) states that the agent who is able is also the agent of the showing action. In adition, (18.a) has a restiction which names the patient of the showing action (i.e., the catalog). On the other hand, (18.b) is also similar to (17), as both of them have the same relation in their corresponding restriction list, but while (18.b) has an agent for the capability (the index *j*), the restiction of posibility has not agent; it just marks a situation in which it is possible that the hearer performs a showing action.

5. Conclusions

In this paper a computational analysis of Spanish auxiliary verbs has been presented. The analysis was centered on the syntactic properties of auxiliaries, in opposition to traditional analysis that are mainly semantically oriented. Five syntactic properties of auxiliaries were identified which allowed us to separate the notions of periphrastic conjugation and auxiliary verbs that are usually taken together in traditional analysis. In particular, auxiliaries in Spanish were identified as subject raising verbs. Our analysis allow us also to identify a number of ambiguities arise as a consequence of the use of the same verbs in different context, whether they function as auxiliaries or not in such context. We are able to identify a new set of auxiliaries that can be formally distinguished, and is also intuitively appropriated. In particular, a set of sixteen verbs that function as auxiliaries has been identified, as follows: *poder, ir a, venir a, volver a, haber de, tener que, deber de, llegar a, acabar de* and *alcanzar a* which take an infinitive complement; *estar, ir, venir, seguir* and *andar* which take a gerund as a complement, and *haber* which accepts past participles as complements. This verbs have been implemented in the LKB develop environment for HPSG, and they will be used in the interpretation of natural languages Spanish sentences on the prototype of the DIME project.

Acknowledgments

Discussions and feedback were very much appreciated, specially from the following people: Erick Schwarz, Antonio Massé, Esmeralda Uraga, Miguel Salas, Luis Villaseñor and Celina López, and also from James Allen, Eduard Hovy and the NL-group at the University of Rochester. The authors also gratefully acknowledge the support of Conacyt's grant 499316-5-C092A.

References

1. Chomsky, Noam: Syntactic Structures. Mouton & Co. 'S-Gravenhage. (1962)
2. Pineda, L.A., Massé J.A., Uraga E., y Villaseñor, L., Salas, M., Schwarz, E. y Meza, I.: El Proyecto DIME. In Procedings of Second International Workshop on Spanish Language Processing and Language Technologies. Jaén, Spain. pages 41-45 (2001)
3. Villaseñor, L., Massé J.A., Pineda, L.A.: The DIME Corpus. In Memorias del *3er. Encuentro Internacional de Ciencias de la Computacion*. Aguascalientes, México (2001) 591-600
4. Sag, I. y Wasow, T.: Syntactic Theory A Formal Introduction. CSLI Publications. Stanford, California (1999)
5. Copestake, A.: The LKB System. Technical report. Stanford University (2001) http://www-csli.stanford.edu/~acc/lkb.html
6. Gili Gaya, S.: Curso superior de sintaxis española. Decimoquinta edición. Vox. Barcelona. (1991) 102-119
7. Alonso, A. y Henríquez, P.: Gramática castellana, segundo curso. 22ª. Edición. Buenos Aires. (1967)

Surface Syntactic Relations in Spanish[*]

Igor A. Bolshakov

Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Mexico City, Mexico
igor@cic.ipn.mx

Abstract. A preliminary inventory of Surface Syntactic Relations (SSynRel) in Spanish is proposed. SSynRels link governor words with their dependents at surface syntactic level of representation defined in Meaning ⇔ Text Model by I. Mel'čuk. Some syntactic peculiarities of Spanish as compared with other languages are revealed. Each SSynRel is supplied with examples and a short discussion. An example of surface syntactic representation of a complete sentence is also given.

1 Introduction

At least three recent decades, a sort of competition was held between two approaches to the description of syntactic relations between words in sentences, namely, between constituency and dependency approaches.

The constituency approach made by N. Chomsky had later branched to numerous descendant theories [11]. It is the obvious mainstream in the modern computational linguistics. In syntax, it sequentially divides each sentence to contiguous word groups (constituents) down to separate words, so that the syntactic link between any two words is their belonging to the same constituent. We also include Head-driven Phrase Structure Grammars [10] to the mainstream. HPSG introduces a head sub-constituent in each constituent, and thus principally permits to consider another, dependency-like, type of links between words, but the authors of this theory had never explicitly allowed them.

The apostles and the majority of followers of the mainstream consider the constituency approach uniquely existing, thus leaving no room for other viewpoints on syntactic structures. Nevertheless, dependency grammars continue their development in parallel, with the same objective—to describe syntax and semantic of European and other languages in a more exact and consistent manner. The most developed linguistic theory keeping to dependency grammars is seemingly the Meaning ⇔ Text Model (MTM) by Mel'čuk [5, 6, 7], though there exist similar theories, e.g., [12]. Below we share methods and terminology of the MTM.

The dependency grammars arrange all word forms in a sentence in a dependency tree, so that each form has its unique governor and can have a number of dependents. The MTM labels dependency arrows pointing from a governor to its dependent, and

[*] The work was done under partial support of CONACyT and SNI, Mexico.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 210–219, 2002.

these labels are names of corresponding syntactic relations. The labels define the dependents as specific clause elements similar to traditional ones (subject, direct object, indirect object, etc.), but their distinguishing ability is more fine-grained than in any traditional approach.

According to postulates of the MTM, there are two syntactic levels, deep and surface, with different types of labels at their arrows. Surface Syntactic Relations (SSynRels) impart to dependency trees the property to reflect a preferable order of words in speech and writing, as well as all semantic features that reveal, together with to the deep morphological representation of all wordforms, the meaning of the sentence.

The dependency approach has at least three advantages:

- It establishes strictly conditioned correspondence of syntactic vs. semantic links between words, so that any other correspondences between dependency tree features and logic predicates of the semantic level are not needed.
- It permits to describe the syntactic constructions referred to as disjoint and non-projective. (By the way, there does not exist any rational decomposition to constituents for non-projective phrases like *la mejor cerveza en el mundo* ‘the best beer in the world.’)
- Through the introduction of SSynRel labels, sentences can be distinguished with the same dependency tree structures and the same labels at each node but with different meaning, such as the Spanish phrases *la palabra clave* ‘the keyword’ vs. *la palabra* “*clave*” ‘the word “palabra.”’

Deeper considerations about advantages of dependency approach in confrontation to constituency one are given in [6, 7, 9].

The MTM-induced inventories of SSynRels are already known for Russian [5], French [1, 2, 9], and English [7, 8]. To our knowledge, Spanish was passed over in such formalization so far, thus making room for Chomskian-type exercises in modern Spanish grammars [4].

The objective of this work is to propose a preliminary inventory of SSynRels in Spanish. As prototypes, we have taken SSynRels for Russian, English, and French. The formal criteria for definition of SSynRels and specific French SSynRels proposed in [9] were especially valuable for us, however we avoided a blind copying. In fact, Spanish has revealed several peculiarities, thus forcing us to introduce several new relations, maybe with awkward names. Our earlier attempt to study in detail Spanish subjectival SSynRel [3] had shown how thorny would be the road to detailing all the rest syntactic relations in Spanish.

2 Subjectival and Objectival Relations

Subjectival SSynRel connects the finite form of a full-meaning or auxiliary verb (the root of predicate subtree) and a noun or other wordform with substantival properties (the root of subject subtree): [*el*] *estudiante* ← *alcanza* [*buenas evaluaciones*]; [*esos*] *dos* ← *llegaron*; *fumar* ← *está* [*prohibido*]. In contrast to French and English, this relation is not obligatory in a full-fledged Spanish clause.

Among objectival relations, the following ones are especially important:

- ***Direct-Objectival*** relation: *dan* → [*la*] *prioridad* [*a...*]; *alcanza* → [*buenas*] *evaluaciones*; *veo* → *a* [*Elena*]; *notemos* → *que* [*un problema es...*]; *quiero* → a [*los míos*].
- ***Clitic-Direct-Objectival*** relation: *lo* ← *veo*; [*me*] *la* ← *darán.*
- ***Indirect-Objectival*** relation: *dan* → [*la prioridad*] *a* [*los problemas...*].
- ***Clitic-Indirect-Objectival*** relation: *les* ← *dan* [*la prioridad a...*]; *me* [*lo*] ← *darán..*

By contrast to corresponding English or French analogues, relations connecting direct and indirect objects may come in pairs in Spanish. This is the so-called pronominal duplication, i.e. the repetition (in some conditions, obligatory) of the same semantic valency on the surface syntactic level in the form (1) of a noun or a prepositional phrase with a noun or full-form (tonic) pronoun, and (2) of a clitic at the verb: ***Le*** *felicito* ***a usted.*** lit. 'You$_{\text{ACC}}$ I congratulate to you' ≈ 'It is you whom I congratulate' (ACC denotes accusative case); ***A Víctor le*** *acusa el director.* lit. 'To Victor him accuses the director.' ≈ 'It is Victor whom director accuses.' The repetition does not exist on the deep syntactic level, where a unique instantiation of the corresponding semantic valency exists.

A peculiarity of Direct-Objectival SSynRel in Spanish is that only inanimate entities subordinate to a verb without any preposition, whereas animated ones join the verb through the preposition *a*. Indirect-Objectival SSynRel joins the object to the verb through *a* in any case.

Quotative-Objectival relation differs from Direct-Objectival by the direct speech nature of the object: [*Juan me*] *dijo*: → ["*Sólo dos*] *palabras*"; ["*El éxito*] *es* [*producto de mucho trabajo*",] ← *señaló.*

Pseudo-Direct-Objectival relation is applicable to 'measuring' verbs: [*Juan*] *pesa* → [100] *kilos* 'Juan weights 100 kilograms'; [*La sesión*] *dura* → [*dos*] *horas.* 'The session lasts two hours.' It differs from Direct-Objectival by its impassiveness: **100 kilos son pesados por Juan* = '100 Kg are weighted by Juan.'

The other objectival relations link a verb with oblique object through prepositions of a broad variety. In their shape and functions, these relations are similar to their French or English analogues:

- ***Oblique-Objectival-1***: *depende* → *de* [*circunstancias*]; *traduzco* → *de* [*ruso a alemán*]; *luchará* → *contra* [*la ignorancia*].
- ***Oblique-Objectival-2***: *traduzco* → [*de ruso*] *a* [*alemán*].

Infinitival-Objectival relation subordinates infinitival objects (more frequent in Spanish than, say, in English) to the verbs through or without a preposition: *quiere* → *amar*; *empieza* → *a* [*trabajar*]; *acaba* → *de* [*trabajar*].

In this section, we consider also relations concerning benefactive actions. In many languages, a verb can subordinate a circumstantial complement reflecting person(s) to whose benefit the verb action is done (Cf. Eng. *to bye* or *to reserve* something for somebody). The beneficiary does not correspond to any semantic valency of such verbs, as compared with the dative verbs *to give, to propose, to concede...* Spanish benefactive verbs *comprar, reservar, conservar...* possess the property of the pronominal duplication similar to semantic valencies: *Emma* ***le*** *reservó unos lugares* ***a su***

familia. lit. 'Emma it$_{\mathrm{DAT}}$ reserved places to her family' (DAT denotes dative case). We introduce ***Benefactive*** relation governing benefactive prepositional group through *a* or *para*: [*le*] *compré* → [*un regalo*] *a* [*mi hermano*]; [*me*] *reservó* [*un lugar*] → *a* [*mí*]; whereas ***Clitic-Benefactive*** relation governs a clitic that plays the same role: [*le*] ← *compré* → [*un regalo*]; *me* ← *reservó* [*un lugar*]. Even when cooccurred, they correspond to a unique dependency at deeper levels.

3 Other Relations Controlled by Semantic Valencies

Several other relations are connected with semantic valencies, in a way different from those mentioned above. The following three permit to express a semantic valency filled by a situation with at least two actants:

- ***Subject-Copredicative*** relation: [*Jorge*] *regresó* → *rico.*
- ***Object-Copredicative*** relation: [*Jorge*] *quiere* → [*a Elena*] *delgada*; *creamos* → [*la estructura*] *posible*; [*me*] *considera* → *feliz.*
- ***Infinitive-Object-Copredicative*** relation: *oí* → [*a Juan*] *decir* [*algo a sus amigos*].

Agentive relation forms passive verbal and substantival constructions (*escritas* → *por* [*el presidente*]; *llegada* → *de* [*los turistas*]; *traducción* → *de* [*Lic. Ulloa*]), as well as absolute constructions with gerund (*estando* → *Pedro* [*aquí, no temo nada*]).

Absolute-Predicative relation is used in absolute constructions with participle: *arregladas* ← [*mis*] *maletas*, [*tomé el autobús a Cuernavaca*]; *garantizada* ← [*la*] *libertad* [*de creencias, dicha educación será laica*].

Copular relation connects a copulative finite verb to nominal part of predicate. Copulas are rather varied in Spanish: *es* → *fácil*; *son* → *posibles*; *están* → *cansados*: *es* → *de* [*baja eficiencia*]; *permanece* → [*el*] *profesor*; *estoy* → *de* [*acuerdo*]; *hace* → *frío*; *aparece* → *en* [*una posición difícil*].

Comparative relation connects the compared word with *que* or *de* preceding the second counterpart of the comparison: *mayor* → *que* [*Juan*]; *más* → *de* [*cinco*].

Adnominal-Completive relation connects pairs of nouns, where each represents a potential agent of an action or an action as such, except of when the dependent is the agent of an action expressed by the governor (cf. Agentive relation): *revisión* → *de* [*estado de vehículo*]; *estado* → *de* [*vehículo*]; *traducción* → *del* [*texto*]; *entrega* → *de* [*compras*]; *Secretaría* → *de* [*Transporte*]; *Presidente* → *de* [*México*]; *programa* → *de* [*desarrollo*]; *sistema* → *de* [*vida*]; *luchador* → *con* [*el narcotráfico*]. In fact, this relation is a dump of those relations between nouns, which differ from Agentive and Adnominal-Attributive (see below).

4 Relations for Modification, Attribution, Determination, etc.

Modificative relation, very common in Spanish, usually connects a noun with postpositioned adjective or participle agreed with the noun in number and gender: *ciencias*

→ *matemáticas*; *traje* → *nuevo*; *país* → *grande*; *hombre* → *bueno*; *ella* → *misma*; *evaluaciones* → *obtenidas*. In rarer cases the dependent word is a noun (*palabras* → *clave*; *términos* → *multipalabra*; *color* → *beige*; *Alemania* → *nazi, carta* → *poder*) and the agreement is absent. The same relation without agreement is recognized when gerund plays the role of a modifier: *mujer* → *peinándose*; *niño* → *corriendo* [*en la playa*].

Pre-Modificative relation differs from modificative one by the order of the linked words: *nuevo* ← *traje*; *primera* ← *dama*; *gran* ← *país*; *buen* ← *hombre*; *cualquier* ← *doctrina*; *cuánto* ← *dinero...*; [*el*] *primer* ← *caso*; *sola* ← *mujer*; *unas* ← *consideraciones*; *unos* [*veinte*] ← *hombres*. This relation occurs more rarely in Spanish and is introduced because some modifiers are used only before the modified noun, while other only after it or in the both positions. Several adjectives (*bueno, grande, triste, nuevo,* etc.) have different meaning in pre- and postposition, so that it is reasonable to introduce two homonymous lexemes indicating for each of them its uniquely possible relation of modificative type.

Descriptive-Modificative relation introduced agreed modifiers isolated with prosody in speech or with commas in the writing: [*esas*] *camas*, → *cómodas* [*y no tan caras,...*].

Relative relation points to the root of a subordinate modificative clause: [*el*] *artículo* → [*que*] *leí* [*ayer*]; [*la*] *muchacha* → [*la cual*] *llegó* [*aquí...*].

Descriptive-Relative relation differs from the previous one by the isolated character of the clause usually delimited with commas in the writing: *este artículo,* → [*que*] *leí* [*ayer,...*]; *Juan*, [*quien la*] *ama* [*tanto,...*].

Adnominal-Attributive relation connects a noun with a non-agreed modifier in the shape of preposition-noun group, usually without an article, so that none of the two connected nouns can fills a semantic valency of another: *puerta* → *de* [*madera*]; *aceite* → *de* [*girasol*]; *obreros* → *con* [*experiencias diferentes*].

Infinitival-Attributive relation connects a noun with a non-agreed modifier in the shape of preposition-infinitive group: *caso* → *de* [*detectar*]; *objetivo* → *de* [*estudiar*].

Descriptive-Attributive relation differs from the previous one by the isolated character of its non-agreed modifier usually delimited with commas: [*el Doctor*] *Aguirre*, → *de* [*Colombia, estuvo presente también*].

Quantitative relation connects a countable noun with a number: *cuatro* ← *hombres*; [*cincuenta y*] *tres* ← *libros*.

Determinative relation connects a noun or a substantivized word to a determiner, i.e. an article or a pronoun incompatible with an article: *el* ← *mundo*; *mi* ← *mamá*; *nuestro* ← *amor*; *aquella* ← *mujer*; *esos* ← *cinco*; *un* ← *restaurante*; *la* ← *cual*; [*Pedro*] *el* ← *Grande*. Some determiners can occasionally convert a wordform of any part of speech to a noun with generalized meaning: *el* ← *concluir*; *aquel* ← *murmurar*; *lo* ← *nuevo*; *lo* ← *mío*; *lo* ← *cerca*. We do not consider as determiners indefinite articles in plural (*unos* / *unas*), since they covey a special uncertainty meaning to the whole phrase (Cf. Modificative relation).

Appositive relation connects a noun with another noun or substantivized word, thus giving other name for the same object: *Pedro* → [*el*] *Grande*; *General* ←

Eisenhower; [*el*] *término* → "*sufijo*"; *interfaz* → *hombre* [– *máquina*]; *Doctor* ← *Guzmán*; *página* → *238*; *pacientes* → *mujeres*.

Descriptive-Appositive relation differs from the previous one by its isolative nature. The dependent group is usually delimited with commas or pasted dashes: [*Este*] *término* → *–sufijo–* [*se considerará más tarde*]; [*olvidaste a*] *mí,* → [*tu*] *madre*; [*el*] *Estado* → *–Federación*, [*estados y municipios– impartirá la educación*].

Appositive-Absolutive relation connects the root of a clause with appositive substantial group characterizing the governor clause as a whole: [*Cirujanos*] *operaron* → [*a un paciente en otro país, una*] *primicia* [*tecnológica real*].

Elective relation connects an adjective under comparison with prepositional group delimiting the condition of comparison: [*el*] *mejor* → *en* [*el mundo*]; [*la*] *primera* → *entre* [*las compañeras*].

Restrictive relation connects word of any part of speech to a word belonging to a closed group of 'logical restriction': *aún* ← *más* [*alta*]; *no* ← *aquí*; *más* ← *tarde*; [*eres*] *tan* ← *buena*; *no* ← *queremos*; *no* ← *sólo* ← *como* [*una estructura*].

5 Adverbial-Type and Parenthetical Relations

There are several adverbial-type relations in Spanish.

Adverbial relation connects governing verbs to adverbs or adverbial combinations (usually prepositional group): *caminan* → *lentamente*; *llegó* → [*la*] *semana* [*pasada*]; *hace* [*dos horas*] ← *llegó* [*el tren*]; *alcancé* → [*a Juan*] *caminando*; *murió* → *en* [*seguida*]; [*se*] *mantendrá* → *por* [*completo ajena...*].

Attributive-Adverbial relation differs from the previous by isolated character of the dependent: *En* [*el extranjero, Juan*] ← *trabajaba* [*poco*]**.**

The next two relations are adverbial in their syntactic functions but similar to modificative and appositive relations respectively in their outer form:

- ***Modificative-Adverbial***: *Elegante* [*como siempre*, *Juan se*] ← *fue*.
- ***Appositive-Adverbial***: [*Un*] *hombre* [*viejo, Juan*] ← *trabajaba* [*poco*].

Parenthetical relation differs from all the previous. It provides for a minimal semantic link between parenthesized phrase (usually, an opinion of the author of the utterance) and the main clause: *Por* [*desgracia*, *Juan*] ← *trabajaba* [*poco*]; *Afortunadamente*, [*Juan*] ← *trabaja* [*mucho*]; *fomentará,* → *a* [*la vez, el amor a la Patria*].

Adjunctive relation links an affirmative or negative word with the main clause: *Sí*, ← *estoy* [*de acuerdo*]; *Claro*, ← *es* [*posible*]; *No*, ← [*todavía no*] *llegaron.*

6 Analytical-Type and Clitic-Induced Relations

Strictly speaking, analytical structures of Spanish verbs are formed through the following three relations:

- ***Perfect-Analytical***: *hemos* → *comido*; *han* → *estado.*
- ***Passive-Analytical***: *fue* → *construida*; *está* → *prohibida*; *son* → *vendidos.*
- ***Progressive-Analytical***: *estoy* → *trabajando.*

The following relations exhibit features similar to analytical ones:

- ***Numeral-Junctive***: *cuarenta* ← *y* ← *ocho*; *ciento* ← *treinta*; *dos* ← *mil* ← *doscientos* ← *diecisiete*; *vigésimo* ← *tercero.*
- ***Nomination-Appositive***: *Juan* → *María* → *Gutiérrez* → *de* → *Albornoz*. Maybe, this relation requires splitting to two or more, connecting separately chains of first and last names.
- ***Sequential***: [*interfaz*] *hombre* → – *máquina.*

The clitic *se* imparts various meaning to the adjacent verbal forms, depending on a specific verbal lexeme and syntactic environment:

Passive-Junctive relation (*se* ← *venden* [*casas*] 'houses are on sail') reflects 'passive' contexts. On the deep syntactic level, the branch with *se* does not exist, and the whole structure coincides with passive construction without an explicit agent.

Reflexive-Junctive relation reflects an action directed to its own agent: [*Jorge*] *se* ← *afeita* 'Jorge shaves himself.'

Reciprocal-Junctive relation reflects the bilateral action: [*Elena y Marco*] *se* ← *besaron* 'Elena and Marco kissed each other.' The dictionary should mark possibility of reflexive or reciprocal meaning at a corresponding verb.

Identifying-Junctive relation reflects an action identified by the very clitic, since the verb has quite different meaning while used with or without *se*. In the example [*Pedro*] *se* ← *porta* [*bien*] the clitic refers to the verb *portarse* 'to behave', while the verb *portar* means 'to bring.'

For the three last relations, the branch with *se* does not exist either at the deeper level.

7 Prepositional and Conjunctional Relations

These are relations implied by a preposition:

Adnominal-Prepositional relation connects a preposition with a noun: *en* → [*la*] *cama*; *sin* → *temor.*

Verbal-Prepositional relation connects a preposition with a verb usually in infinitive form: *para* → *tener*; *sin* → *temblar*; [*finaliza*] *de* → *cantar*; *a* → *ver*; [*antes*] *de* → *resolver*. However, in Spanish the target verb in finite form is also possible: *según* → *comunican* [*periódicos...*].

The following are relations implied by a conjunction or a conjunctional word:

Subordinate-Conjunctional relation links *que* with a subordinate clause: [*Supongamos*] *que* → [*Jorge*] *llegue.*

Coordinate-Conjunctional relation links coordinative conjunction to the last member of a coordinative group: [*Juan*] *y* → *Elena*; [*clara*] *y* → *distintamente.*

Comparative-Conjunctional relation links *que* or *de* with the second counterpart of the comparison: [*mejor*] *que* → *Juan*; [*más*] *de* → *cinco*.

Binary-Junctive relation links two parts of a disjoint conjunctional construction: *si* [...] → *entonces* [...]; *tanto* → [*explícito*] *como* [*implícito*]; *o* [...] → *o* [...]; *bien* [...] → *bien* [...]; *no* [*triunfaran los cobardes*] → *sino* [*los valientes*].

8 Coordinative Relations

There are two variants of coordination in Spanish.

Coordinative relation is similar to that in other languages: *Juan* → *y* [*Clara*]; *sano* → *y* [*salvo*]; [*educación*] *preescolar*, → *primaria* → *y* [*secundaria*]; *bueno* → *pero* [*corto*].

Reductive-Coordinative relation, specific to Spanish, is applicable only for coordinating two or more adverbs with the suffix *-mente*: *estricta* → *y* [*exactamente*]. At the deep syntactical level, the cut forms coming first are restored to the full adverbs: *estrictamente* → *y* → *exactamente*.

9 Examples of Distinguishing Capabilities and of a Dependency Tree

Let us give now examples of distinguishing capabilities of SSyntRel labels. Namely, some phrases containing the same words coming in the same order can have the same dependency tree structure but with different meaning. We insist that these differences can be expressed just through various SSyntRels proposed above:

- [*la*] *palabra* $\xrightarrow{\text{Modific.}}$ *clave* ‘the keyword’ vs. [*la*] *palabra* $\xrightarrow{\text{Apposit.}}$ “*clave*” ‘the word “clave”’ vs. [*esta*] *palabra* $\xrightarrow{\text{Descrip.-Apposit.}}$ –“*clave”–...* ‘this word, “clave,”...’
- [*Él me*] *dijo* $\xrightarrow{\text{Dir.-Obj.}}$ [*dos*] *palabras*. ‘He said me two words.’ vs. [*Él me*] *dijo* $\xrightarrow{\text{Quotat.-Obj.}}$ [“*Dos*] *palabras*”. ‘He said me: “Two words.”’
- *Podemos* $\xrightarrow{\text{Adverbial}}$ [*hacerlo*] *naturalmente* ‘We can do this naturally’ vs. *Podemos* $\xrightarrow{\text{Parenth.}}$ [*hacerlo,*] *naturalmente*. ‘Naturally, we can do this.’

Following is an example of the dependency tree for a sentence taken from a newspaper:

Sin embargo, la Secretaría de Transporte podría hacer una nueva revisión de el estado legal de el vehículo extranjero en caso de detectar alguna irregularidad y negaría el reemplacamiento.

Each wordform is given below literally except for the composites *del* split to *de el*. For each wordform, except for the tree root, the governor’s number is given, thus picturing the whole tree:

1.	*Sin*	$\xleftarrow{\text{Parenth.}}$	7
2.	*embargo*	$\xleftarrow{\text{Prepos.}}$	2
3.	*la*	$\xleftarrow{\text{Determ.}}$	4
4.	*Secretaría*	$\xleftarrow{\text{Subject.}}$	7
5.	*de*	$\xleftarrow{\text{Adnom.-Compl.}}$	4
6.	*Transporte*	$\xleftarrow{\text{Prepos.}}$	5
7.	*podría*	(**Tree Root**)	
8.	*hacer*	$\xleftarrow{\text{Inf.-Object.}}$	7
9.	*una*	$\xleftarrow{\text{Determ.}}$	11
10.	*nueva*	$\xleftarrow{\text{Pre-Modif.}}$	11
11.	*revisión*	$\xleftarrow{\text{Dir.-Obj.}}$	8
12.	*de*	$\xleftarrow{\text{IAdnom.-Compl.}}$	11
13.	*el*	$\xleftarrow{\text{Determ.}}$	14
14.	*estado*	$\xleftarrow{\text{Prepos.}}$	12
15.	*legal*	$\xleftarrow{\text{Modif.}}$	14
16.	*de*	$\xleftarrow{\text{Adnom.-Compl.}}$	14
17.	*el*	$\xleftarrow{\text{Determ.}}$	18
18.	*vehículo*	$\xleftarrow{\text{Prepos.}}$	16
19.	*extranjero*	$\xleftarrow{\text{Modif.}}$	18
20.	*en*	$\xleftarrow{\text{Adverb.}}$	8
21.	*caso*	$\xleftarrow{\text{Prepos.}}$	20
22.	*de*	$\xleftarrow{\text{Infin.-Attribut.}}$	21
23.	*detectar*	$\xleftarrow{\text{IVerb.-Prepos.}}$	22
24.	*alguna*	$\xleftarrow{\text{Pre-Modif.}}$	25
25.	*irregularidad*	$\xleftarrow{\text{Dir.-Obj.}}$	23
26.	*y*	$\xleftarrow{\text{Coordin.}}$	7
27.	*negaría*	$\xleftarrow{\text{Coord.-Junct.}}$	26
28.	*el*	$\xleftarrow{\text{Determ.}}$	29
29.	reemplacamiento	$\xleftarrow{\text{Dir.-Obj.}}$	27

10 Conclusions

Surface Syntactic Relations introduced above (there are as many as 59 of them) permit to represent any Spanish sentence as a surface syntactic structure. Their distinguishing capability seems sufficient to make out all necessary semantic subtleties expressible on the surface syntactic level. However, the full-fledge elaboration of each relation only starts thereby. For each relation, it is necessary to determine all possible combinations of part of speech and narrower morpho-syntactic classes for both the governor and the dependent, as well as all permissible contexts within the dependency tree and in the linear order, including the words to be linked.

Acknowledgements

I thank Prof. Igor Mel'čuk for his drastic suggestions sometimes unintelligible for me, the earnest reviewer Marga Alonso Ramos for constructive criticism, and my students S.N. Galicia Haro and S. Miranda Jiménez for checking Spanish examples.

References

1. Apresian, Yu. D. Linguistic tools of the Franco-Russian automatic translation system ETAP-1. IV. Syntactic analysis of French (in Russian). Preprint 159. Moscow: Institute of Russian Language Publ., 1984.
2. Apresian, Yu. D., I. M. Boguslavsky et al. Linguistic tools of the Franco-Russian automatic translation system ETAP-1. V. Syntactic analysis of French (in Russian). Preprint 160. Moscow: Institute of Russian Language Publ., 1984.

3. Bolshakov, I. A., A. F. Gelbukh, G. O. Sidorov. On Subject-Predicate Agreement in Spanish. Proc. Intern. Conf. on Intelligent Text Processing and Computational Linguistics CICLing-2000, February 2000, Mexico City, p. 59-67.
4. Fuentes, J. L. Gramática moderna de la lengua española. Edit. Bibliográfica Internacional; Colombia; 1988.
5. Mel’čuk, I.A. An experience in the “Meaning-Text” linguistic models (in Russian). Moscow: Nauka Publ., 1974.
6. Mel’čuk. I. Dependency Syntax: Theory and Practice. SUNY Press, NY. 1988.
7. Mel’čuk, Igor. Dependency in Linguistic Description. Amsterdam: Benjamin (to be published).
8. Mel’čuk, I., N. Pertsov. Surface syntax of English. A Formal Model within the Meaning-Text Framework. Amsterdam: Benjamin, 1987.
9. Iordanskaja, L., I. Mel'čuk. The Notion of Surface-Syntactic Relation Revisited (Valence-Controlled Surface-Syntactic Relations in French). In: L. Iomdin, L. Krysin (eds.) Slovo v tekste i v slovare. Sbornik statej k semidesjatiletiju akademika Ju.D. Apresjana. Moscow: Jazyki russkoj kil'tury Publ.. 2000, p. 391-433.
10. Sag, I. A., and T. Wasow. Syntactic Theory: Formal Introduction. Stanford: CSLI Publ., 1997.
11. Sell, P. Lectures on Contemporary Syntactic Theories. Stanford: CSLI Publ., 1985.
12. Sgall, P., E. Hajičová, J. Panevová. The Meaning of the Sentence in its Semantic and Pragmatic Aspects. Dordrecht: Rejdel, 1986.

Parsing Ill-Formed Inputs with Constraint Graphs

Philippe Blache and David-Olivier Azulay

LPL, Université de Provence,
29 Avenue Robert Schuman,
13621 Aix-en-Provence, France
pb@lpl.univ-aix.fr

Abstract. We present in this paper a constraint-based account of robust parsing. More precisely, we propose a parsing technique relying on constraint satisfaction and its implementation by means of graphs. We show how constraint graphs constitute a flexible parsing framework both in terms of representation and implementation. They allow in particular to take into account non-connected elements, frequent in particular when parsing spoken languages. This approach is illustrated with some problematic examples (hesitations, phatics, repairs, etc.) taken from spoken french corpora.

1 Introduction

An important problem for robustness in NLP, in particular since building treebanks is concerned (see for example [11]), comes from parsing ill-formed inputs. Several techniques have been proposed, most of them consisting in recovering parsing errors or compensating inadequate parsing techniques with heuristics (see [10], [16]). One method consists for example in modifying the grammar by introducing new rules or new categories (e.g. an anonymous one).

Rather than ad hoc techniques, we propose in this paper a general approach relying on the idea that a linguistic description should be flexible by definition. More precisely, many problems encountered in robust parsing come from the linguistic formalisms themselves which cannot generate non-canonical structures. Moreover, they rely on the idea that a syntactic representation must connect all the linguistic objects (this is the case for generative theories as well as dependency ones). We propose to give up this restriction and allow the possibility of building non connected structures. One immediate consequence is the necessity of replacing trees with graphs. Such a representation offers other advantages such as the possibility of implementing crossing relations. This can be useful, according to the chosen linguistic formalism, for the analyze of some complex constructions such as parasitic gaps or adjunct extraction (see [13] or [6]). The interpretation of such structures, as shown in the next section, relies on an explicit representation of semantic relations.

Representing syntactic structures with graphs can be done in several ways. We adopt here the most general point of view which consists in representing directly the linguistic information by means of sub-graphs. We present in the first

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 220–229, 2002.

section of this paper an account of linguistic information in terms of constraints. We propose in particular the use of a limited set of constraints for the representation of syntax. The second section shows the equivalence between constraint sets and their representation by means of graphs. The third section describes how to use such graphs for building a syntactic representation. Finally, the last section presents this approach in the perspective of parsing spoken language inputs.

2 Linguistic Constraints

One of the important results of modern linguistics is that all linguistic information can be conceived as constraints over linguistic objects (see for example [12], [13], [14], [3] or [8] for a description of parsing in terms of constraint satisfaction). Constraints are typically relations between two (or more) objects: it is then possible to represent information in a flat manner by means of such relations. The first step in this work consists in identifying the kind of relations usually used in syntax. This can be done empirically and we suggest, adapting a proposal from [2], the set of following constraints: *linearity, dependency, obligation, exclusion, requirement* and *uniqueness.* In a phrase-structure perspective all these constraints participate to the description of a phrase. The following figure roughly sketches their respective roles, illustrated with some examples for the *NP*.

Constraint	Definition	Example
Linearity ($\prec$)	Linear precedence constraints.	$Det \prec N$
Dependency ($\rightsquigarrow$)	Dependency relations between categories.	$AP \rightsquigarrow N$
Obligation (*Oblig*)	Set of compulsory and unique categories. One of these categories (and only one) has to be realized in a phrase.	*Oblig(NP)* = {*N, Pro*}
Exclusion ($\not\Leftrightarrow$)	Restriction of cooccurrence between sets of categories.	*N[pro]* $\not\Leftrightarrow$ *Det*
Requirement ($\Rightarrow$)	Mandatory cooccurrence between sets of categories.	*N[com]* $\Rightarrow$ *Det*
Uniqueness (*Uniq*)	Set of categories which cannot be repeated in a phrase.	*Uniq(NP)* = {*Det, N, AP, PP, Pro*}

In this approach, describing a phrase consists in specifying a set of constraints over some categories that can constitute it. These constraints are specified in the same manner as GPSG (cf. [9]) first introduced the linear precedence statement. This information was stipulated in terms of *possible* realization: a LP constraint of the form $a \prec b$ indicates that b cannot precede a. This constraint is relevant only when these two categories cooccur into a phrase. In the present approach, a constraint is specified as follows. Let $\mathcal{R}$ a symbol representing a constraint relation between two (sets of) categories. A constraint of the form $a\mathcal{R}b$ stipulates that if a and b are realized, then the constraint $a\mathcal{R}b$ must be satisfied. The set of constraints describing a phrase can be represented as a graph connecting several categories. The following example illustrates some constraints for the *NP*.

- *Linearity:* *Det ≺ N; Det ≺ AP; AP ≺ N; N ≺ PP*
- *Requirement:* *N[com] ⇒ Det*
- *Exclusion:* *N ⇎ Pro; N[prop] ⇎ Det*
- *Dependency:* *Det ⇝ N; AP ⇝ N; PP ⇝ N*
- *Obligation:* *Oblig(NP) = {N, Pro, AP}*

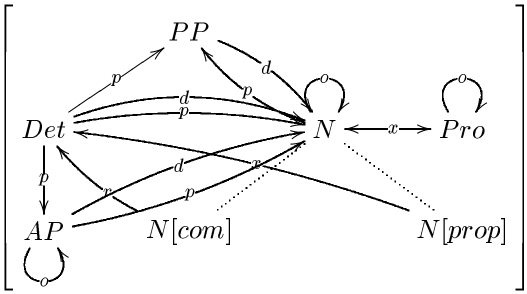
This toy description, in addition to the classical linear constraints, proposes a requirement relation between the common noun and the determiner. Such a constraint implements the complementation relation. The opposite relation, exclusion, indicates cooccurrency restriction. This kind of constraint allows together with requirement a precise description of what categories can constitute the described phrase. In our example, constraints indicate an impossible realization of a noun with a pronoun or a proper noun with a determiner. One can notice the use of sub-typing: as it is usually the case in linguistic theories, a category has several properties that can be inherited when the description of the category is refined (in our example, the type *noun* has two sub-types, *proper* and *common* represented in feature based notation). All constraints involving a noun also hold for its sub-types.

The dependency relation is a semantic one. It indicates that the dependent must combine its semantic features with the governor. In the same way as HPSG does now with the DEPS feature (cf. [6]), this relation concerns any category, not necessarily the governed ones. In this way, the difference between a complement and an adjunct is that only the complement is selected by a requirement constraint, both of them being constrained with a dependency relation. This also means that a difference can be done between the syntactic head (indicated by the *oblig* constraint and the semantic one (the governor of the dependency relation), even if in most of the cases, these categories are the same. Moreover, one can imagine the specification of dependencies within a phrase between two categories other than the head. More generally, it is always possible to specify relations directly between any categories of the phrase.

3 Constraint Graphs

A constraint graph is a set of constraints describing a phrase. Each constraint is represented by a set of nodes (categories) and an edge. A constraint graph is not necessarily connected. The set of constraints for the *NP* presented in the previous section can be represented by the following constraint graph:

(1) *NP* ↦



In this graph, each edge has a label indicating the kind of constraint (d stands for dependency, x for exclusion, p for precedence, o for oblig, r for requirement[1]). For example, the edge $\langle\ Det \xleftarrow{p} AP\ \rangle$ represents the linear precedence constraint $Det \prec AP$. The obligation constraint (as the uniqueness one, not represented here) is particular in the sense that it stipulates a constraint over one category. It is then represented as a reflexive relation. The sub-typing relation between the noun and its subcategories is indicated with a dotted line which does not correspond to a constraint edge. This schema illustrates the fact that relations can be specified indifferently over a category or its specifications which inherit the constraints of the root.

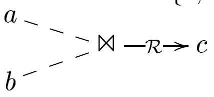
A constraint graph, as represented in the example (1), has a label corresponding to the phrasal category it describes. A specific relation, in the HPSG's head feature principle way, exists between the head of the graph (indicated by the obligation relation) and this label (its root).

A grammar is thus formed by a set of constraint graphs, each one corresponding to the description of a phrase. The basic idea consists then, given a set of categories, to verify its satisfiability according to the constraints. Let us call such a set, using a constraint programming terminology, an *assignment*. Each constraint of the grammar, independently from its membership to a constraint graph, is verified for an assignment. In this case, constraints are activated when the constrained variables correspond to some elements of the assignment.

When a constraint graph $\mathcal{G}$ is activated (i.e. the assignment contains categories constrained by $\mathcal{G}$), its label is instantiated and the corresponding category is considered as realized. If an assignment satisfies all the constraints of a constraint graph, the corresponding category is well-formed. But each assignment, whatever its form, can receive a characterization constituted by the state of the constraint graph after evaluation (formed by satisfied and non satisfied constraints). This means that it is possible to give a characterization of any object, even ill-formed.

One can notice an interesting characteristics: no constituency information belongs to the set of constraints. This constitutes an important difference with other phrase-structure approaches in which such information as to be encoded more or less implicitly. This comes from the fact that these methods define grammaticality in terms of constraints over a hierarchical structure. It is then necessary to build such a structure before verifying its properties. Insofar as building the structure is done following rules, only "well-formed" structures can be built. It is even the case in HPSG in which all the elements of a phrase has to be selected by a head or its complements.

[1] In this approach, constraints can be stipulated over sets of categories. We propose to represent this information by introducing a virtual node noted $\bowtie$. The set of categories is represented by the conjunction of nodes at the origin of a relation $\bowtie$. The constraint $\{a, b\}\mathcal{R}c$ is then represented by the graph:



In the approach described here, a phrase is instantiated only after constraint verification. Such a verification relies on relations between categories, as described above, and doesn't necessitate any hierarchical information: the projection notion doesn't play a role in this process and the government information is a relation among others. Moreover, we can notice at this point that an assignment can contain more categories than that actually constrained.

4 Description Graphs

In the same way as constraints in HPSG play the role of feature structure descriptions, constraint graphs allow to build the graph representing the syntactic structure of a given input. We call this graph a *description graph.* The mechanism consists, starting from the set of lexical categories corresponding to the words of the input, in building the subsets of categories (corresponding to assignments) that can be characterized by a constraint graph. When a subset is characterized, the corresponding phrase-level category is instantiated and completes the initial set of categories. New assignments can then be built, the process ends when no assignment can be characterized.

Characterizing an assignment $\mathcal{A}$ comes to evaluate the constraint system (i.e. the entire grammar) for $\mathcal{A}$. At the end of this process, it is possible to identify a subset of constraint graphs relevant[2] for $\mathcal{A}$. The constraint system after evaluation is formed by three different kind of constraints: *satisfied, violated* and *unspecified* constraints. This last state concerns constraints which variables don't belong to $\mathcal{A}$. A description graph is then formed with a constraint graph specifying these different states for each constraint.

The following example illustrates this process. The input here is formed with a determiner and a noun, the considered assignment is $\mathcal{A} = \{Det, N[com]\}$. As described in the previous section, the satisfiability of $\mathcal{A}$ is calculated over the entire constraint system (the grammar). In this example, we can suppose that all activated constraints belong to the *NP* constraint graph described in (1). More precisely, the following constraints can be evaluated for $\mathcal{A}$:

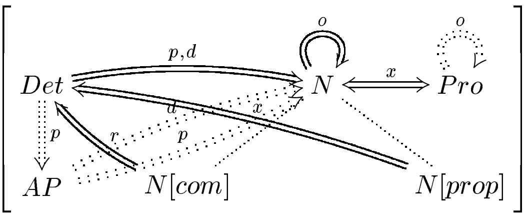
(1) $Det \prec N$
(2) $N[com] \Rightarrow Det$
(3) $N \not\Leftrightarrow Pro$
(4) $N[prop] \not\Leftrightarrow Det$
(5) $Det \leadsto N$
(6) $Oblig(NP) = \{N, Pro\}$

All these constraint are satisfied, which means that $\mathcal{A}$ is well-formed. The category corresponding to the constraint graph label (here *NP*) can then be instantiated and a new edge (labeled with *NP*) covering the categories of the assignment is created. The description graph built from (1) has the following form[3]:

[2] Note that a classical constituency-driven process would consist in first identifying the relevant constraint graphs and then evaluating them.

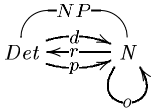
[3] In some graphs, for clarity, we can indicate as label of the same arrow several constraint types. This notation factorizes different constraints provided that they have the same source and target.

(2) *NP* ↦



In the remaining, for clarity, we only indicate in the description graphs the satisfied and violated constraints for realized categories. Moreover, the category instantiated by the description graph is specified as a label of an edge covering the assignment. The description graph in (2) is then represented as follows:

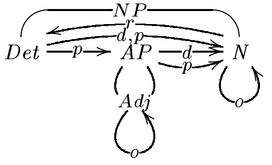
(3)



The instantiated category can be used to its turn into another assignment. We can notice that two different edge types coexist in this graph: edges representing constraints (which are labeled with the constraint name) and edges representing the coverage of a phrase. All categories immediately dominated by this edge are considered as constituents of the phrase.

Let's complete the previous example with an adjective phrase composed with a single adjective realized between the determiner and the noun. The new assignment is $\mathcal{A} = \{Det, AP, N[com]\}$. Following the same mechanism, the resulting description graph is of the form:

(4)



In this graph, a description graph has been built for the *AP* (a single relevant constraint is satisfied, the obligation one, no other constraint is violated). This category has then been instantiated and can be used, as any other category, for the rest of the process. The description graph of *AP* is thus integrated to the general description graph of the *NP*. What is interesting in this approach is that a relation can be expressed between any categories (for example directly between the adjective and the determiner), not necessarily into a government framework. This means that, contrarily to most of other approaches, a relation between two categories does not need a specific role for the head. Such a phenomenon can be for example illustrated by the anaphoric reference between a pronoun and a noun phrase (typically in dislocated constructions): in this case, one can stipulate a relation between the anaphoric clitic and the determiner requiring the determiner to be definite.

5 Parsing Ill-Formed Inputs

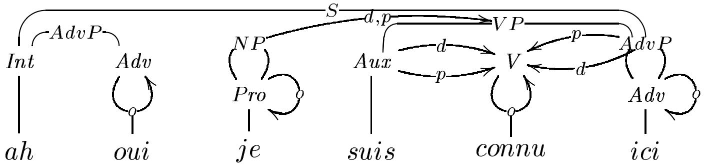
This approach is flexible in the sense that nothing is said more than the information explicitly stipulated by constraints. This is an open conception of syntactic information: what is explicit (the constraints) has to be verified, the rest simply cohabits with the structure. The point is that constraint verification doesn't mean imperative satisfiability (in other words, constraints are relaxed). As explained above, the characterization of an assignment can contain satisfied or non-satisfied constraints. Both are important for analyzing the properties of the corresponding phrase. In a robust parsing perspective, this means that it is possible to characterize incomplete or ill-formed phrases. Moreover, and perhaps more importantly, an assignment can contain unconstrained categories. Such categories don't have any relation with other categories without any consequence over the characterization itself. Such phenomenon occurs in particular in spoken languages with so-called *associated* elements (see [5]).

This property is possible because constituency is no more considered as a constraint. The set of categories belonging to a constraint graph only indicates some possible constituents, not necessarily the complete set of them. The following example illustrated this property.

(a) ah oui je suis connu ici
oh yes I am known here

This input, taken from a spoken French corpus[4], contains an interjection *ah* followed by an adverb *oui*. The problem consists in finding a status for the interjection and the adverb which is not in this case a sentence modifier.

(5)



We consider here that, in spite of the fact that no direct relation exists between the interjection and the adverb, these two categories can form an assignment characterizing an *AdvP*. More precisely, no constraint is violated by the assignment $\mathcal{A} = \{Int, Adv\}$ when verifying the *AdvP* constraint graph. It is then a possibility to consider *Int* as a possible element of *AdvP*. In the same way, the *AdvP* is not directly connected to the rest of the input. This representation shows that associated elements are part of the input (they are covered by an edge S) without having exact relation with the rest of the sentence.

The second example[5] illustrates the capacity of integrating other kind of non-connected elements such as phatic or interpolated clauses.

[4] Corpus "Bouliste", GARS, Université de Provence.

[5] Corpus "La bijoutière", GARS, Université de Provence.

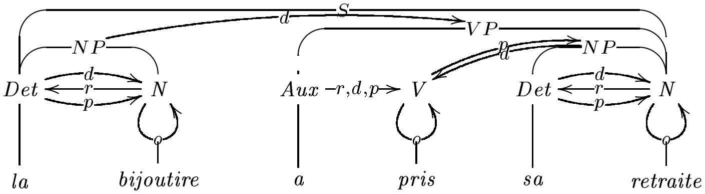
(b) la bijoutière a pris sa retraite et donc c'était une amie d'enfance de
the jeweler has taken her retirement and then it was a friend of childhood of
ma belle-mère d'ailleurs et donc euh elle a posé elle la question à ma belle mère
my mother-in-law moreover and then uh she has asked the question to
my mother-in-law

We isolate, for clarity, three sub-components of the input:

$S1$: *la bijoutière a pris sa retraite*
$S2$: *c'était une amie d'enfance de ma belle-mère d'ailleurs*
$S3$: *elle a posé elle la question à ma belle mère*

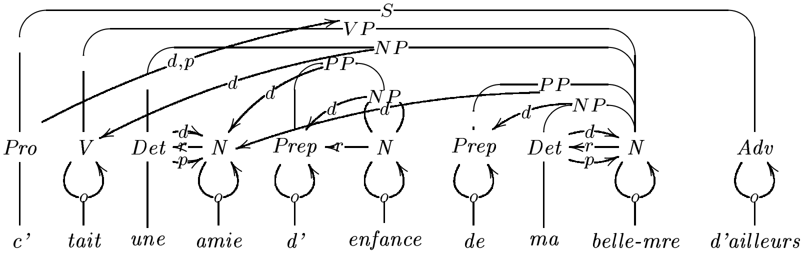
The first component $S1$ doesn't present any difficulty. Its description graph is presented in (6). This graph is connected: all its categories have some relation with the others. In this analysis, the auxiliary depends from the main verb. This analysis could be changed easily: if we consider that the dependency direction is the opposite (from the verb to the auxiliary) the edge direction simply has to be reversed.

(6)



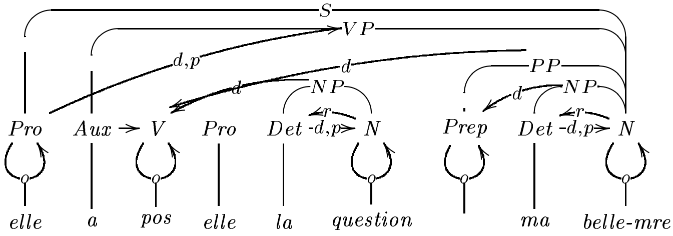
The second component $S2$ can be described with the description graph (7). Note that, for readability, some relations are not mentioned. The final adverb can be considered, as for the previous example, as an associated element, its role being more enunciative than syntactic. This category thus belongs to the S edge, but is not connected with any other category.

(7)



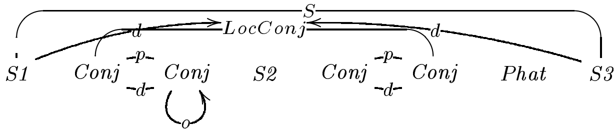
The last subcomponent $S3$ contains one problematic element: the repetition of the clitic *elle* in an unexpected position. This phenomenon can be interpreted as an hesitation or an insistence construction. In all cases, it cannot be analyzed as having the same kind of relation with the structure as the first clitic. We think then preferable to consider it as a non-connected element of the structure.

(8)



The characterization of the entire input can be done in the same manner. It contains several elements that cannot be directly connected to the structure: the phatic *euh* and the interpolated clause $S2$ (a repetition of the conjunctive locution together with some specific prosodic events not described here reinforce the interpolation interpretation). The description graph of the input can be presented as follows:

(9)



In this analysis, we consider that one of the conjunction is the head, the repetition doesn't play a specific role here. The conjunctive locution is formed by the two repetitions, each conjunct ($S1$ and $S3$) has a dependency relation with it (the locution plays the role of semantic head). In this analysis, the interpolated, the repetition and the phatic belong to the structure, but don't have any specific relation with other categories.

6 Conclusion

This technique has been tested over a written text corpus[6] of 80,000 words annotated with disambiguated POS tags. Two different systems were under evaluation: a shallow parser and a deep one. The result in both cases is a graph (or a set of graphs in the case of the deep parsing technique). The first system takes about 40 seconds to parse the entire corpus, the second about 3 minutes. It generates around 60 edges per sentence.

The approach described in this paper proposes a flexible account of ill-formed inputs and more generally for robust parsing. Unexpected elements, hesitations, phatics, repetitions, interpolated, etc. can easily be integrated to the syntactic description of an input. This makes this method well adapted for parsing spoken languages.

One of the interesting consequences of such a fully constraint-based account of syntax is that it calls into question the classical notion of ill-formedness (and then well-formedness). In this approach, any kind of input can be characterized. Some of them can satisfy all the constraints, some other not. In the end, the interesting

[6] A corrected version of the corpus CLIF built by Talana, see http://www.talana.linguist.jussieu.fr.

point when parsing a language (especially in the perspective of human-machine communication) is to give a description (i.e. analyze) of an input (whatever its form) more than calculating whether or not it belongs to a language. We have shown in this paper that a constraint-based approach (withdrawing the constituency notion) constitutes an efficient parsing technique.

References

1. Abney S. (1996) “Partial Parsing via Finite-State Calculus”, in proceedings of *ESSLLI'96 Robust Parsing Workshop.*
2. Bès G. & P. Blache (1999) “Propriétés et analyse d'un langage”, in proceedings of *TALN'99.*
3. Blache P. (2000) “Constraints, Linguistic Theories and Natural Language Processing”, in *Natural Language Processing*, D. Christodoulakis (ed.), Lecture Notes in Artificial Intelligence, Springer.
4. Blache P. & J.-M. Balfourier (2001) “Property Grammars: a Flexible Constraint-Based Approach to Parsing”, in proceedings of *IWPT- 2001.*
5. Blanche-Benveniste C. (1997) *Approches de la langue parlée en français*, Ophrys.
6. Bouma G., R. Malouf & I. Sag (2001) “Satisfying Constraints on Extraction and Adjunction”, in *Natural Language and Linguistic Theory*, 19:1, Kluwer.
7. Chanod J.-P. (2000) “Robust Parsing and Beyond”, in *Robustness in Language Technology*, Kluwer.
8. Duchier D. & R. Debusmann (2001) “Topological Dependency Trees: A Constraint-Based Account of Linear Precedence”, in proceedings of *ACL.*
9. Gazdar G., E. Klein, G. Pullum, I. Sag (1985), *Generalized Phrase Structure Grammar*, Blackwell.
10. Grinberg D., J. Lafferty & D. Sleator (1995), A robust parsing algorithm for link grammars, *CMU-CS-95-125*, Carnegie Mellon University.
11. Kübler S. & E. Hinrichs (2001) “From Chunks to Function- Argument Structure: A similarity-Based Approach”, in proceedings of *ACL.*
12. Maruyama H. (1990), “Structural Disambiguation with Constraint Propagation”, in proceedings of *ACL'90.*
13. Pollard C. & I. Sag (1994), *Head-driven Phrase Structure Grammars*, CSLI, Chicago University Press.
14. Sag I. & T. Wasow (1999), *Syntactic Theory. A Formal Introduction*, CSLI.
15. Tesnière L. (1959) *Eléments de syntaxe structurale*, Klincksieck.
16. van Noord G., G. Bouma, R. Koeling & M.-J. Nederhof (1998), in *Natural Language Engineering*, 4:1.

Part-of-Speech Tagging with Evolutionary Algorithms

Lourdes Araujo

Dpto. Sistemas Informáticos y Programación,
Universidad Complutense de Madrid
lurdes@sip.ucm.es

Abstract. This paper presents a part-of-speech tagger based on a genetic algorithm which, after the "evolution" of a population of sequences of tags for the words in the text, selects the best individual as solution. The paper describes the main issues arising in the algorithm, such as the chromosome representation and the evaluation and design of genetic operators for crossover and mutation. A probabilistic model, based on the context of each word (the tags of the surrounding words) has been devised in order to define the fitness function. The model has been implemented and different issues have been investigated: size of the training corpus, effect of the context size, and parameters of the evolutionary algorithm, such as population size and crossover and mutation rates. The accuracy obtained with this method is comparable to that of other probabilistic approaches, but evolutionary algorithms are more efficient in obtaining the results.

1 Introduction

The process of labeling each word in a sentence with its part-of-speech (noun, verb, etc) is called tagging and is a key step in the parsing process. Part-of-speech tagging is used in many language processing and generation applications: parsing, machine translation, information retrieval, speech recognition and generation, etc. The research in automatic part-of-speech tagging has increased a lot in the last years, probably due to the increasing availability of large tagged corpora.

Because languages are ambiguous and a word may have more than one tag, disambiguation methods are required to proceed with the tagging. There are different approaches for the disambiguation task, which can be classified in statistical [Jel85,DeM90,Mer94,CKPS92,SS94,Cha93] and rule-based[Qui93,Bri97]. Statistical taggers use large amount of data to establish the probabilities of each situation and neither require knowledge of the rules of the language nor try to deduce them. Most of these systems are based on Hidden Markov Models. Rule-based approaches apply language rules to improve the accuracy of the tagging. The Brill system [Bri95] extracts these rules from a training corpus, obtaining competitive performance with stochastic taggers.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 230–239, 2002.

The model presented in this paper belongs to the stochastic approach. In this model disambiguation is introduced by assigning different probabilities to a given tag depending on which are the neighbouring tags (context). The second important feature of the model is the use of an evolutionary algorithm to find the tagging of new sentences. Evolutionary algorithms are among the most efficient methods to deal with complex optimization problems, and can improve the performance of the typical algorithms used for the same purpose in other stochastic tagging approaches (such as the widely used of Viterbi).

Evolutionary algorithms mimic the principles of natural evolution: heredity and survival of the fittest individuals. Systems based on evolutionary algorithms maintain a population of potential solutions, and are provided with some selection process based on the fitness of individuals. The population is renewed by replacing individuals with those obtained by applying "genetic" operators to selected individuals. The usual "genetic" operators are *crossover* and *mutation*. Crossover obtains new individuals by mixing, in some problem dependent way, two individuals, called parents. Mutation gives a new individual by performing some kind of change on an individual. The production of new generations continues until resources are exhausted or until some individual in the population is fit enough. Evolutionary algorithms have been shown to be practical search and optimization methods, applied in diverse areas, such as planning or machine learning [Mic94]. They have also been applied to different issues of natural language processing, such as query translation [MD96], inference of context-free grammars [Wya91,TS95] and parsing [Ara00].

This work describes the evolutionary algorithm designed for the tagging task. In this algorithm, individuals are sequences of tags assigned to the words of a sentence. The computation of the fitness of the individuals, what decides their opportunities of surviving for the next generation, is based on the data extracted of a training corpus tagged by hand. These data are organized as *contexts*. The fitness function considers contexts whose length varies when necessary to increase the precision of the algorithm.

The rest of the paper proceeds as follows: Section 2 presents the evolutionary algorithm for tagging, including each genetic operator; Section 3 describes and discusses the experimental results, and Section 4 draws the main conclusions of this work.

2 Evolutionary Tagging

The evolutionary tagger is able to learn from a training corpus so as to produce a table of rules (contexts) called *training table*. Chromosome evaluation is done according to the training table. This table records the different contexts of each tag. The table can be computed by going through the training text and recording the different contexts and the number of occurrences of each of them for every tag in the training text.

Furthermore, the training corpus used to extract this stochastic information, can also be used here to tune the parameters of the evolutionary algorithm in a automatic way (Figure 1 shows a scheme of the process).

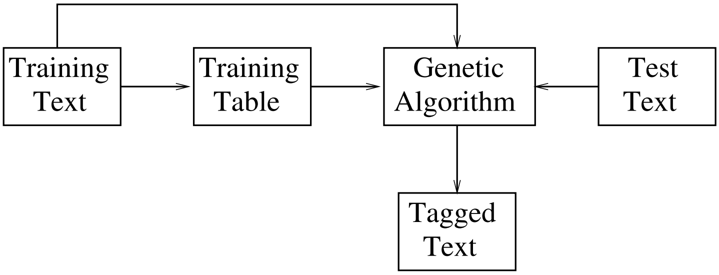


Fig. 1. Evolutionary Algorithm Training Scheme

The evolution process is run for each sentence in the text to be tagged. Evolution steps aim to maximize the total probability of the tagging of the sentences in the test corpus. The process finishes either if the fitness deviation lies below a threshold value (convergence) or if the evolutionary process has been running for a maximum number of generations. Let us consider the different elements of the algorithm.

2.1 Chromosome Representation

Chromosomes are sequences of genes which correspond to each word in the sentence to be tagged. Each gene is composed of a tag and additional information useful in the evaluation of the chromosome, such as counts of contexts for this tag according to the training table.

Initial Population. For a given sentence of the test corpus, the chromosomes forming the initial population are created by randomly selecting from a dictionary one of the valid tags for each word, with a bias to the most probable tag. Words not appearing in the dictionary are assigned the tag which appears more often with that given context in the training text.

2.2 Fitness: Chromosome Evaluation

The fitness function is a critical point in the evolutionary algorithm design, since it determines the quality of the solutions. In our case this function is directly related to the objective to be maximized, the total probability of each solution or chromosome. The raw data to obtain this probability are extracted from the training table. The fitness of each chromosome is evaluated according to the following points:

- Evaluation goes from left to right.
- Let n be the number of words in the sentence. Let w_i be the word in position i of the sentence. Let $f(g_i)$ be the estimated accuracy, or fitness, of the tag in position i or gene g_i. The fitness or measure of the adaptation of a chromosome is computed according to the formula

$$\sum_{i=1}^{n} f(g_i)$$

– In order to compute the fitness, each chromosome position or *gene* is previously evaluated according to the training table, so as to obtain an estimation of its accuracy. This is done in the following way:
  - Let $\mathcal{T}$ be the set of possible tags for the word at the position to be evaluated, i.
  - We are considering contexts of the following form:

    $$LC(T_{l_1}, \cdots, T_{l_{l_{LC}}}), T, RC(T_{r_1}, \cdots, T_{r_{l_{RC}}})$$

    where $T \in \mathcal{T}$ is the tag currently assigned to the word at the position being evaluated, LC represents the left part of context, with length l_{LC}, and composed of the tags $T_{l1}, \cdots, T_{l_{l_{LC}}}$, and RC represents the right part of the context, with length l_{RC} and tags $T_{r_1}, \cdots, T_{r_{l_{RC}}}$. Let occ_i be the number of occurrences of this context in the training text, which can be directly obtained from the training table.
    Let sum_i be the sum of all the occurrences of any context of the form:

    $$LC(T_{l_1}, \cdots, T_{l_{LC}}), T', RC(T_{r_1}, \cdots, T_{r_{RC}})$$

    $\forall T' \in \mathcal{T}$, which can be computed from the training table.
    Then, the adaptation or fitness of the gene at position i, $f(g_i)$, is computed as:

    $$f(g_i) = log(\frac{occ_i}{sum_i})$$

  - If the context does not correspond to any entry of the training table for this tag, then the context size is reduced by one tag and the counts for the new context are calculated from the table by adding all the entries in which the new context is present.
  - If even the shortest context does not appear in the table, then $f(g_i)$ is calculated according to the probability of the tag:

    $$\frac{\#T \text{ in any context}}{\sum_{T' \in \mathcal{T}} \#T' \text{ in any context}}$$

    $\#T$ denotes the number of occurrences of T.

– Some entries to the training table are too small —with respect to other entries for the same tag— for them to be significant, and so they can be ignored. In this case, it is necessary to determine a threshold. After some experiments, the adopted value is 5% of the number of occurrences of the most probable context for that tag.

The population is evaluated after each evolution step, and the new average fitness is computed.

2.3 Genetic Operators

Each evolution step some chromosomes from the current population are applied genetic operators to produce new individuals and renew the population. The genetic operators used herein are *crossover*, which combines two chromosomes to generate a new one, and *mutation*, which creates a new chromosome by changing a randomly selected gene in a chromosome of the previous generation. The design of the genetic operators must balance between the inheritance of the ancestors' properties and the exploration of new areas of the search space. The efficiency of the algorithm is very sensitive to the rates of crossovers and mutations performed at each step, which are input parameters.

Crossover. The crossover operation can be summarized as follows:

- Two chromosomes are randomly selected in the population with a probability proportional to its fitness.
- Then a crossover point is randomly selected, thus dividing the chromosome in two parts. In the selection of this partition point, positions corresponding to genes with low fitness are preferred.
- At this point two different kinds of crossover are considered, depending on the state of the evolution process:
  - For the first steps of the process (before the average fitness doubles that of the initial population): The first part of one parent is combined with the second part of the other parent thus producing two offsprings. This kind of crossover is expected to produce individuals quite different from the parents, thus helping to explore new areas of the search space.
  - For the following steps: At this stage it is expected to have quite accurate individuals, so the crossover applied is more conservative, trying to maintain the good characteristics of the parents. Thus, now only the tag at the crossover point is exchanged between the parents, producing again two new offsprings.

Mutation. The mutation operation can be summarized as follows:

- Mutation is applied to the chromosomes resulting of the crossover operations.
- Mutation is applied to each gene of these chromosomes with a probability given by the mutation rate.
- The tag of the mutation point is replaced by another of the valid tags of the corresponding word. The new tag is randomly chosen according to its probability.

3 Experimental Results

The corpus used to train the tagger has a huge influence on the performance. It must be a good sample of the language and most of the times the corpus used

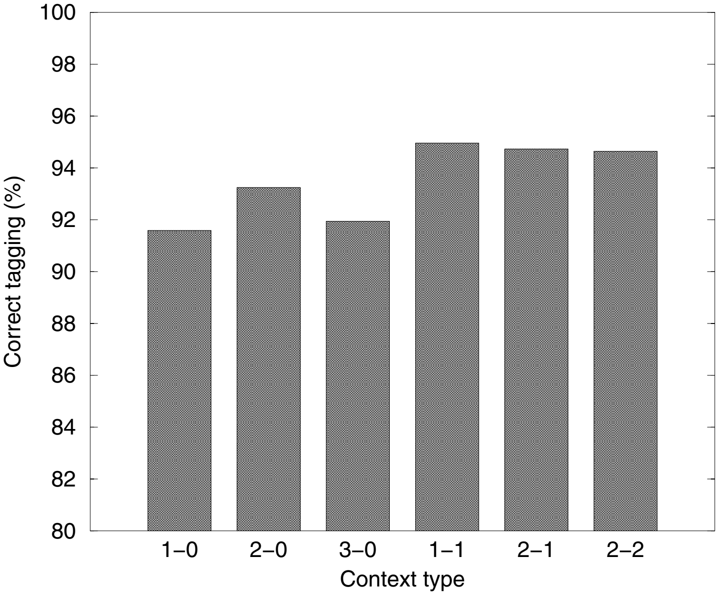


Fig. 2. Accuracy rate obtained for different sizes of the context, using a training corpus of 185000 words, a test text of 2500 words, a population size of 20 individuals, a crossover rate of 50%, and a mutation rate of 5%.

is domain-specific. In this work we have used the *Brown* corpus. The tag set is not too large, what favours the accuracy of the system, but at the same time is large enough to make the system useful.

Different experiments have been carried out in order to study the main factors affecting the accuracy of the tagging: size and shape of the contexts used for the training table, size of the training corpus and evolutionary parameters.

3.1 Influence of the Amount of Context Information

The way in which the context information is used by the tagger influences its performance. This piece of information can be fixed or variable and have different sizes. In order to determine the maximum length of contexts we have carried out a statistical analysis of the correlation between the tag at a given position in the Brown corpus and the tags separated a certain distance d; that is, we have computed $P(X_d|X_0)$ for different values of d. From those data we can determine the smallest value of d for which

$$P(X_d|X_0)/P(X_d) \approx 1$$

i.e. the minimum distance d for which the tag of the word at position 0 has no influence over the tag at position d. From this analysis we observe that $d = 3$ is

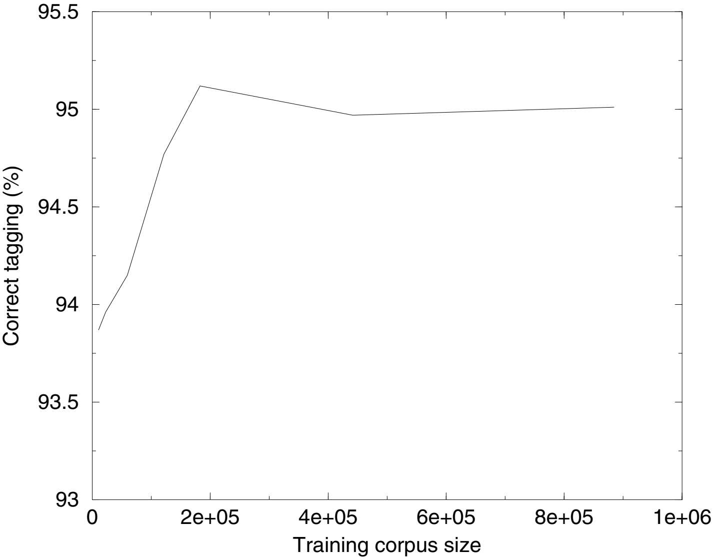


Fig. 3. Accuracy rate reached with different sizes of the training corpus, using contexts of the form 1-1, a test text of 2500 words, a population size of 20 individuals, a crossover rate of 50%, and a mutation rate of 5%.

a safe value and in most cases $d = 1$ is enough. Therefore, contexts longer that 3 are irrelevant.

Figure 2 shows the accuracy rates reached with different context sizes (always with $d \leq 3$) and shapes. Results show that the best performance is reached for small context sizes, such as 1-1, probably because with larger contexts the number of occurrences for many entries of the training table is not significant enough.

3.2 Influence of the Size of the Training Text

The next step is determining the influence of the size of the training text. Though intuitively it may seem obvious that the larger the training corpus, the better, we must take into account that the size of the training table slows down the evolutionary process, so only a significant increase of the accuracy can justify an increase of the training corpus. Figure 3 presents the increase of the accuracy with the size of the training corpus, showing that it saturates beyond a certain size (around 200,000 words).

The best Hidden Markov models typically perform at about the 95% level of correctness [Cha93]. Brill's model [Bri97], based on transformation rules, with a training set of 120,000 words and a separate test set of 200,000 words, obtained a tagging accuracy of 95.6%, which increased up to 96.0% by expanding the training set to 350,000 words. Therefore our results are comparable to that of

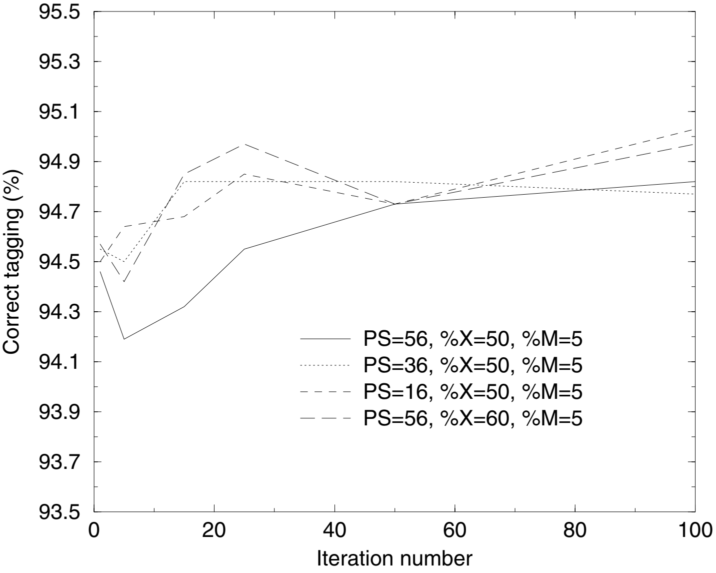


Fig. 4. Accuracy rate reached as a function of the number of iterations. PS stands for the population size, %X for the crossover rate, and %M for the mutation rate.

other probabilistic and rule-based approaches, with the advantage that they can be obtained in a very efficient way thanks to the evolutionary algorithm. Besides, in this case the algorithm turns out to be particularly fast because the best tagging can be reached with small populations and just a few iteration steps.

3.3 Study of the Evolutionary Algorithm Parameters

We have also investigated the parameters of the evolutionary algorithm: population size and crossover and mutation rates. Figure 4 shows the results obtained. We have observed that small populations are enough to obtain high accuracy rates, because the sentences are tagged one by one and so in general a small population is enough to represent the variety of possible taggings. This leads to a quicker algorithm. Crossover and mutation rates must be in correspondence with the population size: the larger the population, the higher the rates. It is therefore not only unnecessary but also inconvenient to increase the population.

4 Conclusions

The complexity of the tagging process for texts of important length can be treated by using stochastic methods that provide an approximate solution in a reasonable time, such as genetic algorithms. This work has developed a genetic algorithm that works with a population of potential taggings for each input sentence in the text. The evaluation of individuals is based on a training table composed of contexts extracted from a set of training texts.

Results indicate that the evolutionary approach is robust enough for tagging texts of natural language, obtaining accuracies comparable to other statistical approaches. The tests indicate that the length of the contexts extracted for the training is a determining factor for the results. However, there is a limit beyond which no further improvement can be obtained. Results have also shown the influence of the size of the training texts.

A study of the most frequent errors in the tagging has revealed the following points:

- As expected, words that require a tag that is not the most frequent or that appears in an odd context tend to be wrongly tagged.
- In general, the longer the sentence to be tagged, the better the results, because in large sentences there will be enough contexts to compensate the weight of some erroneous taggings.
- When deciding the size of the training table we must take into account that when tagging sentences whose words require one of its frequent tags and that appear in a frequent context, the longer the training text, the more accurate the tagging. However, when tagging sentences with words that adopt some of their most rare tags, the length of the training corpus can spoil the results.
- Another observation is the correlation between the parameters of the algorithm and the complexity of the texts to be analysed. The more complex the text (number of ambiguous words), the larger the population size required to quickly reach a correct tagging.
- Besides, as the population size increases, higher rates of crossover and mutation are required to maintain the efficency of the algorithm.

References

Ara00. L. Araujo. Evolutionary parsing for a probabilistic context free grammar. In *Proc. of the Int. Conf. on on Rough Sets and Current Trends in Computing (RSCTC-2000)*, 2000.

Bri95. E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4), 1995.

Bri97. E. Brill. Unsupervised learning of disambiguation rules for part of speech tagging. In S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*. Kluwer Academic Press, 1997.

Cha93. E. Charniak. *Statistical Language Learning*. MIT press, 1993.

CKPS92. D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proc. of the Third Conf. on Applied Natural Language Processing*. Association for Computational Linguistics, 1992.

DeM90. C. DeMarcken. Parsing the lob corpus. In *Proc. of the 1990 of the Association for Computational Linguistics*. Association for Computational Linguistics, 1990.

Jel85. F. Jelinek. Self-organized language modelling for speech recognition. In J. Skwirzinski, editor, *Impact of Processing Techniques on Communications*. Dordrecht, 1985.

MD96. T. Dunning M. Davis. Query translation using evolutionary programming for multilingual information retrieval II. In *Proc. of the Fifth Annual Conf. on Evolutionary Programming*. Evolutionary Programming Society, 1996.

Mer94. B. Merialdo. *Tagging english text with a probabilistic model*. 1994.

Mic94. Z. Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 2nd edition, 1994.

Qui93. J.R. Quinlan. *C 4.5: Programs for Machine Learning*. Morgan Kaufmann Publisher, 1993.

SS94. H. Schutze and Y. Singer. Part od speech tagging using a variable memory markov model. In *Proc. of the 1994 of the Association for Computational Linguistics*. Association for Computational Linguistics, 1994.

TS95. I.H. Witten T.C. Smith. A genetic algorithm for the induction of natural language grammars. In *Proc. IJCAI-95 Workshop on New Approaches to Learning Natural Language*, pages 17–24, Montreal, Canada, 1995.

Wya91. P. Wyard. Context free grammar induction using genetic algorithms. In *Proc. of the 4th Int. Conf. on Genetic Algorithms*, pages 514–518, 1991.

Formal Methods of Tokenization for Part-of-Speech Tagging[*]

Jorge Graña, Fco. Mario Barcala, and Jesús Vilares

Departamento de Computación,
Facultad de Informática,
Universidad de La Coruña,
Campus de Elviña s/n,
15071 – La Coruña,
Spain
{grana,barcala,jvilares}@dc.fi.udc.es

Abstract. One of the most important prior tasks for robust part-of-speech tagging is the correct tokenization or segmentation of the texts. This task can involve processes which are much more complex than the simple identification of the different sentences in the text and each of their individual components, but it is often obviated in many current applications.

Nevertheless, this preprocessing step is an indispensable task in practice, and it is particularly difficult to tackle it with scientific precision without falling repeatedly in the analysis of the specific casuistry of every phenomenon detected.

In this work, we have developed a scheme of preprocessing oriented towards the disambiguation and robust tagging of Galician. Nevertheless, it is a proposal of a general architecture that can be applied to other languages, such as Spanish, with very slight modifications.

1 Introduction

Current taggers assume that input texts are already tokenized, i.e. correctly segmented in *tokens* or high level information units that identify every individual component of the texts. This working hypothesis is not realistic due to the heterogeneous nature of the application texts and their sources.

Some languages, like Galician or Spanish, show phenomena that we have to handle before tagging. Among other tasks, the segmentation process takes charge of the identification of information units such as sentences or words. This process can be more complex than it may seem *a priori*. For instance, the identification of sentences is usually performed by considering certain punctuation marks. However, a simple dot can indicate the end of a sentence, but it could also correspond to the end of an abbreviation.

[*] This work has been partially supported by the European Union (under FEDER project 1FD97-0047-C04-02), by the Spanish Government (under project TIC2000-0370-C02-01), and by the Galician Government (under project PGIDT99XI10502B).

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 240–249, 2002.

In the case of words, the problem is that the spelling of word does not always coincide with the linguistic concept. Therefore, we have two options:

1. The simpler approaches just consider "spelled words" and extend the tags in order to represent relevant phenomena. For instance, the Spanish word `reconocerse` (*to recognize oneself*) could be tagged as `V000f0PE1`[1] even when it is formed by a verb and an enclitic pronoun, and the words of the Spanish expression `a pesar de` (*in spite of*) would be respectively tagged as `C31`, `C32` and `C33` even when they constitute only one term. However, this approach is not valid for Galician because its great morphological complexity would produce an excessive growth of the tag set.
2. Another solution is not to extend the basic tag set. As advantages, the complexity of the tagging process is not affected by a high number of tags, and the information relating to every linguistic term can be expressed more precisely. For instance, values of person, number, case, etc., can now be assigned to what was a simple pronoun before. As a drawback, this approach makes the tasks of the tokenizer more complex. Now, it not only has to identify "spelled words", but often also has either to split one word into several words, or join several words in only one.
   The greatest troubles arise when this segmentation is ambiguous. For instance, the words in the Spanish expression `sin embargo` will normally be tagged together as a conjunction (*however*), but in some context they could be a sequence of a preposition and a noun (*without seizure*). In the same way, the Spanish word `ténselo` can be a verbal form of `tener` with two enclitic pronouns (*hold it for him, her or them*), or a verbal form of `tensar` with only one pronoun (*tauten it*). This phenomenon is very common in Galician, not only with enclitic pronouns, but also with some expressions. For instance, the Galician word `polo` can be a noun (*chicken*), or the contraction of the preposition `por` and the article `o` (*by the*), or even the verbal form `pos` with the enclitic pronoun `o` (*put it*).

In our work, we have chosen the second option, i.e. to split and to join (to split e.g. the verb and their pronouns, and to join e.g. the different constituents of an expression). The first option, i.e. to work at the level of "spelled words", would in any case need a postprocessing step after tagging in order to identify the different syntactic terms of a text. This postprocessing step would perform tasks analogous to the ones involved in our preprocessor.

In this way, the aim of the present work is to develop a modular preprocessor, with generic algorithms, that can be used for different languages, but with better performance when linguistic information related to a particular language is provided. Therefore, it is also important to define what type of linguistic information is useful and how it will be integrated in the system in the cases where it is available.

[1] The tags that appear in this work come from projects GALENA (*Generation of Natural Language Analyzers*) and CORGA (*Reference Corpus of Current Galician*). Ap. A shows the description of every used tag. See `http://coleweb.dc.fi.udc.es` for more information of both projects.

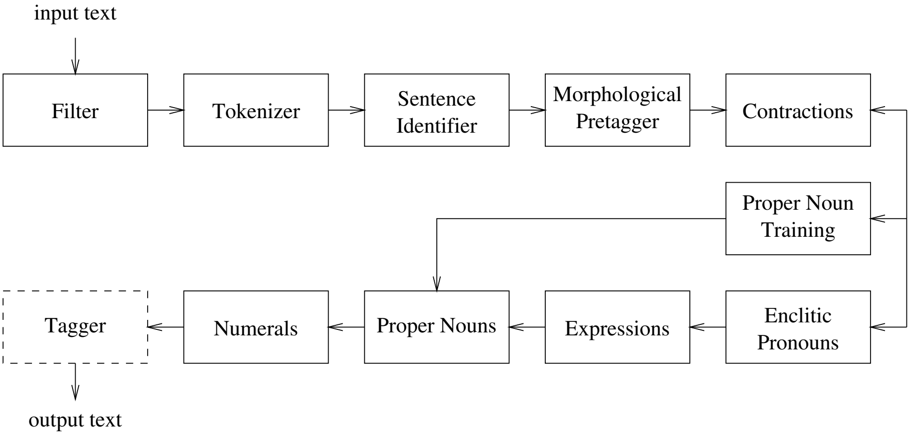


Fig. 1. General architecture of the preprocessor

Moreover, our preprocessor is specially designed as a prior phase of tagging, and it will also perform pretagging tasks. The underlying idea consists of letting the module that has the most information about a given phenomenon disambiguate this phenomenon. Therefore, as a second objective, we will also give the theoretical description of our tagger in order to complete the global presentation of the whole disambiguation process.

2 General Architecture of the Preprocessor

This section describes the different modules that are present in our preprocessor. These modules are shown in Fig. 1.

Filter. This module compacts delimiters (e.g. it removes multiple blanks or blanks at beginning of sentences) and performs conversions from typical source formats (e.g. HTML or XML) to plain text.

Tokenizer. The main function of this module is to identify and separate the tokens present in the text, in such a way that every individual word as well as every punctuation mark will be a different token. The module considers abbreviations, acronyms, numbers with decimals, or dates in numerical format, in order not to separate the dot, the comma or the slash (respectively) from the preceding and/or following elements. For this purpose, it uses two dictionaries (one of abbreviations and another one of acronyms), and a small set of rules to detect numbers and dates.

Sentence identifier. This module identifies sentences [3,5,6]. This task is more complex than it may seem *a priori*. The general rule consists of separating a sentence when there is a dot followed by a capital letter. However, we must take into account certain abbreviations to avoid marking the end of a sentence at their dots. For instance, this is the case of `Sr. González` (*Mr. González*). The module also considers acronyms so as not to separate their individual capital letters.

Morphological Pretagger. The function of this module is to tag elements whose tag can be deduced from the morphology of the word, and there is no more reliable way to do it. In this way, numbers are tagged with `Cifra`, and the tag `Data` is assigned to dates in formats like `7/4/82` or `7 de abril de 1982` (*April 7th, 1982*). In this latter case, we use the symbol `&` to join the different elements of the token, as can be seen in the following output:

```
7&de&abril&de&1982 [Data 7&de&abril&de&1982]
```

where the items inside the square brackets correspond to the tag and the lemma of the token under consideration.

Contractions. This module splits a contraction into their different tokens. At the same time, it assigns a tag to every one of them, by using external information on how contractions are decomposed. The module can work over other languages just by changing this information. For instance, the corresponding output for the Galician contraction `do` (*of the*) is:

```
de [P de]
+o [Ddms o]
```

i.e. `do` has been decomposed into the preposition `de` and the article `+o`. Note that the symbol `+` shows that an excision has taken place.

Proper Noun Training. Following [4,5,6], this module identifies the words that begin with a capital letter and appear in non-ambiguous positions, i.e. in positions where if a word begins with a capital letter then it is a proper noun. For instance, words appearing after a dot are not considered, and words in the middle of the text are considered. These words are added to a dictionary which is used later by the module Proper Nouns.

Enclitic pronouns. This module analyses the enclitic pronouns that appear in verbal forms. This is a major problem in Galician, where we can find up to four or five pronouns joined to the verbal form. The objective is to separate the verb from its pronouns and tag every one of them correctly. In order to perform this function, this module uses the following:

- A dictionary with as many verbal forms as possible.
- A dictionary containing the greatest possible number of verbal stems capable of presenting enclitic pronouns.
- A list with all the valid combinations of enclitic pronouns.
- A list with the whole set of enclitic pronouns, together with their tags and lemmas.

For instance, the decomposition of the Galician word `comelo` (*to eat it*) is:

```
come [V0f000 comer]
     [V0f1s0 comer]
     [V0f3s0 comer]
     [Vfs1s0 comer]
     [Vfs3s0 comer]
+o   [Raa3ms o]
```

where we can see that the components are `comer` (which can be infinitive, conjugated infinitive or subjunctive future) and `+o` (which is the pronoun).

Expressions. This module joins together the different tokens that make up an expression [2]. It uses two dictionaries: the first one with the expressions that are uniquely expressions, e.g. `a pesar de` (*in spite of*), and the second one with those that may be expressions or not, e.g. `sin embargo` (*however* or *without seizure*). In this case, the preprocessor simply generates all the possible segmentations, and then the tagger selects one of those alternatives later. The formalism used by our preprocessor to represent this kind of phenomenon has the following aspect:

```
<alternative>
   <alternative1>
      sin
      embargo
   </alternative1>
   <alternative2>
      sin&embargo
   </alternative2>
</alternative>
```

In Sect. 4 we will explain further how the tagger considers this representation in order to perform the disambiguation properly.

Proper Nouns. This module uses a specific dictionary of proper nouns to which proper nouns identified by the Proper Noun Training module can be added, as we saw above. With this resource, this phase of the preprocessor is able to detect proper nouns whether simple or compound, and either appearing in ambiguous positions or in non-ambiguous ones.

Numerals. This module joins together several numerals in order to build a compound numeral. For instance, every component of `mil ciento veinticinco` (*one thousand one hundred and twenty-five*) is joined with the rest in the same way as the components of an expression, obtaining only one token. Unlike the case of expressions, the tag assigned by the preprocessor here is definitive.

3 Mixed Problems

In order to form an impression of the complexity of the problems detected, we give some examples of typical cases that were solved.

Example 1. Consider the Galician expression polo tanto. It is an uncertain expression, i.e. polo tanto can be an expression (*therefore*); in its turn, polo can be a noun (*chicken*), a contraction (*by the*) or a verb with an enclitic pronoun (*put it*); and on the other hand, tanto can be a noun (*goal*) or an adverb (*so much* or *both*), when it does not form part of the expression. The preprocessor represents all the alternatives as follows:

```
<alternative>
   <alternative1>
      polo [Scms polo]
      tanto
   </alternative1>
   <alternative2>
      por  [P por]
      +o   [Ddms o]
      tanto
   </alternative2>
   <alternative3>
      po   [Vpi2s0 pór] [Vpi2s0 poñer]
      +o   [Raa3ms o]
      tanto
   </alternative3>
   <alternative4>
      por&+o&tanto
   </alternative4>
</alternative>
```

The following set of sentences contains examples of every different sense:

- **Noun+Adverb**:
  Coméche-lo polo tanto, que non quedaron nin os osos
  (*You chewed the chicken so much that not even the bones are left*).
- **Preposition+Article+Noun**:
  Gañaron o partido polo tanto da estrela do equipo
  (*They won the match by the goal of the star of the team*).
- **Verb+Pronoun+Adverb**:
  Pois agora, polo tanto ti coma el
  (*So now, both you and he should put it*).
- **Expression**:
  Estou enfermo, polo tanto quédome na casa
  (*I am ill, therefore I am staying at home*).

Example 2. As we saw before, an example of conflict between two possible decompositions of enclitic pronouns is the Spanish word ténselo, which can be tense plus lo (*tauten it*), or ten plus se plus lo (*hold it for him, her or them*), yielding these two alternatives:

```
<alternative>
   <alternative1>
      ténse  [V2spm0 tensar]
      +lo    [Re3sam el]
   </alternative1>
   <alternative2>
      tén    [V2spm0 tener]
      +se    [Re3yyy se]
      +lo    [Re3sam el]
   </alternative2>
</alternative>
```

4 The Tagger

Although the presentation of our tagger is not one of our main objectives, a short description of its working principles is of certain importance. Due to the ambiguous segmentations described above, this tagger must be able to deal with streams of tokens of different lengths. That is, it not only has to decide the tag to be assigned to every token, but also to decide whether some of them form or not the same term, and assign the appropriate number of tags on the basis of the alternatives provided by the preprocessor. For instance, we show in Fig. 2 the streams to be evaluated if the third word has four possible segmentations.

To perform this process, we could consider the individual evaluation of every trellis and their subsequent comparison, in order to select the most probable one. It would therefore also be necessary to define some objective criterion for that comparison. If the tagging paradigm used is the framework of the hidden Markov models [1], as is our case, that criterion could be the comparison of the normalization of the cumulative probabilities[2]. One reason to support the use of hidden Markov models is that, in other tagging paradigms, the criteria for comparison may not be so easy to identify.

Be that as it may, the individual evaluation of every possible combination of alternatives could involve a very high computational cost. For instance, if another word with two possible segmentations appears in the same sentence, we would have $4 \times 2 = 8$ different streams of tokens. For this reason, we prefer to design an extension of the Viterbi algorithm, able to evaluate streams of tokens of different lengths over the same trellis (see Fig. 3) with a time complexity comparable with that of the classic algorithm. This dynamic extension is still an item of future work, but it will constitute the final step and its output will be precisely the now segmented and disambiguated text.

[2] Let us call p_i the cumulative probability of the best path (the path marked with the thickest line) in the trellis i of Fig. 2. These values, i.e. p_1, p_2, p_3 and p_4, are not directly comparable. But if we use logarithmic probabilities, we can obtain normalized values by dividing them by the number of tokens. In this case, $p_1/5$, $p_2/6$, $p_3/7$ and $p_4/7$ are now comparable.

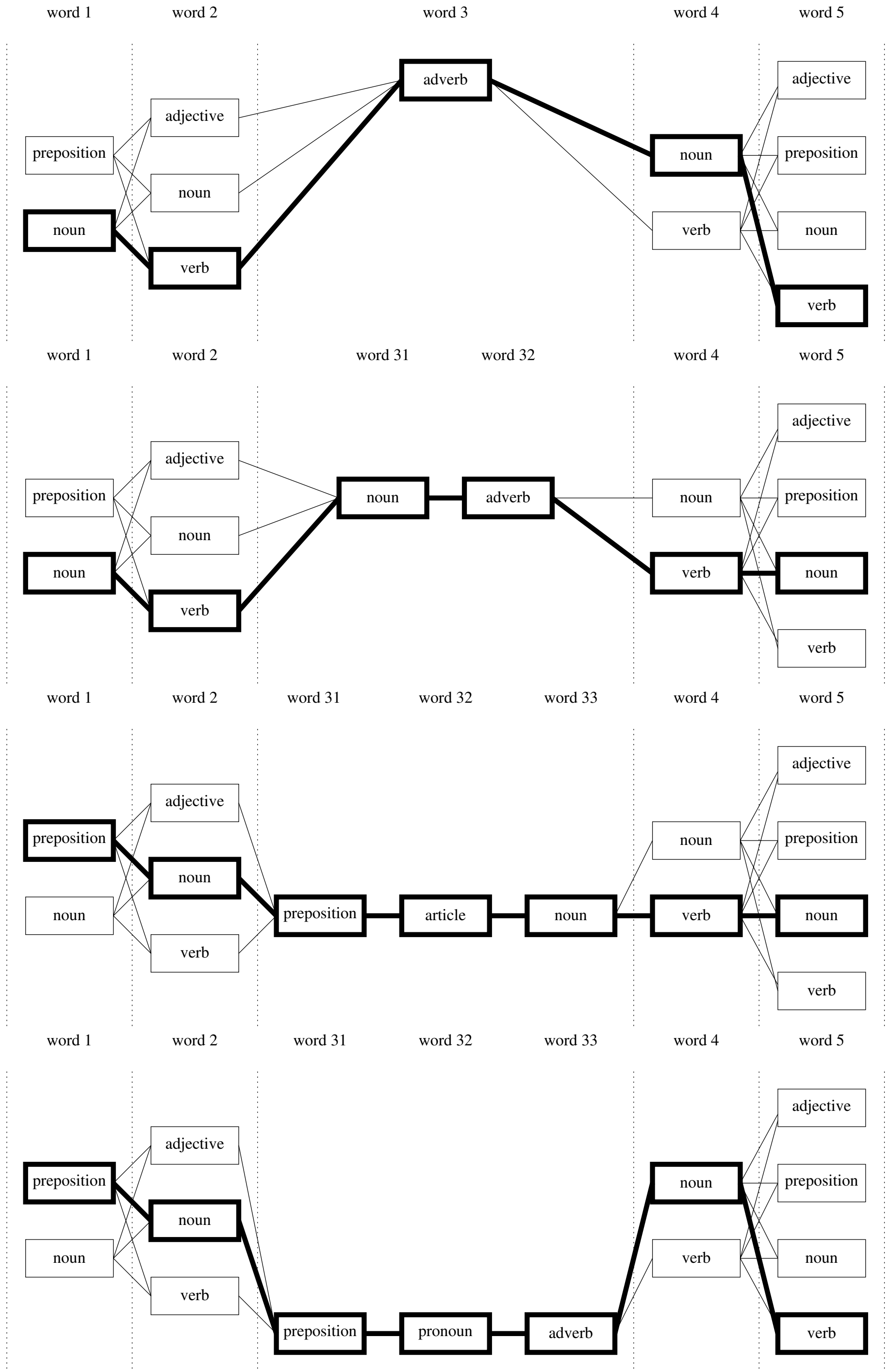


Fig. 2. Set of trellises for a set of different segmentations

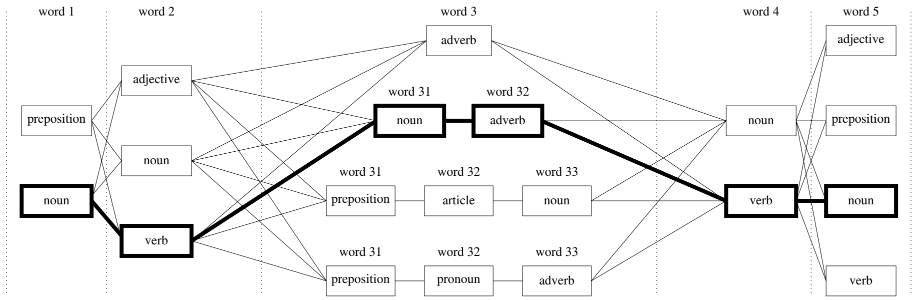


Fig. 3. A set of different segmentations represented in the same trellis

5 Conclusion

This work is focused on the description of specific and formal methods for preprocessing and tokenization. As we have shown, the complexity of the phenomena that appear at this level is so high that there does not even exist a strategy to determine the correct order of handling these phenomena. Our proposal attempts to fill this gap by a general scheme that avoids the particular casuistry of every phenomenon detected and every language.

The explanation has been oriented towards obtaining improvements in tagging. Nevertheless, tokenization is not only useful for automatic disambiguation. The place of the tagger could be filled by any other kind of analyser (syntactic, semantic, etc.), or simply by a *scanner* which provides all the possible segmentations and their corresponding tags, allowing the tasks involved in the manual process of building new reference texts to be performed more comfortably. In fact, this latter use is currently being intensely exploited for Galician, a language for which linguistic resources hardly exist.

Among our future proposals, besides the implementation of a dynamic version of the Viterbi algorithm, we aim to improve the generalization of our algorithms to simplify their adaptation to other languages. On the other hand, we also need to cover more and more preprocessing tasks that are still under study, but without using a great amount of linguistic resources that may not exist. If they are available, they would be used simply to refine the global behaviour.

The final objective of the general architecture of preprocessing presented here is its integration in an information retrieval system, and we expect that the modules described will contribute to improving the performance of the system.

References

1. Brants, T. (2000). TnT - A statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seattle.
2. Chanod, J.-P.; Tapanainen, P. (1996). A Non-deterministic Tokeniser for Finite-State Parsing. In *Proceedings of the Workshop on Extended finite state models of language (ECAI'96)*, Budapest.

3. Grefenstette, G.; Tapanainen, P. (1994). What is a word, What is a sentence? Problems of Tokenization. In *Proceedings of 3rd Conference on Computational Lexicongraphy and Text Research (COMPLEX'94)*, July 7-10.
4. Mikev, A. (1999). A knowledge-free Method for Capitalized Word Disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, June 20-26, Maryland.
5. Mikev, A. (2000). Document Centered Approach to Text Normalization. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2000)*, July 24-28, Athens, pp. 136-143.
6. Mikev, A. (2000). Tagging Sentence Boundaries. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'2000)*, Seatle, pp. 264-271

A Description of the Tags

This appendix describes the tags that appear in the examples. As we mentioned above, these tags come from the tag sets of the projects Galena and Corga.

Galena tags	
`C31`	First element of a conjunction of three elements
`C32`	Second element of a conjunction of three elements
`C33`	Third element of a conjunction of three elements
`Re3sam`	Pronoun: enclitic, accusative, masculine, third person singular
`Re3yyy`	Pronoun: enclitic, accusative or dative, masculine o feminine, third person, singular or plural
`V000f0PE1`	Verb: infinitive with one enclitic pronoun
`V2spm0`	Verb: present, imperative, second person singular

Corga tags	
`Cifra`	Number
`Data`	Date
`Ddms`	Article: determinant, masculine, singular
`P`	Preposition
`Raa3ms`	Pronoun: atonic, accusative, masculine, third person singular
`Scms`	Noun: common, masculine, singular
`V0f000`	Verb: infinitive
`V0f1s0`	Verb: infinitive conjugated, first person singular
`V0f3s0`	Verb: infinitive conjugated, third person singular
`Vfs1s0`	Verb: future, subjunctive, first person singular
`Vfs3s0`	Verb: future, subjunctive, third person singular
`Vpi2s0`	Verb: present, indicative, second person singular

Sepe: A POS Tagger for Spanish

Héctor Jiménez[1] and Guillermo Morales[2]

[1] Faculty of Computer Science, Autonomous University of Puebla,
C.U. 72570, Puebla, Mexico
hjimenez@fcfm.buap.mx

[2] Molecular Engineering Program, Mexican Institute of Petroleum,
on leave of absence from Computer Science Section, CINVESTAV, Mexico
gmorales@cs.cinvestav.mx

Abstract. We describe a part-of-speech tagging system specially designed to tag Spanish texts using small linguistic resources. Nevertheless, the tagger obtains encouraging results. We have found and exploited useful contextual parameters to tag ambiguous and unknown words. Our tagger is mainly supported by word lists and one corpus with around 10^4 words. The system has been tested for texts of the so called "news" genre and is still on continuous development.

Keywords: Spanish language, part-of-speech tagging.

1 Introduction

There are several part-of-speech (POS) taggers for the Spanish language ([6], [9], [11]). Even when the tagging performance is the most important matter, when building a tagger the size of available linguistic resources and the complexity of all involved parameters are quite relevant. In case of restricted domains it might be convenient to develop more precise taggers, as is suggested in [7]. We think that our experience is useful in new taggers development. Our tagger, named Sepe, has been easily implemented since it does not require neither copious resources at the beginning nor so much programming effort. The available corpus influenced strongly Sepe's implementation. We have started from a set of small resources and we followed simple criteria. The first design criterion for Sepe was to tag well-known words; the tagging of uncertain words was considered later. From the right beginning we collected a list of relevant words, a list of suffixes, a set of conjugation rules and a corpus composed by texts of "news" genre with around 10^4 words, extracted from *Corpus del Español Mexicano Contemporáneo* (CEMC) [5] (*Contemporary Mexican Spanish Corpus*). Sepe refines its criteria at each step. In the last steps, Sepe is strongly supported by a supervised learning method, applied to word contexts. The corpus is essential to determine the most important features of ambiguous and unknown words contexts. By aid of the corpus some patterns leading to additional morphosyntactic rules are identified. Also, word endings alleviate the small number of possible contexts in our corpus without restricting word tags.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 250–259, 2002.

This paper is divided in five sections. In section 1 some notation and the learning algorithm to choose the POS tag for some ambiguous and unknown words is introduced. Section 2 describes the resources used in the tagger system. A tagging example is presented in section 3. In section 4 a performance test is shown. At the end the conclusions appear.

2 Background

The POS of a word is a tag on the set {`VERB, NOM, ADJ, ADV, CONJ, ART, PRON, CONT, NP, NUM`}, corresponding to *verbal form, noun, adjective, adverb, conjunction, article, pronoun, contraction, proper noun*, and *number* respectively, or punctuation signs as *comma, period*, etc.: `COM, PTO, PTC, DPT, INT, AIN, ADM, AAD, GUI,` and `SUS`. A text $T = [w_i]_i$ is a sequence of words pertaining to a vocabulary: for each i, $w_i \in \mathcal{V}$. A representative text for certain linguistic phenomena is called *corpus*. For each word w, we will denote its *ending* of length k as w_k. A *context* for w occurring in a text T, say at position j, is a subsequence $\bar{w}_{j-p} \dots \bar{w}_{j-1}\ \bar{w}_{j+1} \dots \bar{w}_{j+q}$, with $p, q > 0$, where each $\bar{w}_i$ is either a word in T, or a feature as an ending or a tag. A *dictionary* is a collection of words or suffixes. For each element in a dictionary a POS is assigned.

We distinguish several types of words with respect to a given corpus: a *definite word* has an invariable POS in all contexts, e.g. the article “el”; an *ambiguous word* has at least two contexts with different POS, e.g. “la” can have POS `PRON` (as in “*yo la amo*” (I love her)) or `ART` (as in “*la novia*” (the bride)); an *unknown word* neither appears on the dictionary nor satisfies any rule related to its endings. The 100 *most frequent words* (MFW), taken from the analysis carrying out in the CEMC [5] are furtherly classified: the *definite frequent word* are the MFW that are definite words, the *ambiguous frequent word* are MFW ambiguous words, and the *frequent verbal forms*, which are conjugations of verbs in MFW. If a frequent verbal form is also ambiguous, then it will be taken as an ambiguous frequent word.

2.1 Learning Algorithm

Contexts, Words and Features. The contexts around an ambiguous or unknown word help us to determine the word tag. In order to face this task, we will consider a *training set* of contexts S, whose elements are thus *training instances*, and their features. Let us go into some technicalities in order to introduce the measures that will allow us to choose the attributes in contexts useful in determining word tags.

Let $\mathcal{A} = \{A_1, \dots, A_m\}$ be a collection of attributes, for each $A \in \mathcal{A}$ let $D(A)$ be its domain (set of features) and let $U = \prod_{A \in \mathcal{A}} D(A)$ be the universe of instances. Given any set of training instances $S \subset U$, for any $A_i \in \mathcal{A}$ and any $a \in D(A_i)$, let $S_{A_i \leftarrow a} = \{X = (x_1, \dots, x_m) | x_i = a\}$. With respect to a given collection of classes $\mathcal{C} = \{C_1, \dots, C_n\}$, where $\forall j,\ C_j \subset U$, the *entropy* of S is

$$\mathit{info}_{\mathcal{C}}(S) = -\sum_{j=1}^{n} \mathrm{freq}_j(S) \cdot \log_2 \mathrm{freq}_j(S) \tag{1}$$

where $\mathrm{freq}_j(S) = \frac{\#(S \cap C_j)}{\#(S)}$ is the "relative frequency of elements in S that fall in class C_j". For each $A \in \mathcal{A}$ let $N_{S_A} = \frac{\#(S_{A \leftarrow a})}{\#(S)}$ and

$$\mathit{information\ gain}: \ \mathit{gain}_{\mathcal{C},A}(S) = \mathit{info}_{\mathcal{C}}(S) - \sum_{a \in D(A)} N_{S_A} \mathit{info}_{\mathcal{C}}\left(S_{A \leftarrow a}\right) \tag{2}$$

$$\mathit{split\ info}_A(S) = -\sum_{a \in D(A)} N_{S_A} \log_2 N_{S_A} \tag{3}$$

$$\mathit{gain\ ratio}_{\mathcal{C},A}(S) = \frac{\mathit{gain}_{\mathcal{C},A}(S)}{\mathit{split\ info}_A(S)} \tag{4}$$

($\mathit{split\ info}_A(S)$ does not depend on $\mathcal{C}$). The weights to be used as contextual features are given as $p_i = \mathit{gain\ ratio}_{\mathcal{C},A_i}(S)$.

MBL. We use Memory-Based Learning (MBL) [1] to classify words. Since it is a supervised learning method, it requires a collection of instances in order to classify any new instance: MBL assigns the new instance to the class of the most likely instance from the training set, or equivalently, to the class of the "closest" training instance towards the new instance. Hence, a distance function should be used in the classification. Let us introduce suscintly the formal details:

Suppose fixed a set of training instances S and a current partition $\mathcal{C}$ of classes. Given two instances $X = (x_1, \ldots, x_m)$, $Y = (y_1, \ldots, y_m) \in U$ let $\Delta(X,Y) = \sum_{i=1}^{m} p_i \cdot \overline{\delta}(x_i, y_i)$, where $\mathbf{p} = (p_1, \ldots, p_m) \in (\mathbb{R}^+)^m$ is a vector of weights and $\overline{\delta}$ is a *complementary Kroenecker delta*: $\overline{\delta}(a,b) = \begin{cases} 0 & \text{if } a = b, \\ 1 & \text{if } a \neq b. \end{cases}$ ($\Delta : (X,Y) \mapsto \Delta(X,Y)$ is a distance function realized as a *weighted average* of the "discrepancies" on attributes.) For any $X \in U$ let $\mathrm{argmin}_{Y \in S} \Delta(X,Y)$ be any element in S that minimizes the map $Y \mapsto \Delta(X,Y)$ on S:

$$Y_0 = \mathrm{argmin}_{Y \in S} \Delta(X,Y) \ \Leftrightarrow \ Y_0 \in S \ \& \ \forall Y \in S : \ \Delta(X,Y_0) \leq \Delta(X,Y).$$

Hence, any new instance X will be classified in the class of $\mathrm{argmin}_{Y \in S} \Delta(X,Y)$, denoted from now on as $\mathrm{Class}(\mathrm{argmin}_{Y \in S} \Delta(X,Y))$.

In fact we may analyze also the context around an unknown feature: Given $X = (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_m)$ and an attribute index $i \in \{1, \ldots, m\}$, a *context of* x_i is a substring c_i of X including x_i: $X = X_1 * c_i * X_2$, for some possibly empty strings X_1, X_2. For any possible value y_i of the i-th attribute let $c_i(y_i)$ be the string obtained from c_i substituting x_i by y_i. For any $Y = (y_1, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_m) \in S$ we shall estimate the probability that y_i appears in the context c_i of x_i. A measure of likeness of X to Y is

$$\overline{\delta}'(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 - \Pr(c_i(y_i) | c_i) & \text{if } x_i \notin D(A_i) \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

MBL Implementation. A natural way to implement MBL is using a *trie*. The IGTree method [2] has the following characteristics:

- It compresses the training instances into a decision tree saving thus both search time and memory space.
- It faces the problem of exact matching failure –a feature value on the new instance which is not contained in any training instance– by using default information on the last non-terminal matching node. The default can be taken from "the most probable class for exact matchings", i.e. the feature values minimizing the gain-ratio values.

Several taggers following this approach have been reported in the literature. In [3] it is reported 97.8% in accuracy when tagging a text of 89×10^3 words using 711×10^3 training instances.

Certainly, MBL is an efficient and simple method quite adequate for natural language processing classification tasks (e.g. [12] [13]). In our approach, we have implemented a modification of IGTree [4]. We made the modification by means of a different distance function (represented by $\overline{\delta}'$), that was able to manage the unknown features into an instance. Now, the IGTree method is slightly modified:

1. If the current instance has an unknown feature, respect to the training set, with "high" gain ratio then using $\overline{\delta}'$ we select the closest class,
2. Else we proceed the classification as in IGTree.

$\overline{\delta}'$ improves the tagging of unknown words, and therefore the global tagging too.

3 Resources

The tagging system uses three types of resources: dictionaries, morphosyntactic rules and context instances. Some resources are corpus-independent, they are composed by conjugation rules, suffix lists and dictionary parts. The other resources are corpus-dependent. The main design criteria look to point the tagger to solve the most difficult problems, e.g. unknown words tagging. With this in mind, first of all it was carried out the tagging of simple words: definite words and words whose taggings were supported by morphosyntactic rules.

3.1 Dictionaries

As we have mentioned our tagger is based on a part of the CEMC. This text collection has 9,224 words, 2,644 distinct signs and an average of 2.02 tags per word. Using the corpus ambiguous, definite or unknown words were identified and put in corresponding dictionaries: DICCD, for frequent definite words[1]; DICCA, for frequent ambiguous words; DICCV, for frequent verbal forms; and DICCS joins the punctuation signs. Furthermore, an auxiliary source[2] to process manually a list of proper names, NOMP, was used. Table 1 lists the used dictionaries.

[1] DICCD was enriched with additional frequent definite words of the corpus with average rank 46.71.

[2] A collection of 127 articles from the Mexican magazine **PROCESO** of 1998.

Table 1. Known words and its appearing percentage in the corpus.

Dictionary	**Size**	**Word type**	**Occurrence** (%)
DICCD	184	Frequent definite	28.38
DICCA	163	Frequent ambiguous	17.59
DICCS	11	Punctuation	11.20
DICCV	835	Frequent verbal form	5.20
NOMP	3,337	Proper noun	2.08
Total	**4,530**		**64.47**

3.2 Morphosyntactic Rules

According to the criteria cited above, it was carried out a general exploration on the corpus in order to determine the tags for some ambiguous frequent words and unknown words using morphosyntactic rules. The main parameter was the probability of occurrence of the word in the context:

> If $\Pr(tag(w) = m|C) = 1$, where C is a predetermined context, then we may conclude $C \Rightarrow (tag(w) = m)$.

Two examples that satisfy the preceding assumption are the following:

$$\Pr\left((m_{i-1}, v_i, m_i) \in \{\texttt{PREP}\} \times loas \times \{\texttt{ART}\}|\, (m_{i-1}, v_i) \in \{\texttt{PREP}\} \times loas\right) = 1$$

$$\Pr\left((v_{i-1}, m_i, v_{i+1}) \in \{el, al\} \times \{\texttt{NOM}\} \times ddel|\, (v_{i-1}, v_{i+1}) \in \{el, al\} \times ddel\right) = 1$$

where $ddel = \{de, del\}$, and $loas = \{la, las, lo, los\}$.

The verbal endings set from the COES spelling system [10] was used. This is a function that maps a conjugation ending into several possible infinitive endings. If the non-ending part of a supposed verbal form concatenated with an obtained ending matches an infinitive verb, then the original word is a right verbal form. A list VERBO of verbs in infinitive form is required and is provided by COES. The noun endings were selected from the inventory contained in [8]. It is represented by TERD and contains the following Spanish endings: "acia", "ad", "amento", "amiento", "ancia", "anda", "ato", "encia", "icia" "idumbre", "ón", "tad", "tura" and "ud". It is assumed that no ending in this list coincides with an ending of a verbal conjugation. Let $\mathcal{M}$ be the set of verbal endings. Let TERC be the intersection of both sets TERD and $\mathcal{M}$. It contains the endings: "ías", "ado", "ido", "ía", "to", "so", "es", "as", "o", "a", "era", "ijo", "iño", "ite" and "uelo", and were collected to identify the ambiguity `NOM/VERB`. Actually, when a word has an ambiguous ending of this type it is provisionally tagged `VERP`: *probable verb*. Besides, a regular expression detects some clitics. Indeed, they are represented in reversal order:

```
^s?[oa]l(son|e[mts])r[\'a\'e\'{\i}]|
^s?[oae]lr[aei]|
^s?([oae]l|[oa]l(e[mts]|son))odn\'a|
^s?([oae]l|[oa]l(e[mts]|son))odn\'ei|
^s?([oae]l|[oa]l(e[mts]|son))odn\'e
```

3.3 Contexts

For ambiguous, unknown and the frequent ambiguity (NOM/VERB) words the method MBL is applied, using the training set from the corpus. The features to be considered should be selected. Fig. 1 shows the gain ratio curve corresponding to 19 endings of length 6 at each side of an ambiguous word and, consequently, to the same number of tags surrounding the ambiguous frequent word. After performing this analysis for each of the three types of words the following features were selected:

Ambiguous frequent words: Two word endings from words which are at each side of the ambiguous word, as well as the tags of those words.

Ambiguity NOM/VERB: Two word endings from words which are at each side of the word with ambiguity NOM/VERB, as well as the tags of those words.

Unknown words: Two word endings immediate before the unknown word as well as the tags of those words.

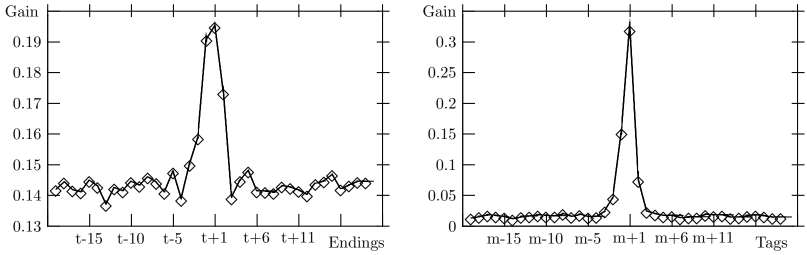


Fig. 1. Gain ratio curve for 19 elements around an ambiguous word.

Let INSTA, INSTN and INSTD be the training sets for ambiguous, NOM/VERB resolution and unknown words, respectively. The required resources as a whole were:

1. Dictionaries
2. Morphosyntactic rules
3. Verb conjugation rules
4. Instances of ambiguous frequent words
5. Instances of NOM/VERB ambiguity
6. Instances of unknown words

The tagging steps followed the same order as listed above, see fig. 2.

4 Tagging Example

Figure 3 contains a text[3] used as a test for the tagger. Table 4 is the final result of the tagging. There is a row of text followed by a row of tags and an index (#) to be used as a reference. Finally, table 2 summarizes the accuracy of each step applied to the example text, making reference to the resource used.

[3] This is a part of the news written by Carlos Acosta Córdova and Guillermo Correa, published by **PROCESO**, May 1999.

```
Let inflex : $\mathcal{M} \to 2^{\mathcal{M}}$ be the function that gives a set of endings from an
initial ending taken from a supposed verbal form, and let $T$ be an untagged
text. On each step we setup $T$:

step 1 For each member $w_i$ of $T$:
    If $w_i$ is in DICCD, DICCV, DICCS, or NOMP, then define its tag $m_i$
    as the dictionary that contains $w_i$ indicates.
step 2 For each non tagged member $w_i$ of $T$:
    If any morphosyntactic rule is able to be applied to a context of $w_i$, then
    define $m_i$ according to that rule.
step 3 For each non tagged member $w_i$ of $T$:
    If there exists $k$ such that $w_{i\_k} \in$ TERC, then $m_i = \texttt{VERP}$.
    Else if there exists $k$ such that $w_{i\_k} \in \mathcal{M}$, $w_i = xw_{i\_k}$, and for some
    $y \in \text{inflex}(w_{i\_k})$ it holds $xy \in$ VERBO, then $m_i = \texttt{VERB}$.
step 4 For each non tagged member $w_i$ of $T$, and member of DICCA:
    Let $X$ be the context of $w_i$. Use INSTA to define $m_i =$
    $\text{Class}(\text{argmin}_{Y \in S} \Delta(X, Y))$.
step 5 For each member $w_i$ of $T$ tagged with VERP:
    Use INSTN and apply IGTree to define $m_i$.
step 6 For each non tagged member $w_i$ of $T$:
    Use INSTD an apply IGTree with $\bar{\delta}'$ to define $m_i$.
```

Fig. 2. Main tagging steps.

y el beneficiario directo de todo ello será , sin duda , el PRI , según reconoció el secretario de hacienda ante analistas , académicos e inversionistas en el consejo de las américas de Nueva York , horas antes de participar en la reunión de los organismos financieros internacionales . dijo , sin ambages , que si la economía sigue bien y se mantiene la disciplina , el partido revolucionario institucional tendrá buenas posibilidades no sólo en la contienda por la presidencia , sino en la elección del gobierno capitalino y , aun , en recuperar la mayoría absoluta en la cámara de diputados . ... en efecto , en su examen anual de la economía mexicana , que hizo público el jueves 29 , la OCDE - el club de los 29 países más ricos del mundo , al que México ingresó en 1994 - admite que el desempeño económico del país fue positivo en los últimos tres años , pero señala una serie de ineficiencias en la política económica - inestabilidad presupuestal , dependencia del petróleo y deficiente sistema tributario - que impiden sacar al país del subdesarrollo y contrarrestar los niveles de pobreza extrema .

Fig. 3. Example text.

5 Performance

With a text of 9,000 words a test was carried out. The input text was divided into ten parts. When processing each part, the training text was increased by adding the former part. This is represented in the x-axis of the graphs in fig. 5. The experiments results are shown in the graphs. The y-axis represents the accuracy, and was calculated as the number of right taggings divided by the number of text words. The first graph contains the average of accuracy and the minimum and maximum of the tagging process in an error bar graph. The second graph compares the average accuracy using $\bar{\delta}$ and $\bar{\delta}'$. In both graphs the performance is shown as the training corpus grows.

#	Text/Tags
1	y el beneficiario directo de todo ello será , sin `CONJ ART NOM ADJ PREP ADJ PRON VERB COM PREP`
11	duda , el PRI , según reconoció el secretario de `NOM COM ART NP COM PREP VERB ART NOM PREP`
21	hacienda ante analistas , académicos e inversionistas en el consejo `VERB PREP NOM COM NOM CONJ ADJ PREP ART NOM`
31	de las américas de Nueva York , horas antes de `PREP ART NOM PREP NP NP COM NOM ADV PREP`
41	participar en la reunión de los organismos financieros internacionales . `VERB PREP ART NOM PREP ART NOM ADJ VERB PTO`
51	dijo , sin ambages , que si la economía sigue `VERB COM PREP NOM COM CONJ CONJ ART NOM VERB`
61	bien y se mantiene la disciplina , el partido revolucionario `ADV CONJ PRON VERB ART NOM COM ART NOM ADJ`
71	institucional tendrá buenas posibilidades no sólo en la contienda por `ADJ VERB NP NOM ADV ADV PREP PRON VERB PREP`
81	la presidencia , sino en la elección del gobierno capitalino `ART NOM COM CONJ PREP ART NOM CONT NOM ADJ`
91	y , aun , en recuperar la mayoría absoluta en `CONJ COM NOM COM PREP VERB ART NOM ADJ PREP`
101	la cámara de diputados . / en efecto , en `ART NOM PREP NOM PTO SUS PREP NOM COM PREP`
111	su examen anual de la economía mexicana , que hizo `ADJ NOM ADJ PREP ART NOM ADJ COM PRON VERB`
121	público el jueves 29 , la OCDE - el club `NOM ART NOM NUM COM ART NP GUI ART NOM`
131	de los 29 países más ricos del mundo , al `PREP ART NUM NOM ADJ NOM CONT NOM COM CONT`
141	que México ingresó en 1994 - admite que el desempeño `CONJ NP VERB PREP NUM GUI VERB CONJ ART NOM`
151	económico del país fue positivo en los últimos tres años `ADJ CONT NOM VERB VERB PREP ART NOM ADJ NOM`
161	, pero señala una serie de ineficiencias en la política `COM CONJ VERB ART VERB PREP NOM PREP ART NOM`
171	económica - inestabilidad presupuestal , dependencia del petróleo y deficiente `ADJ GUI NOM ADJ COM NOM CONT NOM CONJ NOM`
181	sistema tributario - que impiden sacar al país del subdesarrollo `ADJ ADJ GUI CONJ VERB VERB CONT NOM CONT NOM`
191	y contrarrestar los niveles de pobreza extrema . `CONJ VERB ART NOM PREP NOM VERB PTO`

Fig. 4. Example text with tags.

Table 2. Accuracy from each step of the example text tagging.

Step	# tags	# right tags	%
Dictionaries	103	102	99.02
Morphological	17	17	100.00
Verbal forms	13	10	76.92
Frequent ambiguous	16	14	87.50
`VERB/NOM`	9	7	77.77
Unknown words	40	34	85.00
Total	**198**	**184**	**92.92**

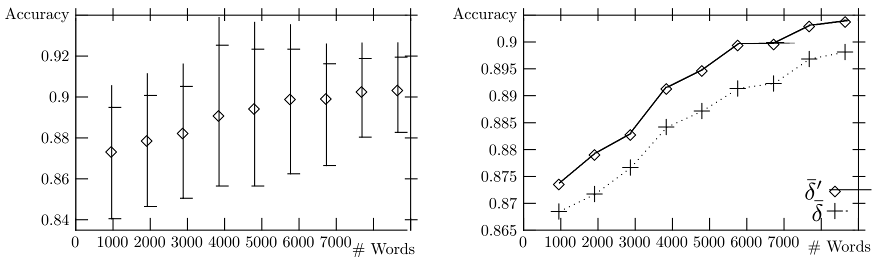


Fig. 5. Performance of the tagger.

6 Conclusions

A part-of-speech tagger for Spanish language based on small resources and minimum programming effort has been built. Such conditions may be available to develop a new tagger and speed up the initial stage.

The tagging accuracy of SEPE is greater than 0.9. Of course, the behavior of our tagger can be improved. It is remarkable that the low verbs tagging accuracy and the low `VERB/NOM` ambiguity resolution accuracy at the test text can be increased by updating the TERC list. The word "*serie*" at position 165 on the example text is tagged as `VERB` by the `VERB/NOM` ambiguity solving procedure. However, this procedure only uses 515 instances (the least of the three training sets). Therefore, at the next stage we are considering:

- To increase the number of known words.
- To make use of derivational and inflectional lists to cope with partially known words. This should support the previous point.
- To increase the corpus to train the learning method as well as to grow the corpus valid-rules.
- To carry out performance test to compare to other methods. This requires to change the tag set and the corpus.

In spite of the fact that there is a small difference between $\bar{\delta}$ and $\bar{\delta}'$ we expect a greater improvement with a greater corpus. Further, a greater corpus might help to debug the ambiguity grammem lists as well as the morphosyntactic rules.

References

1. Daelemans, Walter: Memory-based lexical acquisition and processing, *Lecture Notes in Artificial Intelligence*, 898, Springer Verlag, pp 85-98, 1995.
2. Daelemans, Walter; Durieux, Gert & van-den-Bosch, Antal: Towards inductive lexicon, *Proc. of LREC Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications*, Granada, http://ilk.kub.nl/, 1998.
3. Daelemans, Walter; van-den-Bosch, Antal; Zavrel, Jakub; Veenstra, Jorn; Buchholz, Sabine & Busser, Bertjan: Rapid development of NLP modules with memory-based learning, *Proc. of ELSNET in Wonderland*, pp 105-113, 1998.

4. Jiménez-Salazar, Héctor & Morales-Luna, Guillermo: Instance metrics improvement by probabilistic support, *Lecture Notes in Artificial Intelligence*, 1793, Springer Verlag, pp 699-705, 2000.
5. Lara, Luis Fernando; Ham-Chande, Roberto & García-Hidalgo, Ma. Isabel: *Investigaciones lingüísticas en lexicografía*, Jornadas 89, El Colegio de México, 1979.
6. Màrquez, Lluís & Rodríquez, Horacio: Part-of-speech tagging using decision trees, *Lecture Notes in Artificial Intelligence*, 1398, pp 25-33, 1998.
7. Marques, N. & Pereira, G.: A POS-tagger generator for unknown languages, *Procesamiento del Lenguaje Natural*, Rev. No. 27, SEPLN, pp 199-206, Spain, 2001.
8. Moreno de Alba, Jose G.: *Morfología derivativa nominal en el español de México*, National University of Mexico (UNAM), Mexico 1986.
9. Pla, F.; Molina, A. & Prieto N.: Evaluación de un etiquetador morfosintáctico basado en bigramas especializados para el castellano, *Procesamiento del Lenguaje Natural*, Rev. No. 27, SEPLN, pp 215-221, Spain, 2001.
10. Rodríguez, Santiago & Carretero, Jesús: Building a Spanish speller, `http://www.datsi.fi.upm.es`, 1997.
11. Ruiz, L.: *Desarrollo de un modelo computacional para el procesamiento de corpus textuales basado en la etiquetación automática*, Ph. D. dissertation, Universidad de Oriente, Cuba, 2001.
12. van-den Bosch, Antal; Daelemans, Walter; Weijters, Ton: Morphological analysis as classification: an inductive-learning approach, `http://lib-www.lanl.gov/cmp-lg/9607021`, 1996.
13. Zavrel, Jakub; Daelemans, Walter; Veenstra, Jorn: Resolving PP-attachment ambiguities with MBL, *CoNLL*, `http://ilk.kub.nl/`, 1997.

Fuzzy Set Tagging

Dariusz J. Kogut

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
Dariusz.Kogut@cern.ch

Abstract. This paper presents a fuzzy set approach to the rule-based tagging. Both lexical and contextual phases have been shortly discussed to point the potential advantages of using an uncertain part-of-speech information. Obtained results are comparable with the results of other taggers, for example, Brill tagger.

1 Introduction

The appeal of fantasizing about intelligent computers that understand human communication is practically unavoidable. However, the natural language research seems to be one of the hardest problems of artificial intelligence due to complexity, irregularity and diversity of human languages. In this paper an overview of syntactic analysis based on fuzzy set tagging will be presented and hopefully provide some insight for further inquiry.

2 Syntax Analysis – A Fuzzy Set Approach

A fuzzy set syntactic analysis tries to combine a statistical and a rule-based approach [13]. The processing itself is divided into two phases: (1) forming the preliminary fuzzy set membership functions (lexical tagging), (2) verification based on a collection of grammatical principles (contextual tagging). Each word of given sentence is eventually assigned to its part-of-speech category in a sort of fuzzy set tagging procedure.

2.1 Preliminary Membership Functions

The most cases of ambiguity concern four basic part-of-speech categories (nouns, verbs, adjectives and adverbs). Thus, during the first phase of fuzzy set syntax analysis four descrete membership functions (designated as Ψ_N, Ψ_V, Ψ_{Adj} and Ψ_{Adv}) are being constructed on a base of given sentence. Lexicon (like WordNet [11] used in this prototype) stores the basic linguistic information and therefore plays an important role in the whole preliminary phase of syntax analysis. It provides the necessary data on word's senses and categories, so the preliminary functions may be formed.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 260–263, 2002.

Let us assume that:

$\boldsymbol{w}_{(n)}$ – refers to the n^{th}-word of the sentence;
L – refers to the amount of categories that a certain word belongs to;
n_x – refers to the amount of word's senses in the scope of x category, where $x \in \mathcal{M}$, and $\mathcal{M} = \{N, V, Adj, Adv\}$;
N – refers to the amount of word's senses in total (across all categories):

$$N[\boldsymbol{w}_{(n)}] = \sum_{y \in \mathcal{M}} n_y[\boldsymbol{w}_{(n)}]$$

ξ – decides on the importance of categories and senses ($0 \leq \xi \leq 1$, but the tests have shown that ξ close to 0.5 gives the better tagging);

The Ψ_x membership function is defined by:

$$\Psi_x[\boldsymbol{w}_{(n)}] = \xi \cdot \frac{n_x[\boldsymbol{w}_{(n)}]}{N[\boldsymbol{w}_{(n)}]} + (1 - \xi) \cdot \frac{1}{L}$$

In order to illustrate the algorithm, let us take an example: Why_0 do_1 the_2 $people_3$ $drive_4$ on_5 the_6 $right_7$ $side_8$ of_9 the_{10} $road_{11}$? Figure 1 describes a draft of both Ψ_N and Ψ_V functions, where X axe refers to the words' order, and Y axe to the degree of membership.

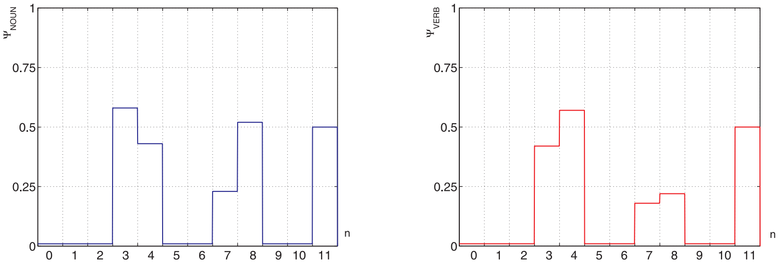


Fig. 1. Ψ_N and Ψ_V membership functions

Certainly, each word must be considered as a specific part-of-speech within the given context. Thus, the proper category must be assigned in a disambiguation process. One of the disambiguation formulas is defined as follows:

$$\boldsymbol{w}_{(n)} \in \mathcal{K}_N \Leftrightarrow \Psi_N[\boldsymbol{w}_{(n)}] \geq \max(\Psi_{Adj}[\boldsymbol{w}_{(n)}], \Psi_V[\boldsymbol{w}_{(n)}], \Psi_{Adv}[\boldsymbol{w}_{(n)}])$$

Unfortunately, the syntactic disambiguation based on the lexicon data only cannot assure the proper tagging. Therefore, an additional processing based on grammatical principles seems to be necessary.

2.2 English Grammar Engine

This fuzzy set tagging model employs its own English grammar engine [10] with a set of simple grammar rules that can verify and approve the Ψ_N, Ψ_V, Ψ_{Adj} and

Table 1. The tagger's accuracy

Gutenberg Project's E-Text Books	$Acc_{lexicon}$	$Acc_{lex+gram}$
Cromwell by William Shakespeare	58%	84%
J.F. Kennedy's Inaugural Address (Jan 20, 1961)	67%	92%
The Poetics by Aristotle	61%	86%
An Account of the Antarctic Expedition by R. Amundsen	63%	89%
Andersen's Fairy Tales by H.Ch. Andersen	70%	94%

Ψ_{Adv} functions. The engine utilises *a procedural parsing* [7], and its grammatical principles has been based on Word Grammar [9]. Thus, the structure of sentence is being described in terms of dependencies between the words, and there is no need to identify any phrases[1] [12]. These dependencies always refer to the word's position in a sentence, therefore they cannot be used to analyse a non-positional language. The basic grammar rules may follow the examples below (let ξ be a degree of dependency between words, $0 \leq \xi \leq 1$):

A. Rule: if n^{th}*-word* is an adjective, thus the $(n+1)^{th}$ *-word* may be a noun or an adjective ($\xi_{A,Adj.Adj} = 0.25$; $\xi_{A,Adj.N} = 0.55$, based on results of a simple analysis of word dependencies in the corpus).

$$\boldsymbol{w}_{(n)} \in \mathcal{K}_{Adj} \Rightarrow \Psi'_{Adj}[\boldsymbol{w}_{(n+1)}] = min((\Psi_{Adj}[\boldsymbol{w}_{(n+1)}] + \xi_{A,Adj.Adj}), 1)$$
$$\boldsymbol{w}_{(n)} \in \mathcal{K}_{Adj} \Rightarrow \Psi'_{N}[\boldsymbol{w}_{(n+1)}] = min((\Psi_{N}[\boldsymbol{w}_{(n+1)}] + \xi_{A,Adj.N}), 1)$$

B. Rule: if n^{th}*-word* is a determiner (belongs to $\mathcal{K}_{Dtrm}$), then $(n+1)^{th}$*-word* could be a noun or an adjective. Moreover, $(n+1)^{th}$*-word* should not belong to the category of verbs ($\xi_{B,Dtrm.Adj} = 0.4$; $\xi_{B,Dtrm.N} = 0.45$; $\xi_{B,Dtrm.V} = 0.6$).

$$\boldsymbol{w}_{(n)} \in \mathcal{K}_{Dtrm} \Rightarrow \Psi'_{Adj}[\boldsymbol{w}_{(n+1)}] = min((\Psi_{Adj}[\boldsymbol{w}_{(n+1)}] + \xi_{B,Dtrm.Adj}), 1)$$
$$\boldsymbol{w}_{(n)} \in \mathcal{K}_{Dtrm} \Rightarrow \Psi'_{N}[\boldsymbol{w}_{(n+1)}] = min((\Psi_{N}[\boldsymbol{w}_{(n+1)}] + \xi_{B,Dtrm.N}), 1)$$
$$\boldsymbol{w}_{(n)} \in \mathcal{K}_{Dtrm} \Rightarrow \Psi'_{V}[\boldsymbol{w}_{(n+1)}] = max((\Psi_{V}[\boldsymbol{w}_{(n+1)}] - \xi_{B,Dtrm.V}), 0)$$

2.3 Empirical Evaluation

A set of tests based on 2000-word samples of e-text books [8] has been done to evaluate this fuzzy set syntactic analysis. The results (in table 1) describe the processing precision ($Acc_{lexicon}$ for the preliminary analysis based on the lexicon data only, and $Acc_{lex+gram}$ for the complete processing). The tests show that the accuracy rate depends on the type of text. The grammar engine significantly improves the score, but as long as a small set of rules is considered (27 at this stage of research) the current results are inferior to the Brill tagger (where, at the begining, 71 *patches* were used to gain a 95% accuracy rate [6]). Therefore, a set of additional grammar rules [2] [3] [5] will be considered in further research. The fuzzy set approach seems to be promising. It helps to handle the uncertain part-of-speech information that may result in better (than in Brill rule-based tagger) accuracy in both lexical and contextual phases. Moreover, this approach might be applied to other well-known taggers to achieve a greater quality [4].

For the time being, this fuzzy set tagging model has been successfully implemented and tested as a part of Natural Language Processor for Web Search (TORCH project) in European Organization for Nuclear Research, in Geneva.

Acknowledgments

The author wish to thank the many individuals who have contributed to this paper through generous sharing of their time and insights. TORCH would not have been possible without the cooperative effort and spirit kind people all over the world. Special thanks to Maria Dimou (CERN), professor Henryk Rybinski (Warsaw University of Technology), Robert Cailliau (CERN), Michael Naumann (Infoseek) and professor Alexander Gelbukh.

References

1. Borsley, R.: Syntactic theory. A unified approach. Arnold Publishing Company, London (1990)
2. Brill, E.: Pattern-Based Disambiguation for Natural Language Processing. EMNLP/VLC, (2000)
3. Brill, E.: Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. In Natural Language Processing Using Very Large Corpora, (1999)
4. Brill, E., Wu, J.: Classifier Combination For Improved Lexical Disambiguation. COLING/ACL, (1998)
5. Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Procesing: A Case Study in Part of Speech Tagging. Computational Linguistics, December, (1995)
6. Brill, E.: A simple rule-based part of speech tagger. Third Conference on Applied Natural Language Processing, Trento, Italy, (1992)
7. Dobryjanowicz, E.: Wybrane metody analizy skladniowej. Akademicka Oficyna Wydawnicza RM, Warsaw (1992)
8. Gutenberg Project. E-Text Books. http://promo.net/pg/
9. Hudson, R.: English Word Grammar. Blackwell, (1990)
10. Kogut, D.: Fuzzy Set Approach to Syntactic Analysis - TORCH project http://home.elka.pw.edu.pl/∼dkogut/torch
11. Princeton University: WordNet Dictionary. http://www.cogsci.princeton.edu
12. Riemsdijk, H., Williams, E.: Introduction to the theory of grammar. MIT Press, Cambridge (1986)
13. Roche, E., Schabes, Y.: Finite-state language processing. MIT Press, Cambridge (1997)

Towards a Standard for a Multilingual Lexical Entry: The EAGLES/ISLE Initiative

Nicoletta Calzolari[1], Antonio Zampolli[1,2], and Alessandro Lenci[2]

[1] Istituto di Linguistica Computazionale del CNR, Pisa, Italy
[2] Dipartimento di Linguistica, Università di Pisa, Pisa, Italy
{glottolo,eagles,lenci}@ilc.pi.cnr.it
www.ilc.pi/comp/lncs/index.html

Abstract. ISLE, a transatlantic standard oriented initiative supported by EC and NSF under the Human Language Technology (HLT) programme, is a continuation of the long standing EAGLES initiative. The objective is to develop widely agreed and urgently demanded standards and guidelines for infrastructural language resources, tools, and HLT products. ISLE targets the areas of multilingual computational lexicons, natural interaction and multimodality, and evaluation. We describe the preliminary guidelines of a standard framework for multilingual computational lexicons, based on a general schema for the “Multilingual ISLE Lexical Entry” (MILE). The needs and features of existing Machine Translation systems provide the main reference points for the process of consensual definition of the MILE. We also provide a brief description of the EU SIMPLE semantic lexicons, built on the basis of previous EAGLES recommendations and now enlarged to real-size lexicons within national projects, thus creating a large infrastructural platform of harmonised lexicons in Europe. EAGLES previous results have already become *de facto* widely adopted standards, and EAGLES itself is a well-known trademark and point of reference for HLT projects and products.

1 The EAGLES/ISLE Initiative

ISLE (*International Standards for Language Engineering*) is a transatlantic standards oriented initiative under the Human Language Technology (HLT) programme within the EU-US International Research Co-operation. It is a continuation of the long standing European EAGLES (*Expert Advisory Group for Language Engineering Standards)* initiative [5], carried out through a number of subsequent projects funded by the European Commission (EC) since 1993.

The objective is to support HLT R&D international and national projects, and industry by developing and promoting widely agreed and urgently demanded HLT standards and guidelines for infrastructural language resources [21] [3], tools that exploit them, and language engineering products. The aim of ISLE is thus to accelerate the provision of standards, common guidelines, best practice recommendations for:

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 264–279, 2002.

- very large-scale language resources (such as text corpora, computational lexicons, speech corpora [8], multimodal resources);
- means of manipulating such knowledge, via computational linguistic formalisms, mark-up languages and various software tools;
- means of assessing and evaluating resources, tools and products [6].

Leading industrial and academic players in the HLT field (more than 150) have actively participated in the definition of this initiative and have lent invaluable support to its execution. It is important to note that the work of EAGLES[1] must be seen in a long-term perspective. Moreover, successful standards are those which respond to commonly perceived needs or aid in overcoming common problems. In terms of offering workable, compromise solutions, they must be based on some solid platform of accepted facts and acceptable practices. EAGLES was set up to determine which aspects of our field are open to short-term *de facto* standardisation and to encourage the development of such standards for the benefit of consumers and producers of language technology, through bringing together representatives of major collaborative European R&D projects, and of HLT industry, in relevant areas. This work is being conducted with a view to providing the foundation for any future recommendations for International Standards that may be formulated under the aegis of ISO.

The current ISLE project[2] targets the three areas of:

- *multilingual computational lexicons*[3],
- *natural interaction and multimodality* (*NIMM*)[4],
- *evaluation of HLT systems*[5].

For *multilingual computational lexicons*, ISLE aims at: extending EAGLES work on lexical semantics, necessary to establish inter-language links; designing and proposing standards for multilingual lexicons; developing a prototype tool to implement lexicon guidelines and standards; creating exemplary EAGLES-conformant sample lexicons and tagging exemplary corpora for validation purposes; and developing standardised evaluation procedures for lexicons.

For *NIMM*, a rapidly innovating domain urgently requiring early standardisation, ISLE work is targeted to develop guidelines for: the creation of NIMM data resources; interpretative annotation of NIMM data, including spoken dialogue in NIMM contexts; metadata descriptions for large NIMM resources; and annotation of discourse phenomena. For *evaluation*, ISLE is working on: quality models for machine translation systems; and maintenance of previous guidelines - in an ISO based framework (ISO 9126, ISO 14598).

Three Working Groups, and their sub-groups, carry out the work, according to the already proven EAGLES methodology, with experts from both the EU and US. Inter-

[1] See EAGLES guidelines, http://www.ilc.pi.cnr.it/ EAGLES96/home.html

[2] Coordinated by A. Zampolli for EU and M. Palmer for US, see http://www.ilc.pi.cnr.it/ EAGLES96/isle/ISLE_Home_Page.htm.

[3] EU chair: N. Calzolari; US chairs: M. Palmer and R. Grishman.

[4] EU chair: N. O. Bernsen; US chair: M. Liberman.

[5] EU chair: M. King; US chair: E. Hovy.

national workshops are used as a means of achieving consensus and advancing work. Results are widely disseminated, after due validation in collaboration with EU and US HLT R&D projects, National projects, and industry.

In the following we concentrate on the Computational Lexicon Working Group (CLWG), and its goal of establishing a general and consensual standardized environment for the development and integration of multilingual resources. The general vision adheres to the idea of enhancing the sharing and reusability of multilingual lexical resources, by promoting the definition of a common parlance for the community of multilingual HLT and computational lexicon developers. The way the CLWG pursues this goal is by proposing a general schema for the encoding of multilingual lexical information, the *MILE* (Multilingual ISLE Lexical Entry). This has to be intended as a meta-entry, acting as a common representational layer for multilingual lexical resources.

We briefly present the result of the first phase of activities of the CLWG, dedicated to the elaboration of a survey of existing multilingual resources both in the European, American and (although still in a more limited extension) Asian research and industrial scenarios. Such a review is also the basis for the process of standard definition, which is the focus of the second ongoing phase, aiming at individuating hot areas in the domain of multilingual lexical resources, which call – and *de facto* can access to – a process of standardisation. We describe the preliminary proposals of guidelines for the MILE, highlighting some methodological principles applied in previous EAGLES, now followed in defining the MILE. We also provide a brief description of the EU SIMPLE semantic lexicons built on the basis of previous EAGLES recommendations.

2 The Computational Lexicon Working Group

2.1 EAGLES Methodology

The basic idea behind EAGLES work is for the group to act as a catalyst in order to pool concrete results coming from major international/national/industrial projects. Relevant common practices or upcoming standards are being used where appropriate as input to EAGLES/ISLE work. Numerous theories, approaches, and systems are being taken into account as any recommendation for harmonisation must take into account the needs and nature of the different major contemporary approaches. The major efforts in EAGLES concentrate on the following types of activities:

- detecting areas ripe for short-term standardisation vs. areas still in need of basic research and development;
- assessing and discovering areas where there is a consensus across existing linguistic resources, formalisms and common practices;
- surveying and assessing available proposals or specifications to evaluate the potential for harmonisation and convergence and for emergence of standards;

- proposing common specifications for core sets of basic phenomena and recommendations for good practice on which a consensus can be found;
- setting up guidelines for representation of core sets of basic features, for representation of resources, etc.;
- feasibility studies for less mature areas;
- suggesting actions to be taken for a stepwise procedure leading to the creation of multilingual reusable resources, elaboration of evaluation methodologies, etc.

2.2 Standards Design and the Interaction with R&D

Existing EAGLES results in the Lexicon and Corpus areas are currently adopted by an impressive number of European - and recently also National - projects, thus becoming "the *de-facto* standard" for LR in Europe. This is a very good measure of the impact – and of the need – of such a standardisation initiative in the HLT sector. To mention just a few key examples:

- the LE PAROLE/SIMPLE resources (morphological/syntactic/semantic lexicons and corpora for 12 EU languages [20][15][11][1]) rely on EAGLES results [16][17], and are now being enlarged to real-size lexicons through many National Projects, thus building a really large infrastructural platform of harmonised lexicons in Europe, sharing the same model;
- the ELRA Validation Manuals for Lexicons [19] and Corpora [2] are based on EAGLES guidelines;
- morpho-syntactic encoding of lexicons and tagging of corpora in a very large number of EU, international and national projects—and for more than 20 languages— is conformant to EAGLES recommendations [12][13][10].

Standards must emerge from state-of-the-art developments. The process of standardisation, although by its own nature not intrinsically innovative, must – and actually does – proceed shoulder to shoulder with the most advanced research. Since ISLE involves many bodies active in EU-US NLP and speech projects, close collaboration with these projects is assured and, significantly, free manpower has been contributed by the projects, as a sign of both their commitment and of the crucial importance they place on reusability issues. As an example, the current NSF project XMELLT on multi-words for multilingual lexicons provides valuable input to ISLE.

Lexical semantics has always represented a sort of *wild frontier* in the investigation of natural language. In fact, the number of open issues in lexical semantics both on the representational, architectural and content level might induce an actually unjustified negative attitude towards the possibility of designing standards in this difficult territory. Rather to the contrary, standardisation must be conceived as enucleating and singling out the areas in the open field of lexical semantics, that already present themselves with a clear and high degree of stability, although this is often hidden behind a number of formal differences or representational variants, that prevent the possibility of exploiting and enhancing the aspects of commonality and the already consolidated achievements. With no intent of imposing any constraints on investigation and experimentation, the ISLE CLWG rather aims at selecting mature areas and results in computational lexical semantics and in multilingual lexicons, which can also be regarded as stabilised achievements, thus to be used as the basis for future research. Therefore, consolidation of a standards proposal must be viewed, by necessity, as a slow process comprising, after the phase of putting forward proposals, a cyclical phase involving ISLE external groups and projects with:

- careful evaluation and testing of recommendations in concrete applications;
- application, if appropriate, to a large number of European languages;
- feedback on and readjustment of the proposals until a stable platform is reached;
- dissemination and promotion of consensual recommendations.

The process of standard definition undertaken by CLWG represents an essential interface between advanced research in the field of multilingual lexical semantics, and the practical task of developing resources for HLT systems and applications. It is through this interface that the crucial trade-off between research practice and applicative needs will actually be achieved.

3 The ISLE Survey Phase and Recommendation Phase

3.1 The Survey Phase

Following the well established EAGLES methodology, the first priority was to do a wide-range survey of bilingual/multilingual (or semantic monolingual) lexicons, so as to reach a fair level of coverage of existing lexical resources. This is a preliminary and yet crucial step towards the main goal of the current CLWG, i.e. the definition of the "*Multilingual ISLE Lexical Entry*" (*MILE*). This is the main focus of the so called "recommendation phase", whose aim is to propose consensual Recommendations/Guidelines. With respect to this target, one of the first objectives is to discover and list the (maximal) set of (granular) *basic notions* needed to describe the multilingual level. The *Survey* of existing lexicons [4] has been accompanied by the analysis of the requirements of a few multilingual applications, and by the parallel analysis of typical cross-lingually complex phenomena. [6] Both these aspects provide the general scenarios in terms of which the survey has been organised, as well as form the reference landmarks for the propositive phase of standard design.

One of the crucial aspects for HLT is how to optimise the production, maintenance and extension of computational lexical resources, as well as the process leading to their integration in applications. An essential precondition to achieve these results is to establish a common and standardized framework for computational lexicon construction. This is even more true when multilingual lexicons is taken into consideration. Here two specific problems arise, which respectively concern *architectural* and

[6] Contributors are: Atkins, Bel, Bertagna, Bouillon, Calzolari, Dorr, Fellbaum, Grishman, Habash, Lange, Lehmann, Lenci, McCormick, McNaught, Ogonowski, Palmer, Pentheroudakis, Richardson, Thurmair, Vanderwende, Villegas, Vossen, Zampolli.

representational issues: (i.) how to build new bilingual (multilingual) lexicons from available monolingual resources; (ii.) how to state in the most proper way the translation correspondences among entries in the multilingual lexicon. With respect to the latter problem, the passage from source language (SL) to target language (TL) makes it necessary to express very complex and articulated transfer conditions, which have to take into account as difficult and pervasive phenomena as argument switching, multi-word expressions, collocational patterns, etc. In turn, the representational issues are crucially connected to the architectural ones, mainly depending on how linguistic information is organized in the monolingual parts, and how it can be accessed at the multilingual layer.

The function of an entry in a multilingual lexicon is to supply enough information to allow the system to identify a distinct sense of a word or phrase in SL, in many different contexts, and reliably associate each context with the most appropriate translation TL. The first step is to determine, of all the information that can be associated with SL lexical entries, what is the most relevant to a particular task. We decided to focus the work of survey and subsequent recommendations around two major broad categories of application: Machine Translation and Cross-Language Information Retrieval. They have partially different/complementary needs, and can be considered to represent the requirements of other application types. It is necessary in fact to ensure that any guidelines meet the requirements of industrial applications and that they are implementable.

A *grid for lexicon description* was prepared to classify the content and structure of the surveyed resources on the basis of a number of agreed parameters of description. This grid has been used to evaluate how the various types of information can be relevant to solve problems usually tackled when processing language in a bilingual or multilingual environment.

3.2 The Recommendation Phase: Focus on Semantic and Collocational Issues

The principle guiding the elicitation and proposal of MILE basic notions in the recommendation phase is, according to a previous EAGLES methodology, the so-called *'edited union'* (term put forward by Gerald Gazdar) of what exists in major lexicons/models/dictionaries, enriched with types of information usually not handled (e.g. those of collocational/syntagmatic nature), to be integrated in a unitary MILE. This method of work has proven useful in the process of reaching consensual *de facto* standards in a bottom-up approach and is at the basis also of ISLE work.

The growing need for dealing with semantics and contents in HLT applications is pushing towards more powerful and robust semantic components. Within the last decade, the availability of robust tools for language analysis has provided an opportunity for using semantic information to improve the performance of applications such as Machine Translation, Information Retrieval, Information Extraction and Summarisation. As this trend consolidates, the need of a protocol which helps normalise and structure the semantic information needed for the creation of reusable lexical resources within the applications of focus, and in a multilingual context, becomes more pressing. Times are thus mature to start tackling the question of how to formulate guidelines for multilingual lexical (semantic) standards.

Sense distinctions are especially important for multilingual lexicons, since it is at this level that cross-language links need to be established. The same is true of syntagmatic/collocational/contextual information. To these areas we are paying particular attention in the recommendation phase, and we are currently examining the extension of the EAGLES guidelines in these and other areas to propose a broad format for multilingual lexical entries which should be of general utility to the community.

In the previous EAGLES work on Lexical Semantics [17] the following technologies were surveyed to determine which types of semantic information were most relevant:

1. Machine Translation (MT)
2. Information Extraction (IE)
3. Information Retrieval (IR)
4. Summarisation (SUM)
5. Natural Language Generation (Gen)
6. Word Clustering (Word Clust)
7. Multiword Recognition + Extraction (MWR)
8. Word Sense Disambiguation (WSD)
9. Proper Noun Recognition (PNR)
10. Parsing (Par)
11. Coreference (Coref)

The results of the previous EAGLES survey are summarized below (each different type of semantic information is followed by the application type in which it figures):

- BASE CONCEPTS, HYPONYMY, SYNONYMY: all applications and enabling technologies
- SEMANTIC FRAMES: MT, IR, IE, & Gen, Par, MWR, WSD, Coref
- COOCCURRENCE RELATIONS: MT, Gen, Word Clust, WSD, Par
- MERONYMY: MT, IR, IE & Gen, PNR
- ANTONYMY: Gen, Word Clust, WSD
- SUBJECT DOMAIN: MT, SUM, Gen, MWR, WSD
- ACTIONALITY: MT, IE, Gen, Par
- QUANTIFICATION: MT, Gen, Coref

It is important to notice that all these semantic information types (except for quantification) are covered by the SIMPLE model. For this reason, the structure and the characteristics of SIMPLE (as a lexical resource designed on the basis of the EAGLES recommendations) has a crucial place in the design of the MILE. One very interesting possibility seems to be to complement WordNet-style lexicons with the SIMPLE design, thereby trying to get at a more comprehensive and coherent architecture for the development of semantic lexical resources.

4 The SIMPLE Lexicons

Given the fact that the PAROLE/SIMPLE Lexicons, based on the GENELEX model [7], are used and critically evaluated as a basis for the definition of the MILE, we briefly provide here some information about these resources. The design of the

SIMPLE lexicons [1] complies with the EAGLES Lexicon/Semantics Working Group guidelines [17], and the set of recommended semantic notions.

The SIMPLE lexicons (see www.ub.es/gilcub/SIMPLE/simple.html for the specifications and sample lexical entries for the various languages) are built as a new layer connected to the PAROLE syntactic layer, and encode structured "semantic types" and semantic (subcategorization) frames. They cover 12 languages (Catalan, Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish, Swedish). The common model is designed to facilitate future cross-language linking: they share the same *core ontology* and the same set of *semantic templates*. The SIMPLE model provides the formal specification for the representation and encoding of the following information:

(i) *semantic type*, corresponding to the template each Semantic Unit (*SemU*) instantiates;
(ii) *domain* information;
(iii) *lexicographic gloss*;
(iv) *argument structure* for predicative SemUs;
(v) *selectional restrictions/preferences* on the arguments;
(vi) *event type*, to characterise the aspectual properties of verbal predicates;
(vii) *links* of the arguments *to the syntactic subcategorization frames*, as represented in the PAROLE lexicons;
(viii) *'qualia' structure*, following the Generative Lexicon [14], represented by a very large and granular set of semantic relations and features;
(ix) information about *regular polysemous alternation* in which a word-sense may enter;
(x) information concerning *cross-part of speech relations* (e.g. *intelligent* - *intelligence*; *writer* - *to write*).
(xi) *semantic relations*, such as hyponymy, synonymy, etc.

The "conceptual core" of the lexicons consists of the basic structured set of "semantic types" (the *SIMPLE ontology*) and the basic set of notions to be encoded for each sense. These notions have been captured in a common "library" of language independent *templates*, which act as "blueprints" for any given type - reflecting well-formedness conditions and providing constraints for lexical items belonging to that type.

There are three main types of formal entities:

- *Semantic Units* **-** word-senses are encoded as *Semantic Units* (*SemU*) and assigned a *semantic type* from the Ontology, plus other sorts of information specified in the associated *template*, which contribute to the characterization of the word-sense.
- *Semantic Type* **-** each type involves structured information represented as *template*. The semantic types themselves are organized into the *Ontology*, which allows for the *orthogonal organisation* of types [14].
- *Template* - a schematic structure which the lexicographer uses to encode information about a given lexical item. The template expresses the semantic type, plus other sorts of information characterising multiple dimensions of a word-sense.

Templates are intended both to provide the semantics of the types (which are thus not simply labels) and to guide, harmonize, and facilitate the lexicographic work, as well as to enhance the consistency among the lexicons. A set of top common templates (about 150) has been defined, while the individual lexicons can add more language-specific templates as needed. Templates provide the information that is type-defining for a given semantic type. Lexicographers can also further specify the semantic information in a SemU. We show here the template associated with the SemU for a sense of *lancet*, instantiating the template `Instrument`:

Usem:	<lancet-1>
BC number:	
Template_Type:	**[Instrument]**
Unification_path:	**[Concrete_entity| Artifact$_{\textbf{Agentive}}$ | Telic]**
Domain:	*Medicine*
Semantic Class:	*Instrument*
Gloss:	a surgical knife with a pointed double-edged blade; used for punctures and small incisions
Event type:	<Nil>
Pred_Rep.:	<Nil>
Selectional Restr.:	<Nil>
Derivation:	<Nil>
Formal:	*isa* (<lancet-1>, <knife>: [**Instrument**])
Agentive:	*created_by* (<lancet-1>, <make>: [**Creation**])
Constitutive:	*made_of* (<lancet-1>, <metal>: [**Substance**]) *has_as_part* (<lancet-1>, <edge>: [**Part**])
Telic:	*used_for*(<lancet-1>, <cut>: [**Constitutive_change**]) *used_by* (<lance-1t>, <doctor>: [**Human**])
Synonymy:	<Nil>
Collocates:	*Collocates* (<SemU1>,…,<SemUn>)
Complex:	<Nil>

5 The Structure of the Multilingual ISLE Lexical Entry (MILE)

5.1 Basic EAGLES Principles

We remind here just a few basic methodological principles from previous EAGLES, which have proven useful in the process of reaching consensual *de facto* standards in a bottom-up approach and are at the basis also of ISLE work.

The MILE is envisaged as a highly *modular* and possibly *layered* structure, with different levels of recommendations. Modularity with \respect to MILE can be thought of at least in the following: i) in the macrostructure (*meta-information*: versioning of the lexicon, languages, updates, status, project, origin, etc. (see e.g. OLIF [18]) and general architecture (to specify the interactions of the various modules, and the general structure in which they are inserted, both in the interlingua- and transfer-based approaches, and in possibly hybrid solutions; the relation between the source language (SL) and target language (TL) portions of a lexicon), and ii) in the microstructure of the MILE, i.e. in the word-sense level (the basic unit in the multilingual layer).

The MILE recommendations should also be very *granular*, in the sense of reaching a maximal decomposition into the minimal *basic notions* that reflect the phenomena we are dealing with. This principle was previously recommended and used to allow easier reusability or mappability into different theoretical or system approaches [9]: small units can be assembled, in different frameworks, according to different (theory/application dependent) generalisation principles.

On the other side, past EAGLES experience has shown it is useful in many cases to accept *underspecification* with respect to recommendations for the representation of some phenomenon (and *hierarchical structure* of the basic notions, attributes, values, etc.).

5.2 The MILE as a Lexical Meta-entry

The MILE is intended as a *meta-entry*, acting as a common representational layer for multilingual lexical resources. The key-ideas underlying the design of a meta-entry can be summarized as follows. Different theoretical frameworks appear to impose different requirements on how lexical information should be represented. One way of tackling the issue of theoretical compatibility stems from the observation that existing representational frameworks mostly differ in the way pieces of linguistic information are mutually implied, rather than in the intrinsic nature of this information. To give a concrete example, almost all theoretical frameworks claim that lexical items have a complex semantic organization, but some of them try to describe it through a multidimensional internal structure, others by specifying a network of semantic relations, and others in terms of argument frames. A way out of this theoretical variation is to augment the expressive power of an annotation scheme both *horizontally*, i.e. by distributing the annotated information over mutually independent "coding layers", and *vertically*, by further specifying the information conveyed by each such layer.

With respect to this issue, the MILE is designed to meet the following desiderata:

- factor out linguistically independent (but possibly correlated) primitive units of lexical information;
- make explicit information which is otherwise only indirectly accessible by NLP systems;
- rely on lexical analysis which have the highest degree of inter-theoretical agreement;
- avoid framework-specific representational solutions.

All these requirements serve the main purpose of making the lexical meta-entry open to task- and system-dependent parameterization. The CLWG has also agreed that the MILE encompasses and is built on the whole monolingual entry, and will include a number of interconnected modules, which in turn further subdivide into more fine-grained structures. The three foreseen major components are:

1. Monolingual linguistic representation – this includes the morphosyntactic, syntactic, and semantic information characterizing the MILE in a certain language. It generally corresponds to the typology of information contained in existing major lexicons, such as PAROLE/SIMPLE, (Euro)WordNet (EWN), COMLEX, and FrameNet, as the result of a deep process of evaluation concerning the usefulness for multilingual tasks and the potentiality of integration into a unitary MILE. Following the general organization of computational lexicons like PAROLE/SIMPLE, at the monolingual level the MILE sorts out the linguistic information into different layers, respectively for phonological, morphological, syntactic and semantic dimensions. Typologies of information to be part of this module include (not an exhaustive list):

- *Phonological layer*
  - phonemic transcription
  - prosodic information
- *Morphological layer*
  - Grammatical category and subcategory
  - Inflectional class
  - Modifications of the lemma
  - Mass/count
- *Syntactic layer*
  - Idiosyncratic behaviour in specific syntactic rules (passivisation, middle, etc.)
  - Auxiliary
  - Attributive vs. predicative function, gradability (only for adjectives)
  - List of syntactic positions forming subcategorization frames
  - Syntactic constraints and property of the possible ‘slot filler’
  - Possible syntactic realizations and grammatical functions of the positions
  - Morphosyntactic and/or lexical features (agreement, prepositions and particles introducing clausal complements)
  - Information on control (subject control, object control, etc.) and raising properties
- *Semantic layer*
  - Characterization of senses through links to an Ontology
  - Domain information
  - Gloss
  - Argument structure, semantic roles, selectional preferences on the arguments
  - Event type, to characterize the actionality behaviour
  - Link to the syntactic realization of the arguments
  - Basic semantic relations between word senses: synonymy (synset), hyponymy, meronymy, etc.
  - More specific semantic/world-knowledge relations among word-senses (such as EWN relations, SIMPLE Qualia Structure, FrameNet frame elements)
  - Regular polysemous alternation
  - Cross-part of speech relations

As can be seen from the list above, some of these types of information provides *explicit* representations of the MILE content through reference to formal resources such as ontologies, feature sets, lists of semantic relations, common predicates or argument structures. A general issue in ISLE concerns whether consensus has to be pursued at the generic level of “type” of information or also at the level of its “values" or actual ways of representation. The answer may be different for different notions, e.g. try to reach the more specific level of agreement also on "values" for types of meronymy, but not for types of ontology.

The monolingual module will be one of the bases to define the transfer conditions, but can also be possibly detached to form a totally independent lexicon to be used in standard monolingual tasks.

2. Collocational information – This module includes more or less typical and/or fixed syntagmatic patterns (collocations, multiwords, etc.) including the lexical head defined by the MILE. It conveys further, more granular uses of the MILE, which simply cannot be expressed through the monolingual representation apparatus provided in 1), thus contributing to perform more subtle and/or domain specific characterisations. It includes at least:

- Typical or idiosyncratic syntactic constructions
- Typical collocates
- Support verb construction
- Phraseological or multiwords constructions
- Compounds (e.g. noun-noun, noun-PP, adjective noun, etc.)
- Corpus-driven examples

In this module – not yet dealt with in the previous EAGLES - we experiment more strongly the limits of the representation means adopted in current lexicons and models. It is however critical in a multilingual context both to characterise a word-sense in a more granular way and to make it possible to perform a number of operations, such as WSD or translation in a specific context. Open issues are: i) what generalisations can be captured and formally characterised, ii) what must be simply listed (but even lists may be partially categorised), iii) what representation can be provided (e.g. a Mel'cuk style characterisation of support verb constructions, FrameNet style description of syntactic-semantic "constructions", etc.). Here, synergies with the NSF-XMELLT project on multi-word expressions are exploited. First proposals for the representation of support verbs and noun-noun compounds in multilingual computational lexicons are laid out, and now tested on some language pairs.

The difference between 1) and 2) above corresponds roughly to the one between i.) *coarse-grained* (general purpose) characterisations in terms of prototypical properties, captured by the formal means in 1), which divides the meaning space in large areas and is sufficient for some NLP tasks; and ii.) f*ine-grained* (domain or text dependent) characterisations, mostly in terms of collocational/syntagmatic properties, which are necessary for specific tasks, such as MT. Different types of information have different operational specialisation, and raise different issues in monolingual vs. multilingual tasks. For instance, some verb-complement pairs, although not representing problematic case in the SL, may call for idiosyncratic transfer in the TL. Similarly, it is well-known that sense distinctions may be different in monolingual and multilingual lexicons.

3. Multilingual apparatus *(e.g. transfer conditions and actions)* – This represents the focal part of the CLWG activities, which concentrates its main effort in proposing a general framework for the expression of multilingual transfers. Some of the main issues at stake here are:

- identify a typology of the most common cases of problematic transfer (actually this task has been partially performed during the survey phase of the project);
- identify which conditions must be expressible and which transformation actions are necessary, in order to establish the correct multilingual mappings;
- select which types of information these conditions must access in the modules 1) and 2) above;
- identify the various methods of establishing SL --> TL equivalence (e.g. translation, near equivalent, gloss, example, example + translation, etc.)
- examine the variability of granularity needed when translating in different languages, and the architectural implications of this.

The line pursued by the CLWG is to define the multilingual layer of the MILE as an additional dimension on top of the monolingual ones. Related units are not modified but rather new 'correspondence' objects are created, pointing to already existing monolingual elements. This grants the maximum degree of flexibility and consistency in reusing existing monolingual resources to build new bilingual/multilingual lexicons. Multilingual correspondences in the MILE may involve different elements, ranging from raw surface strings, to syntactic and semantic units, up to more abstract objects like semantic predicates, concepts, etc. Correspondences can also be filtered or enriched with new (more example-based) information, not present in monolingual lexicons, but essential to establish multilingual links.

There are several dimensions concerning the issue of correspondences, which enter into shaping their actual form:

1. *Contextuality*, i.e. the extent to which context is relevant for the description of a transfer. Two cases usually occur: *simple lexical transfer*, and *complex lexical transfer, or contextual transfer,* ranging from a change in gender to a complete restructuring of a sentence (cf. IT. *nuota volentieri*$_{adv}$ --> EN. *he likes*$_{vrb}$ *to swim*). The multilingual layer must specify the configurational consequences of a correspondence and will contain a whole set of conditions to express complex transformation in the SL to TL transfer, involving argument restructuring, change in the obligatoriness of positions, adjunct specifications, element addition or deletion, etc.
2. *Ambiguity*. There are two basic cases: i) a one-to-one transfer, mainly in special domains; ii) a one-to-many transfer, where a given lexical unit can be translated in several ways: the transfer module needs to have a *test* part to identify the correct reading, referring e.g. to the syntactic configuration of which the lexical unit is a part or to its semantic properties.
3. *Lexical unit internal structure*. There are three basic cases: single words, compounds (the type of German/Dutch/Finnish: agglutinated), multiwords (i.e. several words which together form a semantic/lexical unit). The last two are very frequent in terminology. Sometimes the transfer needs a description of the head of a multiword, or the internal structure is referred to in tests.

As a consequence, the multilingual module of the MILE is structured in at least three parts:

1. *test part*, specifying the context which must hold for a given transfer;
2. *action part*, specifying what needs to be done if this transfer is selected;
3. *typed link*s, specifying the type of the transfer link itself.

Tests and actions will be expressed by making reference to the whole representational apparatus used to characterize the monolingual linguistic information. This way, it will be possible to use all the available data structures in order to formulate the most proper multilingual links. Moreover, the multilingual module may not have the same requirements for different applications: it may be simpler for CLIR, which may resort to a subset (including an ontology or semantic hierarchy) of the information needed for MT.

At the formal level, the MILE architecture will be formalized by using XML, but the possibility of using emerging standards for content description, such as RDF Schema (cf. www.w3.org/RDF), is also carefully evaluated.

6 Current Results and Enlargement to Asian Languages

Results of on-going work are: i) a list of types of information that should be encoded in each module; ii) a list of transfer condition types; iii) linguistic specifications and criteria; iv) a format for their representation in multilingual lexicons; v) their respective weight/importance in a multilingual lexicon (towards a layered approach to recommendations). The MILE is also accompanied by a simple lexicographic tool[7], to allow lexical entries to be encoded according to the MILE structure.

An enlargement of the group to involve also Asian languages is going on, as an important further step, also through new common initiatives. Representatives of Chinese, Japanese, Korean, and Thai languages have contributed to ISLE work. Also the newly formed *Asian Federation of Natural Language Processing Associations* (AFNLPA), chaired by J. Tsujii, declared interest in the ISLE standardisation initiative.

7 Conclusions

In this paper we focused on the MILE, the multilingual lexical meta-entry proposed by the ISLE CLWG as the standard representational format for multilingual computational lexical resources, with particular attention to the needs and requirements of MT systems. The MILE main features are i) its distributed coding architecture and ii) a strong emphasis on representation modularity. Lexical representation is articulated over different information layers, each factoring out different, but possibly interrelated, linguistic facets of information, relevant in order to establish multilingual lexical links.

[7] A first prototype has been built by N. Bel and M. Villegas.

Acknowledgements

We thank all the members of the ISLE CLWG for their active participation and contribution to this enterprise and for their deep commitment to its success.

References

1. Bel N., Busa, F., Calzolari, N., Gola, E., Lenci, A., Monachini, M., Ogonowski, A., Peters, I., Peters, W., Ruimy, N., Villegas, M., Zampolli, A. SIMPLE: A General Framework for the Development of Multilingual Lexicons. In: LREC Proceedings. Athens (2000)
2. Burnard, L., Baker, P., McEnery, A., Wilson, A.: An analytic framework for the validation of language corpora. Report of the ELRA Corpus Validation Group (1997)
3. Calzolari, N.: An Overview of Written Language Resources in Europe: a few Reflections, Facts, and a Vision. In: Rubio, A., Gallardo, N., Castro, R., Tejada A. (eds.): Proceedings of the First International Conference on Language Resources and Evaluation. Granada (1998) 217-224
4. Calzolari, N., Grishman, R., Palmer, M. (eds.): Survey of major approaches towards Bilingual/Multilingual Lexicons. ISLE Deliverable D2.1-D3.1. Pisa (2001)
5. Calzolari, N., Mc Naught, J., Zampolli, A.: EAGLES Final Report: EAGLES Editors' Introduction. Pisa (1996)
6. EAGLES: Evaluation of Natural Language Processing Systems. Final Report. CST, Copenhagen (1996). Also at http://issco-www.unige.ch/projects/ewg96/ewg96.html.
7. GENELEX Consortium: Report on the Semantic Layer. Project EUREKA GENELEX, Version 2.1. Paris (1994)
8. Gibbon, D., Moore R., Winski, R.: Handbook of Standards and Resources for Spoken Language Systems. Mouton de Gruyter, Berlin, New York (1997)
9. Heid, U., McNaught, J.: EUROTRA-7 Study: Feasibility and Project Definition Study on the Reusability of Lexical and Terminological Resources in Computerised Applications. Final report. Stuttgart (1991)
10. Leech, G., Wilson, A.: Recommendations for the morphosyntactic annotation of corpora. Lancaster (1996)
11. Lenci, A., Busa, F., Ruimy, N., Gola, E., Monachini, M., Calzolari, N., Zampolli, A.: Linguistic Specifications. SIMPLE Deliverable D2.1. ILC and University of Pisa (1999)
12. Monachini, M., Calzolari, N.: Synopsis and comparison of morphosyntactic phenomena encoded in lexicons and corpora. A common proposal and applications to European languages. ILC-CNR, Pisa (1996)
13. Monachini, M., Calzolari, N.: Standardization in the Lexicon. In: H. van Halteren (ed.): Syntactic Wordclass Tagging. Kluwer, Dordrecht (1999) 149-173
14. Pustejovsky, J.: The Generative Lexicon. Cambridge, MA, MIT Press (1995)
15. Ruimy, N., Corazzari, O., Gola, E., Spanu, A., Calzolari, N., Zampolli, A.: The European LE-PAROLE Project: The Italian Syntactic Lexicon. In: Proceedings of the First International Conference on Language resources and Evaluation. Granada (1998) 241-248.
16. Sanfilippo, A. et al. : EAGLES Subcategorization Standards. (1996). See www.icl.pi.cnr.it/EAGLES96/syntax/syntax.html
17. Sanfilippo, A. et al. : EAGLES Recommendations on Semantic Encoding. (1999). See www.ilc.pi.cnr.it/EAGLES96/rep2

18. Thurmair, G.: OLIF Input Document. (2000). See http://www.olif.net/main.htm
19. Underwood, N., Navarretta, C.: A Draft Manual for the Validation of Lexica. Final ELRA Report. Copenhagen (1997)
20. Zampolli, A.: The PAROLE project in the general context of the European actions for Language Resources. In: Marcinkeviciene, R., Volz, N. (eds.): TELRI Proceedings of the Second European Seminar: Language Applications for a Multilingual Europe. IDS/VDU, Manheim/Kaunas (1997)
21. Zampolli, A.: Introduction of the General Chairman. In: Rubio, A., Gallardo, N. Castro, R., Tejada A. (eds.): Proceedings of the First International Conference on Language Resources and Evaluation. Granada (1998)

Quantitative Comparison of Homonymy in Spanish EuroWordNet and Traditional Dictionaries*

Igor A. Bolshakov, Sofia N. Galicia-Haro, and Alexander Gelbukh

Center for Computing Research (CIC),
National Polytechnic Institute (IPN), Mexico City, Mexico
{igor,sofia,gelbukh}@cic.ipn.mx

Abstract. A quantitative comparative study of homonymy in four well-known electronic Spanish dictionaries—EuroWordNet and three traditional dictionaries—is presented. It is shown that though structuring of word senses is quite different in all dictionaries under comparison, EuroWordNet differs from the traditional dictionaries much more than these differ from each other. It is also shown that the ordering of the word senses in Spanish EuroWordNet less agrees with the use of the senses in texts than the ordering in traditional dictionaries.

1 Introduction

Different dictionaries usually give different sense sets for the same words. In this work we present quantitative evaluation and comparison of word sense structuring in the following four well-known Spanish electronic dictionaries: of Anaya group [1], by María Moliner [2], of Spanish Royal Academy [3], and EuroWordNet [4]. Our motivation was to proof or disproof the following assumptions:

- The dictionaries tend to have similar sense sets, since[1] (1) all good lexicographers share the same word sense structures in their minds, and (2) if a lexicographer does not elaborate the sense structure for a given word, he or she borrows some parts of it from other dictionaries.
- EuroWordNet dictionaries (in particular, Spanish) have made some disruption in the lexicographic tradition since they were compiled on a different ideological basis—by computer-oriented linguists and without deep lexicographic considerations.

Our motivation was also to check whether simple statistical methods could be useful for selecting a 'better' dictionary for future applications.

2 Comparison of the Dictionaries

Experimental setting. Ideally, the comparison methods discussed below operate on the representation of the dictionaries as very large sets of ordered lists (word senses

* Work done under partial support of CONACyT, SNI, and CGEPI-IPN, Mexico.

[1] I. Mel'čuk, private communication.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 280–284, 2002.

for each word) and the mappings between these lists (the correspondences between the word senses in different dictionaries); what is more, one of our experiments would, ideally, rely on the textual frequencies of specific word senses. However, given the large amount of senses in all four dictionaries, constructing such mappings and counting the frequency of each sense would be too expensive.

To simplify our calculations, we worked with small randomly chosen samples of the dictionaries. Though we realize that our results are then quite approximate, we believe they do show the general tendencies.

First, we constructed a small corpus marked with senses. We started from the well-known LEXESP corpus,[2] which contains a balanced representation of modern Spanish and has the size of 5 million words. Of those, we have randomly (by the position in the file) chosen 158 words and, basing on the context, assigned them the senses from all four dictionaries.

Then, to further simplify our calculations, we eliminated some words from this corpus: (1) In two cases, we eliminated words with the same sense, so that all words in our toy corpus had different senses. Since there were only few repeated senses, this should not affect the results but simplifies our calculations. (2) We also eliminated the words that could not be assigned a sense in at least one dictionary; there were 27% of such words, the majority of them being adjectives absent in EuroWordNet.

After these operations, we obtained a corpus of K = 114 supposedly most frequently used word senses, marked each one with a word sense number according to each of the four dictionaries. A fragment of the complete list of words is presented in Appendix 1.

This, in turn, is equivalent to the selection of a small sample of each of the four dictionaries, reflecting mainly the most frequently used senses. All our calculations described below are based on these samples instead of complete dictionaries.

Comparison of the Number of Word Senses. For each word (letter string) w of our corpus, we found the number x_{wd} of its senses in each dictionary d. The values x_{wd} distributed as follows:

	Anaya	Moliner	Academy	WordNet
Average $\bar{x}_d$	5	4.5	7.3	3.6
Median	4	3	5	2.5

where

$$\bar{x}_d = \frac{1}{K}\sum_{w=1}^{K} x_{wd}\,.$$

It can be seen that EuroWordNet has considerably less senses per headword.

The similarity between the numbers of the senses in the entries of the dictionaries d_1 y d_2 calculated by Pearson's formula [5]

$$P_{xy} = \frac{\frac{1}{K}\sum_{i=1}^{K} x_i y_i - \bar{x}\,\bar{y}}{\sqrt{\left(\frac{1}{K}\sum_{i=1}^{K} x_i^2 - \bar{x}^2\right)\left(\frac{1}{K}\sum_{i=1}^{K} y_i^2 - \bar{y}^2\right)}}$$

where $\bar{x} = \bar{x}_{d_1}$, $\bar{y} = \bar{x}_{d_2}$, $x_i = x_{id_1}$, $y_i = x_{id_2}$, is as follows:

[2] Kindly provided to us by H. Rodríguez of Universitat Politècnica de Catalunya.

	Anaya	Moliner	Academy	WordNet
Anaya	1.000	0.812	0.947	0.565
Moliner		1.000	0.826	0.616
Academy			1.000	0.556
WordNet				1.000

It can be observed that the correlation between EuroWordNet and the other dictionaries is smaller than among these three.

Comparison of the Ordering of Senses. Using our toy corpus, we compared the positions of the senses within their groups for a given word (letter string) in the four dictionaries.

Let i = 1, ..., K be the number of word in our corpus, x the number of dictionary, and k_{ix} = 1, ..., n_{ix} the corresponding sense number out of n_{ix} senses in total for the corresponding letter string in the corresponding dictionary. Then the relative position

$$r_{ix} = \begin{cases} \dfrac{k_{ix} - 1}{n_{ix} - 1}, & \text{if } n_{ix} \neq 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

reflects how far the given sense is from the top of the list of senses for the given word; note that if k_{ix} = 1 then the relative position is 0 independently of the total number of senses n_{ix}. Note also that in the case n_{ix} = 1 it always holds k_{ix} = 1, thus the second option in (1). So we calculate the mean ordering distance between the dictionaries x and y as:

$$D_{xy} = \frac{1}{K} \sum_{i=1}^{K} \left| r_{ix} - r_{iy} \right|$$

The obtained values of D_{xy} are as follows:

	Anaya	Moliner	Academy	WordNet
Anaya	0.000	0.207	0.167	0.386
Moliner		0.000	0.254	0.388
Academy			0.000	0.411
WordNet				0.000

Once more, EuroWordNet dictionary differs from the other three considerably more that these three from each other.

Suitability of the Ordering of Senses. We expect that the lexicographer should list first the (intuitively) most frequent senses. Thus, using our toy corpus of (supposedly) most frequent senses, we considered the distribution of the relative positions of these senses calculated by the formula (1), which proved to be the following:

	Anaya	Moliner	Academy	WordNet
Average ($\bar{x}_d$)	0.271	0.164	0.314	0.419
Median	0.000	0.000	0.184	0.310

As one can see, the ordering of senses agrees very well with the frequencies of usage for Anaya and Moliner dictionaries. For Academy dictionary, the agreement is slightly less probably because it contains many obsolete senses. Finally, the ordering of senses in the EuroWordNet dictionary seems to be close to random.

3 Conclusions

All four dictionaries under comparison are different both in the mean number of senses per word (letter string) and in their ordering of senses for a given word. Hence, our first assumption can be rather rejected. We can admit, however, that a deeper lexicographic research can show whether these differences are mainly due to very infrequent senses, such as dialectic. Spanish is spoken by almost 400 millions of people in many countries that have great dialectic differences.

The three traditional dictionaries have greater differences with EuroWordNet than between each other. Specifically, Spanish EuroWordNet has significantly less number of senses, lacking quite frequently used senses (especially adjectives). While the ordering of senses in the traditional dictionaries agrees quite well with the relative frequencies of their usage, the ordering of EuroWordNet seems to be almost random. Thus, our second assumption has been confirmed.

References

1. Anaya Group. Diccionario Anaya de la lengua. Internet. Marzo 1997.
2. María Moliner. Diccionario de uso del español. GREDOS Primera edición en CD-ROM 1996.
3. Real Academia Española. Diccionario de la Lengua Española. Edición vigésima primera, en CD-ROM de ESPASA CALPE. 1995.
4. EuroWordNet Consortium, Spanish version. 1997-1998.
5. McEnery, T. & A. Wilson. Corpus Linguistics. Edinburgh University Press. 1996.

Appendix 1. Examples of Homonyms in the Text We Investigated

In the table below, the number k_{ix} of the sense in our corpus and the total number n_{ix} of senses for the given word (as letter string) are given. Here (N) stands for noun, (A) for adjective; unmarked words are verbs.

Word (Spanish)	English	Anaya	Moliner	Academia	EWnet
aceleración N	acceleration	1/2	1/3	1/2	2/2
Alcanzar	to reach	4/8	4/8	7/18	2/4
año N	year	2/3	1/3	3/7	2/3
apresurar	to hasten	1/2	1/2	1/2	2/4
asunto N	affair	1/4	1/2	6/6	6/6
atención N	attention	1/2	1/5	1/4	5/7
comida N	dinner	3/4	3/3	2/4	7/8
creer	believe	1/6	1/3	1/5	2/6
dar$_1$	overlook	24/29	10/12	38/47	9/9
dar$_2$	to cause	7/29	4/12	21/47	6/8
decir	to say	1/8	1/10	1/10	3/8
diario N	newspaper	2/3	2/5	4/5	2/6
dormir	sleep	1/6	1/8	1/12	1/2

Word (Spanish)	English	Anaya	Moliner	Academia	EWnet
encontrar	to meet	3/8	2/4	5/8	9/9
enseñar	point out	2/6	2/3	3/6	4/8
girar	to turn	1/5	1/7	1/7	7/15
hombre$_1$ N	male	2/5	1/4	2/10	4/6
hombre$_2$ N	adult	3/5	1/4	3/10	4/6
llamar	to name	7/12	4/10	4/13	8/12
mover	to move	1/9	1/9	1/10	2/2
nido N	nest	1/4	1/7	1/8	2/2
padre N	parents	5/8	7/9	10/12	1/2
pasar	go through	3/35	9/41	24/59	19/21
posible A	possible	1/2	1/2	1/2	1/2
proyecto N	plan	2/3	1/2	4/4	1/3
régimen N	government	2/6	2/3	2/7	3/3
rendimiento N	income	1/3	1/2	4/5	4/4
situación N	situation	2/3	2/2	4/6	4/7
tener	possess	2/13	2/13	2/24	4/4
terreno N	field	4/6	4/5	3/6	4/4
varón N	male	2/3	2/3	2/4	1/2

Compilation of a Spanish Representative Corpus*

Alexander Gelbukh, Grigori Sidorov, and Liliana Chanona-Hernández

Center for Computing Research, National Polytechnic Institute
{gelbukh,sidorov}@cic.ipn.mx, lchanona@mail.com

Abstract. Due to the Zipf law, even a very large corpus contains very few occurrences (tokens) for the majority of its different words (types). Only a corpus containing enough occurrences of even rare words can provide necessary statistical information for the study of contextual usage of words. We call such corpus representative and suggest to use Internet for its compilation. The corresponding algorithm and its application to Spanish are described. Different concepts of a representative corpus are discussed.

1 Introduction

Our motivation for this work was the statistical research on collocations and subcategorization in Spanish. For this, we needed to calculate, for each word, the relative frequencies of different types of the contexts in which the word is used. This research required a statistically significant number of contexts of each word. However, due to the Zipf law, any corpus contains very few occurrences (tokens) for the majority of the different words (types) used in it. Thus, even a very large corpus provides very poor combinatorial information for the vast majority of words. On the other hand, the vast majority of the size of the traditional corpus is wasted on the repetition of the same few words.

For this type of statistical research, we suggest the use of a new type of corpus, which we call a representative corpus. Such a corpus contains a fixed (and sufficient—say, 50) number of contexts for each word under consideration—ideally, for as many words of the language as possible (we call these words the vocabulary of the corpus). Such a corpus can be obtained as a subset of a larger corpus, selecting a fixed number of occurrences (in context) of the words included in the vocabulary. However, such a full corpus from which this subset is selected should be really huge.

It has been suggested to use Internet as a huge corpus—for example, “virtual corpus” [2]. Though we use Internet to create a real (rather than virtual) corpus, our considerations do not significantly depend on this.

An important decision in compiling such a corpus concerns the criteria of selection of specific occurrences. One of our requirements was proportional presence of all inflective forms of the words. Other possible requirements could include proportional representation of syntactic structures or different types of texts (styles or genres).

In this paper, we will describe the procedure we used for the compilation of a Spanish representative corpus, and then discuss the obtained results.

* Work done under partial support of CONACyT, CGEPI/COFAA-IPN, and SNI, Mexico.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 285–288, 2002.

2 Algorithm

We consider a corpus lexically and morphologically representative if it contains the contexts for its inflectional forms of the lexemes of included in its vocabulary, in a specific proportion. There are several possible ways to calculate the proportions, which gives different strategies for the selection of the occurrences:

- A specific number of contexts (possibly depending on the specific paradigm position) per wordform of any lemma.
- A specific number of contexts per lemma (possibly depending on part of speech).

In the latter case, this number of contexts per lemma can, similarly, be distributed among its wordforms:

- in an equal proportion,
- in a proportion depending on the specific paradigm position, or
- in the proportion of their frequencies in texts for the specific lemma.

We have chosen the last approach.

Using *AltaVista* search engine, we looked for the contexts of specific lemmas and wordforms in Internet, selected the contexts according to our criteria, and collected them, thus compiling our corpus. The process is illustrated in Fig. 1.

The algorithm is controlled by the agenda, which is a list of words for which the corpus should contain contexts but does not yet contain them. Iteratively, a word is taken from the agenda, the pages containing this word are retrieved from Internet, and "good" contexts are selected for inclusion into the corpus. Also, the retrieved pages are used to find new words; these are added to the agenda, so that at a later iteration their contexts will be looked for. The process stops when the agenda is empty or after a time limit is exceeded. Below we describe this process in more detail.

First, an initial vocabulary (word list) is compiled from various resources (dictionaries and traditional corpora), see left hand part of Fig. 1. The block of lexical analysis selects only some part of the data (say, for dictionaries, it can select only headwords). This forms the initial seed of data to be put onto the agenda.

This seed is filtered to prevent passing of the non-lexical elements (numbers, web addresses, etc.) to agenda. Then the words are morphologically normalized to form lemmas, which are put onto the agenda. Note that later (upper part of Fig. 1) all inflectional forms of each lemma are generated; thus if the seed contained a form *did*, the contexts will be looked for for the forms *do*, *does*, *done*, *doing* as well.

Then the lemmas are taken (by the control module) from the agenda one by one, all inflectional forms are generated for each lemma, and for each form, the desired number of contexts is determined by the weight calculation module (upper part of Fig. 1).

Recall that we have chosen the proportion of wordforms according to their frequencies in texts. We estimate these frequencies through the number of documents in Internet containing given wordform. Fortunately, AltaVista search engine provides this number. With this search engine it can be also guaranteed that only the pages in specific language (Spanish in our case) are considered. However, we do not resolve lexical ambiguity within the given language: what is looked for is a letter string rather a wordform of a specific lemma; overcoming this difficulty is a topic of our future work.

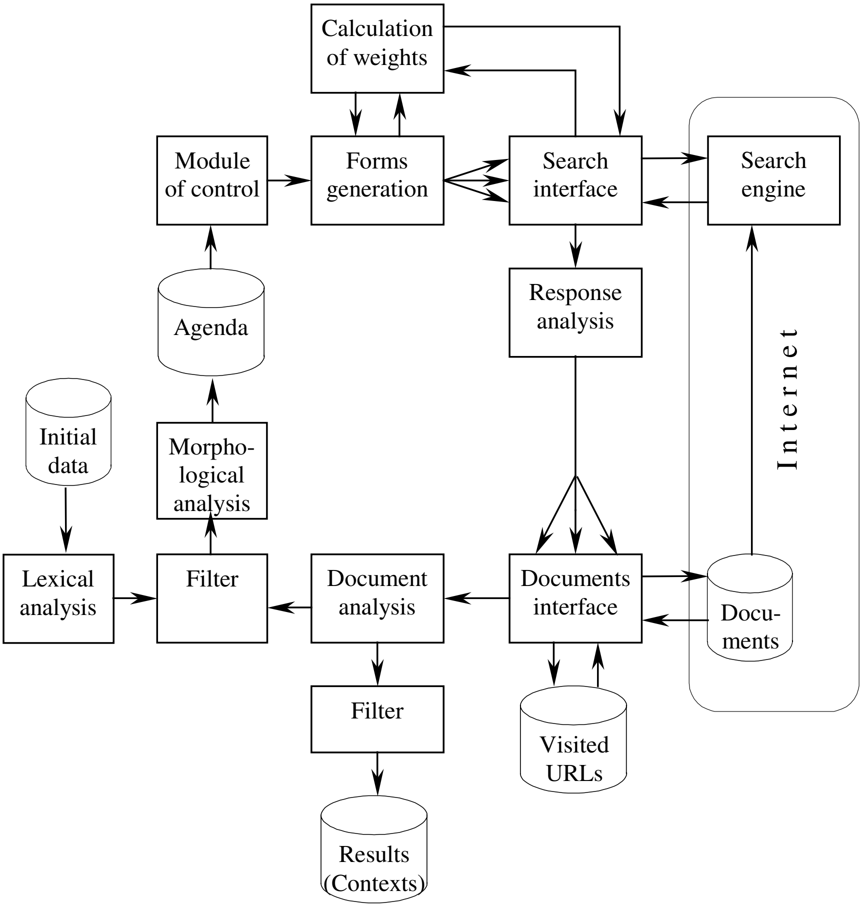


Fig. 1. Automatic compilation of a representative corpus.

After the desired number of contexts for a given wordform is determined, pages containing this wordform are retrieved from Internet (right hand part of Fig. 1; duplicate loading of the same URL is avoided), the contexts are found, filtered using some criteria of suitability (see below) and added to the corpus (bottom part of Fig. 1). For a given wordform, the process is repeated until the desired number of suitable contexts is added to the corpus.

A number of heuristics is used to prevent "bad" contexts from inclusion to the corpus. Say, a "good" context should have enough words around the wordform in question, it should consist of plain text rather than control elements or graphics, it should not resemble too closely a context already existing in the corpus, etc.

Each time a page is retrieved, it is searched for strings absent in the vocabulary. These potentially new words are filtered, analyzed, and added to the agenda in the same manner as the initial seed (as described above), thus enriching the corpus' vocabulary (left hand part of Fig. 1).

3 Experimental Results

Using the headwords from an existing Spanish explanatory dictionary (Anaya), we obtained a seed of about 30,000 lemmas. Since we used only headwords, no morphological normalization was necessary.

For the selection of contexts, we used the value of at least $N = 50$ contexts per lemma; however, if the paradigm of a lemma had more than N wordforms, we included one context per wordform. However, in the morphological paradigms we ignored the Spanish verbal forms with clitics.

We used the following criteria for a context to be included in the corpus:

- It should contain at least 8 words, and
- 3-word contexts should not repeat. A 3-word context is composed by the wordform itself, one significant word to the left and one to the right (by significant word we mean any word but auxiliary words such as articles, prepositions, etc.).

Until now, we have compiled a corpus with the vocabulary of approximately 45,000 lemmas. It consists of over 100 million words.

4 Conclusions

We have suggested a special type of a corpus—a representative corpus—that does not present the problems traditional corpora present dues to the Zipf law. Such corpus is useful for the statistical research on word combinability, where a statistically significant number of contexts of word usage is required. We have also discussed a method for compilation of such corpus using Internet as a source. The method has been applied to compile a large Spanish representative corpus.

The use of this corpus for learning Spanish subcategorization and collocation dictionaries is the topic of our future work.

References

1. Biber, D., S. Conrad, and D. Reppen (1998). *Corpus linguistics. Investigating language structure and use.* Cambridge University Press, Cambridge.
2. Kilgariff, A. (2001). Web as corpus. In: *Proc. of Corpus Linguistics 2001 conference*, University center for computer corpus research on language, technical papers vol. 13, Lancaster University, 2001, pp 342-344.

Aligning Multiword Terms Using a Hybrid Approach

Arantza Casillas[1] and Raquel Martínez[2]

[1] Facultad de Ciencias,
Universidad del País Vasco
arantza@we.lc.ehu.es
[2] Escuela Superior de CC. Experimentales y Tecnología,
Universidad Rey Juan Carlos
r.martinez@escet.urjc.es

Abstract. In the context of parallel corpus alignment research between a pair of languages with various and important distinguishing factors (e.g., structural, lexical, morpho-syntactical), this paper presents an approach that deals with multiword terms alignment. Our system, ALINTEC, implements a hybrid strategy that adds various kinds of linguistic knowledge (an aligned corpus at the sentence level, POS tagging, grammatical patterns, and a bilingual glossary) to quantitative criteria such as frequency and distribution of terms in the corpus. The experiments were undertaken on a parallel corpus consisting on a collection of administrative and legal documents in Spanish and Basque. This pair of languages is representative of the context in which our work is framed. The results show that our approach obtains reasonably good results in aligning terms of a pair of languages of different typology such as Spanish and Basque.

1 Introduction

The difficulty of translating collocations and multiword terms is frequently cited in the specialized bibliography about machine translation. Adding to the problem that many multiword terms are not compositional, knowledge domain is required to find or to generate the correct translation of a source multiword term. On the other hand, the manual compilation and validation of terms and their translations has an enormous cost. Consequently, the identification of terms and their translations with a minimum of human intervention is a critical issue in Natural Language Processing (NLP) research. A tool that can automate these tasks could be used to great advantage in concrete areas such as machine translation, cross-lingual information retrieval, word sense disambiguation, computer-assisted language learning and generation of bilingual documentation.

Of those works which use a statistical approach to align terms, one of the most remarkable is [13]. The authors describe a program, *Champollion*, which aligns collocations and individual words in a bilingual corpus with aligned sentences. They obtain reasonably good results with a subset of Hansards corpus

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 289–299, 2002.

(English-French) and frequent collocations. Additionally, [12] proposes a method for the recognition of multiword non-compositional compounds in bitexts that is based on the predictive value of a translation model. Some proposals add bilingual dictionaries to support alignment ([17] and [11]). Both take into account only noun phrases with results limited by the generative power of the bilingual dictionary.

Much of the work that has been developed in order to establish multiword term correspondences from a bilingual parallel corpus is based on the assumption that a term is always translated into the same lexical unit [2]. Works such as [7] and [6] found the alignment of frequent word pairs in the similarity of the distributions in their respective texts. Others approaches, from [9] to more recent works as [16] and [18], focus on finding structural correspondences of phrase level.

In this work, we tackle the case of a pair of typologically distant languages: Spanish and Basque. We have used a parallel corpus in Spanish and Basque named BOB. It consists of a collection of administrative and legal documents of approximately 500,000 words in each language. Documents in the corpus were composed by Administration clerks and translated by translators. We have noted that in this corpus multiword terms (MWTs) do not always have a consistent translation, so simply using known criteria, such as frequency and distribution, could be insufficient to establish terms alignment. In such cases, we propose a hybrid approach that adds various types of linguistic knowledge to these criteria. The results show that our approach produces reasonably good results in aligning terms taken from a pair of languages in which a number of additional difficult factors exit, such as lexical, structural, and morpho-syntactical differences, and with a domain that is not completely standardize in Spanish into Basque translation.

The organization of the paper is as follows: the corpus and the characteristics of the languages pair are described in section 2; section 3 introduces our approach to term alignment; in section 4 we present the experimental results and discuss them; finally, section 5 summarizes the conclusions drawn from the work carried out.

2 Characteristics of the Pair of Languages

Spanish and Basque, have coexisted since Spanish became a language on its own, evolving apart from its close Romance relatives (French, Portuguese, Italian, or Catalan). All of these Romance languages are SVO (Subject Verb Object) languages with a rather strict head initial behavior, which is most clear within Noun Phrases (NP). In contrast, Basque, which is a pre-Romanic and pre-Indoeuropean language completely surrounded by Romanic and Indoeuropean languages, displays almost completely opposite properties. It is a SOV (Subject Object Verb) language with very strict head final behavior, not only with NPs but in embedded clauses as well.

An additional difference between Romance languages and Basque is related to the nominal morphology. In this case, Spanish and Basque are both inflected languages. Moreover, Basque is, to an important extend, an agglutinative language, which makes it more difficult to identify the nominal morphology. Owing to this characteristic, the Basque lemmatizer returns a large number of nouns as base categories, instead of adjectives or other type of noun complements.

The translation of Spanish terms into Basque is not completely standardized. Hence, terms translation shows sensible differences depending on translators. These discrepancies in translating terms become apparent in the frequency as well as the distribution of pairs of corresponding terms in the corpus.

3 Our Approach in Aligning Multiword Terms

The assumption that a source term has a unique target term, can be plausible considering the languages and domains with which some approaches have obtained good experimental results. Nevertheless, when terms translation is not standardized in a certain domain, terms might not have a consistent translation, thus terms frequency and distribution could present sensible differences. Thus, pure statistical approaches could not be sufficient [10]. Our approach focuses on reinforcing frequency and distribution criteria with linguistic resources that provide new evidence of the translation plausibility.

The additional resources and criteria that we propose are:

- **The use of a bilingual aligned corpus**: the corpus must be aligned at the sentence level. The result of this alignment permits the establishment of the terms's distribution in the corpus.
- **POS tagging and lemmatization**: permits the determination of a category for each token of the multiword term and the replacement of each of them with their corresponding lemmas. In agglutinative languages such as Basque this replacement is required; the Spanish side was processed in the same way. Therefore, term correspondence is established between the lemmas instead of the inflected forms.
- **Grammatical patterns compatibility**: a correspondence table of grammatical patterns of the phrases in the pair of languages can be used to filter out inappropriate pairs of source and target aligned candidate multiword terms (CMWT).
- **Bilingual glossary search**: if we consider the noncompositional nature of the terms, a bilingual dictionary without technical terms could not help in the alignment process. Nevertheless, some terms are compositional and between both, a semi-compositional nature is possible as well [15]. Therefore a bilingual dictionary can support terms alignment.

Our approach begins with a pair of lists of multiword terms to be aligned, one for each language. The notion of ‘technical terminology’ has no satisfactory formal definition [8]. Nevertheless, works as [4], [1], [8] and [5] use an operational definition of ‘multiword term’: noun phrases that frequently appear in domains or subsets of domains. For the experiments, we obtained the pair of lists of multiword terms automatically from a subset of the BOB corpus using the operational

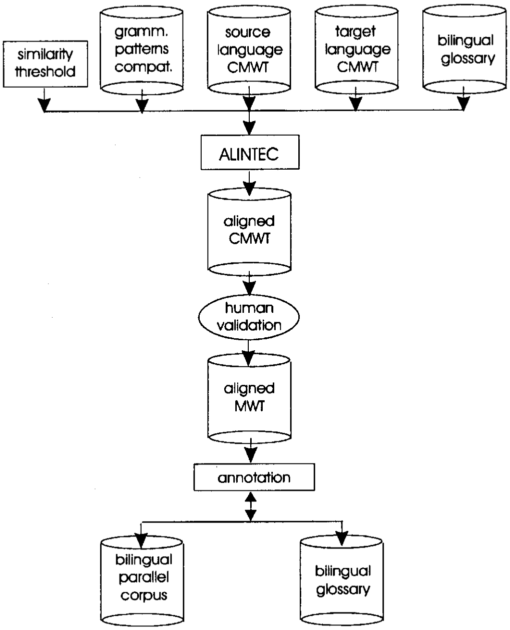


Fig. 1. *ALINTEC* prototype architecture

definition. Thus, criteria such as frequency and grammatical patterns were used in MWT selection in both sides of the corpus. We added some verb phrases to the grammatical patterns.

We have developed a prototype, *ALINTEC*, that combines those resources and criteria to obtain a list with pairs of source and target candidate multiword terms (CMWTsource, CMWTtarget). If there is sufficient alignment evidence, the prototype has a determinist behavior and obtains a sole target candidate MWT for a given source candidate MWT. However, if sufficient alignment evidence does not exit, a list of target candidate MWT's is related to a source candidate MWT. A diagram with the objects and phases involved in *ALINTEC* is shown in Figure 1. The user can choose the next system inputs: (1) The threshold of the similarity coefficient. This value is used to determine the similarity of the distributions of a pair (CMWTsource, CMWTtarget) in the corpus. (2) The correspondence table of grammatical patterns. This file is edited by the user who can decide the patterns with which to test the alignment. (3) Whether an input bilingual glossary is used or not. If yes, the glossary will determine the alignment direction. The language that is the source in the glossary will be the source language in the alignment process.

Our approach obtains aligned terms as a result of the following phases: (1) Compute the distribution vector for each candidate MWT. (2) First selection of pairs (CMWTsource, CMWTtarget) according to the similarity of their distribution vectors. (3) Combine the remainder of available resources to allow the

Freq.	Candidate term	Distribution
50	condición de la licencia	(37,141,149,157,165,174,183,199,208,238, 366,375,469,478,494,503,512,521,531,600, 608,616,648,684,693,702,717,739,748,757, 766,775,784,793,833,842,864,873,882,984, 1100,1163,1248,1275,1284,1293,1679,1688, 1697,1706)
39	lizentzia baldintza	(37,141,149,157,165,174,183,199,208,238, 366,375,469,478,494,503,512,521,531,600, 608,616,717,739,748,757,766,775,784,793, 833,842,984,1100,1163,1679,1688,1697,1706)

Fig. 2. A pair of source and target CMWT with their correspondent distribution vectors

second selection of pairs (CMWTsource, CMWTtarget). (4)A human judge selects the final pairs. Next, we focus on how these phases are executed.

3.1 Compute of the Distribution Vectors

We start from a list of candidate multiword terms in each language. Firstly, the distribution vector corresponding to each term is computed. To determine the distribution we make use of the parallel aligned corpus where the alignment result is expressed by means of SGML tags. This parallel corpus is segmented so that it will have as many segments as aligned blocks of sentences. A vector corresponding to a CMWT will contain as many elements as segments in which the CMWT appears. The value of each element of a vector will be the identification of a segment. As a result of this phase, a distribution vector for each CMWT will be obtained. Figure 2 presents a pair of source and target CMWT's with their respective frequency and distribution vectors.

3.2 First Selection of Corresponding CMWT Pairs

In this phase, the first selection of pairs (CMWTsource, CMWTtarget) is carried out according to the similarity of their distribution vectors. To determine the similarity, the well known *Dice* coefficient (DC) is used [3]. The distribution vector of each source CMWT is compared to the distribution vectors of all target CMWT's. Thus, the prototype selects the pairs (CMWTsource, CMWTtarget) with a *Dice* coefficient greater or equal to a threshold stipulated by the user. Depending on the type of the corpus documents, or even on the typology of the languages, the user can modify the threshold to make the similarity criteria more or less restrictive ($0 \leq DC \leq 1$).

3.3 Second Selection of Corresponding CMWT Pairs

The second selection of pairs is obtained as a result of processing the list of pairs (CMWTsource, CMWTtarget) with the remainder of the available resources (grammatical patterns and bilingual glossary). This process is carried out according to the algorithm of Figure 3.

```
For each CMWTsource do
  If (there is a sole pair (CMWTsource, CMWTtarget) given a CMWTsource) then
    If (syntactic patterns of CMWTsource and CMWTtarget are equivalent) then
      the pair is selected
    else
      If (the complete pair appears in bilingual glossary) then
        the pair is selected
      else
        If (at least one word of the pair is found in bilingual glossary) then
              {it can be a semicompositional term}
          the pair is selected
        else
          the pair is rejected
        endif
      endif
    endif
  else {there is more than one pair given the same CMWT source}
    For each (pair (CMWTsource, CMWTtarget) with the
          same CMWTsource) do
      If (the complete pair appears in bilingual glossary) then
        the pair is selected
      else
        If (at least one word of the pair is found
          in bilingual glossary AND the syntactic
          patterns of the pair are equivalent) then
            the pair is selected
        else
          If (the syntactic patterns of the pair are equivalent)
            then
                the pair is selected
          else
            the pair is rejected
          endif
        endif
      endif
    endfor
  endif
endfor
```

Fig. 3. Algorithm for second selection of corresponding CMWT pairs

Following these three phases, a list of pairs (CMWTsource, CMWTtarget) will be suggested by *ALINTEC*. Finally, a human judge will be able to evaluate the correctness of the aligned terms. The pairs obtained from this evaluation could enhance a bilingual glossary. Thus, the prototype would feedback with the bilingual knowledge generated by itself and, consequently, the prototype itself could improve its efficiency in establishing subsequent alignments.

Table 1. Equivalence syntactic patterns

Spanish	Basque
N. AQ.+	≡ N. AQ.+
N. AQ.	≡ N. N.
N. AQ.	≡ V. N.
N. AQ.	≡ N. V.
N. SP.* AQ.*	≡ N. AQ.+
N. SP. TD.* N.	≡ N.+
N. SP. TD.* N.	≡ N. N. AQ*
V. SP. N.	≡ V. N.
V. V. V.	≡ V.+
V. TD. N.	≡ V. N.
V. TD. N.	≡ N. V.

4 Experiments

4.1 Parameters Used in Aligning CMWT

ALINTEC processed a list of 81 multiword terms in Spanish and a list of 100 multiword terms in Basque. The lists were automatically obtained from a subset of BOB corpus and the term alignment were carried out with a different subset of BOB corpus (of about 30,000 words in each language) with the following inputs:

- A similarity threshold of 0.6. This means that the first selection of pairs (CMWTsource,CMWTtarget) chose those whose distribution vectors were equal or greater than 0.6.
- A series of 11 equivalence syntactic patterns between Spanish and Basque. The algorithm that deals with the second selection of pairs (CMWTsource, CMWTtarget) used the equivalent syntactical patterns shown in Table 1. In that table, N. represents a noun, AQ. represents a qualifying adjective, V. represents a verb, SP. represents a preposition, and TD. represents a definite article. The character '+' represents the quantifier from 1 to n occurrences, and '*', the quantifier from 0 to n occurrences.
- A bilingual glossary containing over 15,000 aligned entries. The granularity of the entries is heterogeneous and oscillates between a word, and an idiom including clauses.

4.2 Results and Discussion

ALINTEC proposed Basque target terms for 49% of the source Spanish MWT's. Of these alignments, 72.5% were entirely correct, 22.5% were partially correct and only 5% were incorrectly aligned. These results are reflected by means of *precision* and *recall* on Table 2. We labeled as "completely correct" those (CMWT-source, CMWTtarget) pairs which were validated by a human judge as correct translations. Those for which only parts of the term (not all the words in compositional or semi-compositional terms) were validated as correctly aligned were labeled "partially correct" pairs. Of the completely correct alignments (72.5%), we distinguished four cases depending on the resources that were used by the algorithm:

Table 2. Results of multiword terms alignment

Aligned pairs	Precision	Recall
(CMTsource, CMTtarget) pairs completely correct	72,5%	35,8%
(CMTsource, CMTtarget) pairs completely correct + partial correct	95%	46,91%

Case 1 There is just one CMWTtarget for one CMWTsource, Dice coefficient is grater than the threshold, and the grammatical patterns are equivalent. This is the case of 44.83% of the completely correct alignments.

Case 2 There are more than one CMWTtargets for one CMWT source and their Dice coefficients are grater than the threshold. In addition, at least one word from the CMWTsource and its correspondent word from the CMWTtarget appear in the glossary; the grammatical patterns are equivalent. This is the case of 34.48% of the completely correct alignments.

Case 3 The complete CMWTsource appears in the glossary with one of the CMWTtargets. Of the completely correct alignments only 13.79% fitted in this description.

Case 4 There are more than one CMWTtargets for one CMWTsource with the Dice coefficient greater than the threshold. No CMWT appears either completely or partially in the glossary. The pairs are selected by the criteria of equivalence between the syntactic patterns. This is the case for 6.89% of the completely correct alignments.

The results clearly show that the glossary has partially or totally supported by fifty percent (48.27%) of the correct alignments. The remaining fifty percent are used to enhance the glossary by adding the pairs of aligned terms that did not appear on it. Thus, this process of glossary feedback with the newly-aligned terms could increase the precision and recall values of subsequent alignments.

Two aspects must be discussed in order to evaluate these results. On one hand, the morpho-syntactic categories assigned by the taggers can vary appreciably between Basque and Spanish. For example, in *proyecto suscrito*, *suscrito* has been categorized as an adjective by the Spanish tagger, whereas in the corresponding *izenpe proiektu*, *izenpe* has been categorized as a verb by Basque tagger. In Table 1, we can see three Basque grammatical patterns (N. N., V. N., N. V.) for the Spanish pattern (N. AQ.). This makes the system admits pairs that in some cases are correct but in other cases introduce noise in the system. On the other hand and of no slight importance, the frequency values between correspondent pairs of terms show sensible differences. In fact, only 27.5% of the correct alignments have the same frequency values in Spanish and Basque. Table 3 shows a selection of the *ALINTEC* output with information about the values of the parameters and the resources that were used.

We have obtained reasonably good precision at the expense of recall. A forthcoming work will strive to improve of recall without reducing precision. We think

Table 3. A sample of the experimental results and the values of the alignment parameters

source TECs	Frq.	target TECs	Freq.	DC	Glossary	Gr.Pat. Sp≡Bq
adjudicación de la plaza	5	plaza adjudikazio	6	0.88	-	N.SP.TD.N. ≡ N.+
carácter general	9	izaera orokor	6	0.8	1	N. AQ. ≡ N. AQ.
desfigure la perspectiva	43	izaera desitxura	34	0.88	-	V. TD. N. ≡ N. V.
licencia de obras	44	obra lizentzia	36		Complete	N. SP. N. ≡ N. N.
otorgar la autorización	84	baimen eman	88	0.90	1	V. TD. N. ≡ N. V.
procedimiento de caducidad	43	iraungipen prozedura	34	0.88	1	N. SP. N. ≡ N. N.
recurso de reposición	28	birjarpen errekurtso	37	0.86	-	N. SP. N. ≡ N. N.
rompa la armonía	43	armonia hauts	37	0.88	-	V. TD. N. ≡ N. V.
acuerdo municipal	43	udal erabaki	34		Complete	N. AQ. ≡ N. N.
concurso ordinario	8	lehiaketa arrunt	9	0.87	-	N. AQ. ≡ N. AQ.

this will be possible through: (1) the enlargement of the equivalent syntactic patterns. We have not yet to carried out an exhaustive linguistic study of the syntactic structure of multiword terms in Spanish and Basque. Works as [14] can be of great help with respect to Basque language. (2) Assigning weights to the equivalent syntactic patterns. Thus, the level of compatibility between patterns could intervene for establishing the degree of reliability for a candidate target term given a source term. (3) Processing more and more parallel corpora in order to enrich the bilingual glossary.

5 Conclusions

We have tackled the alignment of multiword terms in a pair of languages, Spanish and Basque, with important structural, lexical and morpho-syntactical differences. The experiments have been accomplished using a truly parallel corpus with richer mark-up (POS tagger, lemmatizer, sentence alignment).

Our approach consists of a hybrid strategy that combines various levels of linguistic knowledge with quantitative criteria. The different levels of linguistic knowledge were obtained from: the Spanish-Basque corpus aligned at sentence level, POS annotations, grammatical patterns of Spanish and Basque multiword terms, and a bilingual glossary. Moreover, we have used the frequency and distribution of terms in the aligned corpus as quantitative criteria. Consequently, we have developed a tool, *ALINTEC*, that implements this approach obtaining a list of pairs of source and target multiword terms. *ALINTEC* was tested on and the results were analyzed in terms of the number of linguistic resources that were used. The experiments were carried out with a corpus within the administrative domain. If we take into account the differences between the pair of languages the results are encouraging. We believe that our approach in aligning multiword terms could be applicable to other pairs of languages of different typology. This it will be possible if the linguistics resources exit (mainly, a bilingual parallel corpus, POS taggers for the two languages, and a list of grammatical patterns of MWT's).

We have fulfilled to a significant extend the following objectives: (1) To propose an approach that deals with a pair of languages of different typology. (2) To propose an approach that can be used with a parallel corpus not being very large. Although at present the availability of parallel corpora is increasing, there are pairs of languages and domains with parallel corpora of small size. (3) To develop a prototype, *ALINTEC*, that facilitates the experimentation by means of varying the inputs that intervene in the alignment.

With regard to the future, we will focus on studying and setting different levels of compatibility between term patterns. We believe that it could prove significant in establishing the degree of reliability of a candidate target term given a source term. We will also try carry out experiments with others pairs of languages.

Acknowledgements

This research is being partially supported by the Spanish Research Agency, project HERMES, TIC-96-1243-C03-01. We greatly appreciate the help given to us by Joseba Abaitua, Iñaki Alegría and Felisa Verdejo. We also appreciate the help given to us by Paul Murphy for in reviewing the text.

References

1. I. Dagan, K. Church. Termight: Identifying and translating Technical Terminology. *Proceedings Fourth Conference on Applied Natural Language Processing (ANLP'94),* 34–40 (1994)
2. B. Daille, E. Gaussier, J.M. Lange. Towards Automatic Extraction of Monolingual and Bilingual Terminology. *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94)*, 515–521 (1994)
3. L.R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology,* 26:297–302 (1945)
4. P. van der Eijk. Automating the acquisition of Bilingual Terminology. *Proceedings Sixth Conference of the European Chapter of the Association for Computational Linguistic (EACL'93),* 113–119 (1993)
5. K. T. Frantzi, S. Ananiadou. A Hybrid Approach to Term Recognition. *NLP+IA96/TAL+AI96*, 93–98 (1996).
6. Fung, P., Church, K.W.. K-vec: A New approach for Aligning Parallel Texts. *Proceedings of the 15th International Conference on Computational Linguistic (COLING-94)*, 1096–1101 (1994)
7. Fung, P., McKeown, K. Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. *Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA'94)*, 81–88 (1994)
8. Justeson, J.S., Katz, S.M. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1), 9–27 (1995).
9. Kaji, H., Kida, Y., Morimoto, Y., Learning Translation Templates from Bilingual Texts. *Proceedings of the 13th International Conference on Computational Linguistics (COLING'92)*, 672–678 (1992)
10. B. Krenn, S. Evert. Can we do better than frequency? A case study on extracting PP-verb collocations. *Proceedings of the ACL Workshop on Collocations* (2001)

11. A. Kumano, H. Hirakawa. Building a MT dictionary from parallel texts based on linguistic and statistical information. *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94)*, 76–81 (1994)
12. I.D. Melamed: Automatic Discovery of Non-Compositional Compounds in Parallel Data. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing* (1997)
13. F. Smadja, K. McKeown, V. Hatzivassiloglou. Translating Collocations for Bilingual Lexicons: A Statistical Approach. *Computational Linguistics*, 22(1):1–38 (1996)
14. R. Urizar, N. Ezeiza, I. Alegría: Morphosyntactic structure of terms in Basque for automatic terminology extraction. *Proceedings of EURALEX 2000* (2000)
15. E. Viegas, S. Beale, S. Nirenburg: The Computational Lexical Semantics of Syntacmatic Relations. *Proceedings of the 17th International Conference on Computational Linguistics (COLING'98) and 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*, 1328–1332 (1998)
16. Watanabe, H., Kurohashi, S., Aramaki, E.. "Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation." *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, 906–912 (2000)
17. Y. Yamamoto, M. Sakamoto. Extraction of technical term bilingual dictionary from bilingual corpus. *IPSJ SIG Notes,* 94–12 (1993)
18. Yamamoto, K., Matsumoto, Y.. "Acquisition of Phrase-level Bilingual Correspondence using Dependency Structure." *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, 933–939 (2000)

Automatic Selection of Defining Vocabulary in an Explanatory Dictionary[*]

Alexander Gelbukh and Grigori Sidorov

Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Mexico City, Mexico.
{gelbukh,sidorov}@cic.ipn.mx

Abstract. One of the problems in converting a conventional (human-oriented) explanatory dictionary into a semantic database intended for the use in automatic reasoning systems is that such a database should not contain any cycles in its definitions, while the traditional dictionaries usually contain them. The cycles can be eliminated by declaring some words "primitive" (having no definition) while all other words are defined in terms of these ones. A method for detecting the cycles in definitions and selecting a minimal (though not the smallest) defining vocabulary is presented. Different strategies for selecting the words for the defining vocabulary are discussed and experimental data for a real dictionary are presented.

1 Introduction

A natural method to define the meaning of the words for an automatic reasoning system is to define some words through other words, the way it is done in the traditional explanatory dictionaries. To build such definitions, automatic conversion of existing explanatory dictionaries into "computer-oriented" dictionaries looks attractive. However, existing human-oriented dictionaries have a feature that does not permit to directly use them as logical systems: their definitions have logical cycles. For example:

(1) bee: an insect that produces honey.
honey: a substance produced by bees.[1]

There can appear longer cycles: a word *a* is defined though a word *b*, which is defined through a word *c*, etc., which is defined through the word *a*. Obviously, any dictionary that defines all words it mentions must contain cycles; thus, cycles are an inevitable feature of a human-oriented dictionary that tries to define all words existing in the given language.

[*] Work done under partial support of CONACyT, CGEPI-IPN, COFAA-IPN, and SNI, Mexico. Implementation by Gabriela Rivera-Loza. We thank Graeme Hirst and Ted Pedersen for useful discussion.

[1] This is a slightly simplified real example from the Anaya dictionary of Spanish: "*abeja: insecto que segrega miel*"; "*miel: sustancia que producen las abejas*."

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 300–303, 2002.

To convert such a dictionary into a "computer-oriented" logical database for a reasoning system, the cycles are to be eliminated by declaring some words "primitive," i.e., not defined in this dictionary (their meaning is to be stated in a different way).

For example, in school geometry it is possible to expand the definition of any term (say, *bisectrix*) substituting any word in its definition (say, *angle*) with its definition. This system of definitions is constructed in such a way that if such substitution is repeated iteratively, the definition of any term can be expanded into a (very long) definition composed only of the primitive concepts and logical operators. For school geometry, the primitive concepts are *point*, *line*, and *incidence*: these words do not have any definitions in the formal logical system of geometry.[2] We call such a set of primitive words *defining vocabulary*.

DEFINITION. Given a dictionary *D*, a *defining vocabulary* for *D* is a set of words such that if they are declared primitive (i.e., their definitions are removed from the dictionary), the rest of the dictionary does not contain cycles.[3] A defining vocabulary is *minimal* if no its subset is a defining vocabulary. A defining vocabulary is the *smallest* if there is no defining vocabulary for *D* consisting of a smaller number of words.

Indeed, for the same set of words, different defining vocabularies can be chosen. For instance, in the example (1) above, either *bee* can be chosen primitive and *honey* defined, or vice versa. Without going deep into discussion about the nature of semantic primitives (see, for example, [2–4]) or defining vocabulary (such as Longman defining vocabulary), we just note that the problem of selection of a defining vocabulary has so far no widely accepted theoretical solution. For the purposes of this paper it is important that since the meaning of the primitive words is explained in an "expensive" way (say, procedurally), it is highly desirable to minimize their number.

In this paper we present an algorithm that, given a dictionary, selects a minimal defining vocabulary for it. Our method, though, does not build the smallest defining vocabulary (this is the topic of our current investigation).

Below we present the algorithm, then discuss four strategies it can use, and compare them basing on the experimental data obtained with a large Spanish dictionary.

2 Algorithm and Strategies for Selection of Defining Vocabulary

We represent the dictionary as a directed graph ***G***, where the nodes are words[4] and there is an arc from w_1 to w_2 iff w_2 is used in the definition of w_1. Since ***G*** has cycles, our task is to select a minimal (not necessarily the smallest) set ***P*** of nodes such that removing from ***G*** all arcs leaving these nodes makes the resulting graph ***G'*** acyclic.

[2] Their meaning is explained to the students (the "users" of this formal system) by examples or procedurally (showing how to *draw a line* or how to observe that *two lines are incident*).

[3] The formal way we use the term "primitive word" does not completely correspond to the traditional use of this term in semantics [4]. In particular, the words selected by our algorithm as primitive might not be acceptable semantic primitives for a linguist.

[4] By a word, (1) literal string, (2) lemma, or (3) specific word meaning can be understood. The former variant nearly does not make sense. The results obtained for the latter two variants are very similar. Here we present the third variant (words as specific meanings). We used a disambiguation procedure similar to the Lesk algorithm [1].

Initially, both ***P*** and ***G'*** is empty. We consider the nodes of ***G*** one by one in some order (see below) and insert them into either ***P*** or ***G'***: if insertion of the node (and all arcs incident to it in ***G***) into ***G'*** does not cause any cycles in it,[5] the node is inserted into ***G'***, otherwise into ***P***. At the end of this process, ***G'*** is the desired acyclic graph and ***P*** is the corresponding minimal defining vocabulary.

There are different possible strategies to define the order of consideration of the nodes of ***G*** in the algorithm. The nodes considered first tend to belong to ***G'***, while the ones considered last tend to belong to ***P***, i.e., to be declared primitive (cf. example (1) above: if *honey* is considered first, *bee* is declared primitive, and vice versa). In our experiments we used four different strategies.

Strategy 1: *random, uniform*. At each iteration, the next node is chosen randomly (with a uniform distribution) from the nodes not yet processed.

Strategy 2: *by frequencies*. The nodes are ordered by the number of incoming arcs (i.e., frequencies in the definitions[6]), from smallest to greatest. We expected that with this, ***P*** would be smaller because the chosen nodes break more cycles in ***G***.

Strategy 3: *random, by frequencies*. This is a combination of 1 and 2. The random order is used, but with distribution inverse to the frequencies. We expected that some alternations of the rigid order of method 2 might produce better results.

Strategy 4: *by random voting*. $N = 20$ different sets $\mathbf{P}_i$ were generated with the strategy 1. Then for each node w we counted the number $0 < n(w) < N$ of sets $\mathbf{P}_i$ to which it belonged. In the algorithm, we considered first the nodes w with $n(w) = 0$ in the order of their frequencies, as in the strategy 2. Then we considered the rest of the elements in the order inverse to $n(w)$, in each group with the same $n(w)$ using the order inverse to the frequencies. We expected that the nodes with a greater $n(w)$ were better defining words and, thus, should belong to the best defining set.

3 Experimental Results

We applied these strategies to a large explanatory dictionary of Spanish (Anaya dictionary). Before this, auxiliary words had been removed and the content words in the definitions had been lemmatized, POS-tagged, and marked with sense numbers; see [1] for details. The dictionary contained 30725 headwords; 10359 words were used in the definitions (i.e., these words had incoming arcs).

Table 1. Number of primitives obtained with different strategies

Strategy	Size of ***P***	Strategy	Size of ***P***
1. Random, unified	2789, $s = 25$	3. Random, by frequencies	2770
2. By frequencies	2302	4. By random voting	**2246**

[5] A special data structure is used to quickly verify this, which guarantees linear complexity of the whole process.

[6] Without repetitions in the same definition, i.e. several occurrences of a word in the same definition is counted as one occurrence.

Table 1 shows the experimental results. The values given for the first strategy are the average calculated during 20 experiments and the mean quadratic deviation *s*. It is interesting that the deviation is rather small, which means that there is little difference in size of the sets generated in different experiments according to this strategy. As one can see, the best strategy is 4.

Surprisingly, the size of the defining vocabulary we found is very near to 2000, which is considered an approximate number of primitives in human languages. For example, this is the size of the Longman Defining Vocabulary, the number of basic glyphs in Chinese, etc.

4 Conclusions and Future Work

We have presented a method for selecting, given a dictionary *D*, a minimal (though not the smallest) defining vocabulary. The strategies of ordering to be used in the algorithm have been discussed.

Construction of defining vocabularies is helpful in analysis of the structure of dictionaries by the lexicographers, in particular, to evaluate the optimality of relations between words in the dictionary.

The possible future work is the following:

– To give a linguistic interpretation of the obtained defining vocabularies,
– To elaborate linguistic (semantic, rather than statistical) criteria of preferences of inclusion of words into the defining vocabulary (see footnote 3 above),
– To improve the algorithm to find the smallest defining vocabulary; different techniques—for example, genetic algorithms—can be used,
– To develop the software that would allow using the obtained information to help lexicographers in improving (traditional) dictionaries.

References

1. Gelbukh, A., and G. Sidorov (2001). Algorithm of word sense disambiguation in an explanatory dictionary. Proc. of *COMPLEX-2001, Workshop on Computational Lexicography*, Birmingham, Great Britain, June 28-30, 2001, pp. 35-40.
2. Kozima, H., and A. Ito (1997). Context-sensitive word distance by adaptive scaling of a semantic space. In: Mitkov, R. and , N. Nicolov (eds.). *Recent Advances in Natural Language Proceedings: Selected Papers from RANLP'95*, pp. 111-124.
3. Saint-Dizier, P., and E. Viegas (eds.). (1995) *Computational lexical semantics*. Cambridge: Cambridge University Press, 447 p.
4. Wierzbicka, A. (1996) *Semantics: Primes and Universals*. Oxford: Oxford Univ. Press.

Integrated Natural Language Generation with Schema–Tree Adjoining Grammars

Karin Harbusch and Jens Woch

University of Koblenz-Landau,
Computer Science Department
{harbusch,woch}@uni-koblenz.de

Abstract. This paper describes an *integrated generation system (INLGS)* based on the formalism of *Schema Tree Adjoining Grammars with Unification (SU–TAGs)*. According to this system architecture, all knowledge bases are specified in the same formalism and run the same processing algorithm. A main advantage is that *negotiation between generation components* can easily be imposed on the system. Moreover, only one algorithm must be implemented and tested in order to provide the one and only processing unit. In the INLGS a *reversible parser/generator* is deployed. It runs knowledge bases in the formalism of SU-TAGs. SU–TAG comprises a condensed grammar representation and *direct parsing/generation* deals with *partially unspecified schemata*. Instead of developing new knowledge bases from scratch, existing ones are *reused* here. This means all knowledge bases of the INLGS are transformed (e.g., the TAG–based XTAG system and the plan–based interpersonal model of VOTE).

1 Introduction

This paper describes an *integrated natural language system (INLGS)* based on the formalism of Schema-Tree Adjoining Grammars with Unification (*SU–TAGs*). According to this system architecture, all knowledge bases are specified in the same formalism. A main advantage of INLGSs is that *negotiation strategies* on revisions can easily be imposed on the system. This means, any *communication* between generation components (e.g. formulator and conceptualizer) is modeled *implicitly* by overall decision making and backtracking according to concurrent rules taken from the individual knowledge bases.

All knowledge bases of the INLGS are specified in SU–TAGs in which *schemata* fold up subtrees and depict them in terms of *regular expressions (RXs)*. Hence, Schema–TAGs provide a more condensed grammar representation than TAGs. Unification features are attached to nodes as in ordinary TAGs. With the close relation to TAGs with Unification (U-TAGs), it is obvious that SU–TAGs are powerful enough to perform the how–to-say task (e.g., TAG–based descriptions by [Becker *et al.*, 1998], [Nicolov, 1998], [Stone & Doran, 1997], [Webber & Joshi, 1998]) as well as the what-to-say task (Sec. 3) in generation.

In an INLGS, its knowledge bases are of essential importance because the system is inherently declarative. Since it is very time–consuming to develop

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 304–313, 2002.

knowledge bases from scratch existing ones are reused instead. An automatic transformation for TAG format (e.g., the XTAG system [Doran *et al.*, 1994]) and plan–based format (e.g., the interpersonal model in VOTE [Slade, 1994]) has been developed. For ordinary TAGs, the transformation is near at hand. For plan–based descriptions such as in many what–to–say components, the transformation rewrites plan steps and programming language statements by schemata whereas conditions in statements and the pre– and post–conditions are wrapped up in feature specifications.

The only processing unit of the outlined integrated system is a *bidirectional direct parser/generator for SU–TAGs*, which is based on the TAG parser by [Schabes, 1990], which in turn is based on Earley. In order to parse the condensed tree schemata, Earley parsing explores the regular expressions, too. Consequently, the schemata remain partially unspecified as long as no evidence for concrete substructures exists. Thus, combining *direct* parsing/generation strategies with Schema-TAGs lead to a better average case because fewer items are produced (Sec. 4). However, the worst–case behavior remains the same as for TAGs (see [Schabes, 1990]).

The INLGS' parser is parameterized to work for generation adapting the idea of *bidirectional processing* (cf. [Neumann, 1994]). The generator works *input–driven*, i.e. it predicts *semantic heads.* According to [Shieber *et al.*, 1990] this means, two different procedures continue searching for a connection to sub– and the super–derivation trees. The generator predicts *pivots*, i.e. the lowest nodes in the tree such that it and all higher nodes up to the root node or a higher pivot node have the same semantics. The rules according to this path are called *chain rules.* All other rules belong to the set of *non–chain rules.* Appying this idea to (SU–)TAGs, the *chain rules* directly correspond to the elementary lexicalized trees. Adjoining and substitution represent the application of the non–chain rules.

In the following first we address the system architecture (Sec. 2). In Sec. 3, the underlying formalism of SU–TAGs and the reuse of existing knowledge bases is outlined. Afterwards the bidirectional direct parser/generator is depicted (Sec. 4). In the final section we sum up and address future work.

2 Integrated System Architecture

The idea of integrated natural language generation goes back to [Appelt, 1985] (cf. KAMP). In this section we argue how we deploy such a system architecture. The actually underlying formalism is not of particular interest in this section. We take for granted that it is powerful enough to perform any task in the what–to–say and how–to–say part of a generation system[1].

[1] If intermediate results of the processing in a component are concerned we shall use the terms *derivation* and *rule* because our system is grammar–based. However, our claim also holds, e.g., for an achieved goal according to the planning process of the basic plans and the world knowledge (cf. KAMP).

In an INLGS, any knowledge base is specified in the same formalism and represents a generation component. This means, that the knowledge bases provide input to the one and only generator. Thus, the code of the generator has to be written and tested only once. However, this architecture is advantageous for more important reasons. In an INLGS, any *communication* between the generation components is modelled *implicitly*. Supposing all knowledge bases become active as soon as possible, an *incremental generation system* results (cf. KAMP). Here, the currently valid intermediate structures serve as a *blackboard* where all currently applicable rules of the knowledge bases try to prevail.

More elaborate communication strategies influencing the decision making are exploited according to the following two guidelines: order of applying knowledge bases and elaborate backtracking strategies. These guidelines are defined by the user.

According to the first one, e.g., the application of a focus rule possibly determines a sentence in passive voice. All following decisions have to coincide with this decision. Moreover, this guideline can be applied within an individual knowledge base by hierarchically structuring it in order to satisfy basic constraints before more fine grained rules may be applied. Hence, the INLGS does not activate individual components and exchange output structures. Loosely speaking, demons, i.e. always/user defined active processes which try to add their rules of an individual knowledge base, modify the blackboard (similar to hierarchical constraint satisfaction).

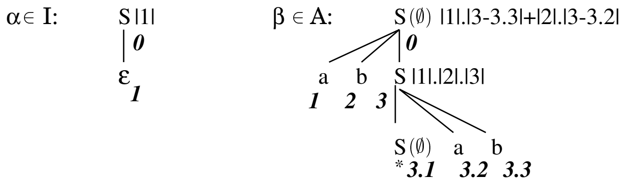
Elaborate backtracking strategies, on the other hand, resolve *dead–ends*, i.e. obtain communication between the generation components. Let us assume an intermediate structure cannot be further exploited, i.e. it cannot deploy an overall derivation. In an INLGS, rules of the individual knowledge bases coming into question for backtracking are compared with respect to the hierarchical constraint satisfaction and user–imposed preferences. Thus, backtracking represents the communication whether specific information must be revised in order to continue the processing. A main difference in comparison to an explicit communication language is that no hypotheses of how to resolve the conflict are generated here. Our claim is that a component with its local knowledge generally cannot propose a reasonable solution to another component with completely disjunct knowledge. Therefore this kind of guidance is basically replaced by hierarchical constraint satisfaction. Generally speaking, an INLGS can remedy the *generation gap* (see [Meteer, 1990]) without defining an explicit communication language (see [Harbusch & Woch, 2000b]).

In the next section we delineate SU–TAGs and outline the transformation of existing knowledge bases.

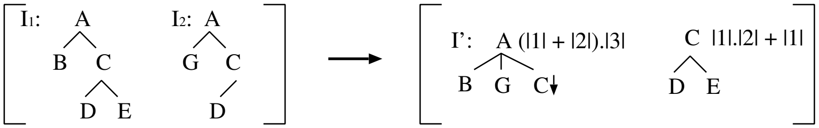
3 Reuse of Knowledge Bases

Schema TAGs were introduced by [Weir, 1987] in order to compress syntactic descriptions. For that purpose, inner nodes of elementary trees (see [Joshi & Schabes, 1997]) obtain *regular expressions (RXs)*, which refer to the inner node's children by their Gorn Number [Gorn, 1967]. Operations on σ and σ', being two RXs, are $\sigma + \sigma'$ (alternatives), and $\sigma.\sigma'$ (concatenation). Additionally, if σ is a RX, then (σ) (bracketing), σ^* (iteration with $n \in \mathbb{N}_0$) and $\sigma^{(0|m)}$ (iteration with $n \in [0..m], m \in \mathbb{N}$) are RX, too. Furthermore, we abbreviate σ^n for $\sigma. \ \ldots \ .\sigma$ (n times), σ^+ for $\sigma.\sigma^*$ and $\sigma^{(n|m)}$ for $\sigma^n.\sigma^{(0|m-n)}$. If $n \in \mathbb{N}$ and a Gorn number g depicts a subtree of n, $|n - g|$ is a RX (*elimination construction*: the subtree rooted by g is eliminated and replaced by the empty leaf (ϵ)). Since RXs enumerate trees, constraints are necessary in order to fulfill the TAG definition, i.e., foot nodes must not be cut off or duplicated.

A (SU–)TAG with *local constraints* restricts the set of adjoinable tree schemata or substitutional tree schemata[2]. *Selective adjoining* [SA] licenses the adjoining of a subset of auxiliary tree schemata, in *obligatory constraint* [OC] at least one adjunction/substitution according to the specified subset has to be explored and *null adjoining* [∅] prohibits any adjunction. The following figure gives an example of a SU-TAG with local constraints.



Now the transformation of different formats of existing knowledge bases into the SU–TAG format is shown. First, we briefly delineate the transformation of an ordinary TAG. The TAG–to–SUTAG–transformation produces a SU–TAG where each label at the root node occurs only once in the set of initial and auxiliary trees (i.e. compression is enforced). The component performs the following steps. In all elementary trees all subtrees are rewritten by substitution in order to find small shared structures. The new substitution nodes are associated with obligatory constraint of the tree cut off to prevent the grammar from unintended overgeneration. Now, all initial and auxiliary trees with the same root label are summed up in two schemata enumerating all existing branches and yielding all actual trees in terms of alternatives in a RX. The resulting RXs can be reformulated applying the law of distributivity. The following figure depicts the condensation of a simple TAG:



[2] Only obligatory substitution is defined because null and selective substitution produces incomplete derived trees.

Note, that different compressing strategies result in different RXs. For generation, the factorization of common heads is more adequate. The individual strategies can be parameterized in the INLGS.

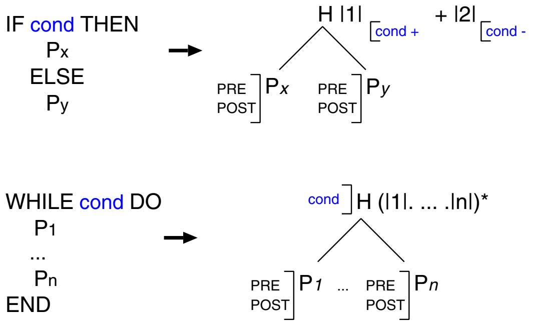
On the basis of this procedure, XTAG [Doran *et al.*, 1994] was transformed into a SU–TAG. The knowledge bases of SPUD [Stone & Doran, 1997], and those described in, e.g., [Becker *et al.*, 1998] or [Webber & Joshi, 1998] can be transformed into a SU–TAG applying the same method. Doing so, the generator is extended towards a *generation workbench* which provides libraries of knowledge bases from which the user can select a personal generation system.

Concerning the knowledge bases of a what–to–say component, we only concentrate on the particular class of plans which is widely applied in generation systems, i.e. the classical *plan–based plans* (cf. [Yang, 1997]). As an illustration a plan of VOTE [Slade, 1994] is transformed here.

A *plan* consists of n steps, any of them in turn may be an *action* or a plan again. Each step consists of *pre-* and *postconditions*, as well as controlling elements of a programming language (e.g. `IF-THEN-ELSE`). Assuming a current situation and a goal, a plan can successfully be applied, if its preconditions match the current situation and its postconditions match the goal. As long as plan steps are non–atomic they are replaced by the according plan. If a plan step is atomic, i.e. an *action*, the action is performed by replacing the preconditions with the postconditions in the current situation.

Given that, the PLAN–to–SU–TAG tranformation consists of the following steps:

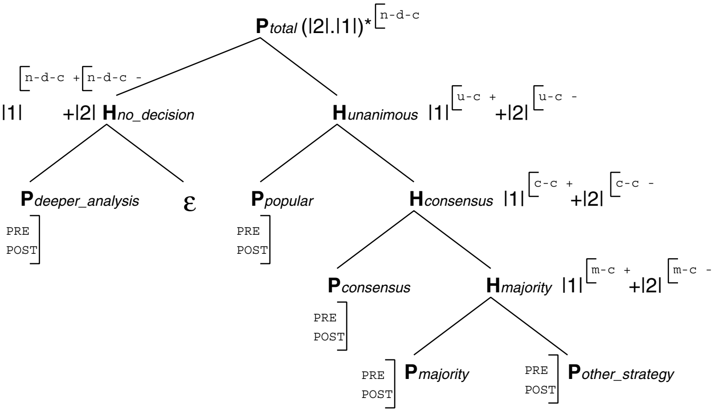
1. Each plan step in a sequence becomes an individual node of an elementary scheme under a common root node.
2. The chronological sequence of plan steps is rewritten via concatenation in the respective RX.
3. Pre- and postconditions of plans are wrapped up in feature specifications at the corresponding nodes.
4. The conditions of controlling elements of the programming language are realized by unification too, whilst the branches and repetitions itself are transformed into RXs. That is, Px and Py in the alternative of the figure below are rewritten as siblings and enumerated as alternatives (with their appropriate feature sets), whereas the iteration is rewritten as an infinitely repeated sequence.



The behavior of the steps 1, 2 and 4 is exemplified in the following figure[3]. This plan describes the decision making process in the system VOTE.

According to step 4, `IF-THEN-ELSE` statements are converted into binary sums which represent the choice of one branch. The individual conditions are exploited by features (e.g., `u-c=+` for `?Unanimous` at its `THEN`–branch RX). `WHILE` in line (1) is transformed into a Kleene Star, which stops as soon as its condition in the feature `n-d-c` at the root node is satisfied. At the same node, the concatenation represents the sequence of the two `IF-THEN-ELSE` statements in line (2) and (8) (step 1 result in |2|.|1| according to the order of branches).

```
(1) WHILE ?no--decision
(2)   IF ?Unanimous
(3)     THEN Plan$_{Popular}$                         $\Longrightarrow$
(4)     ELSE IF ?Consensus
(5)       THEN Plan$_{Consensus}$
(6)       ELSE IF ?Majority THEN Plan$_{Majority}$
(7)                                   ELSE Plan$_{Other-Strategy}$;
(8)   IF ?no--decision THEN Plan$_{Deeper-Analysis}$;
(9) DO
```



After transforming the plans of VOTE the facts of VOTE are to be transformed too, but this is straightforward, so we omit that here.

4 Generator

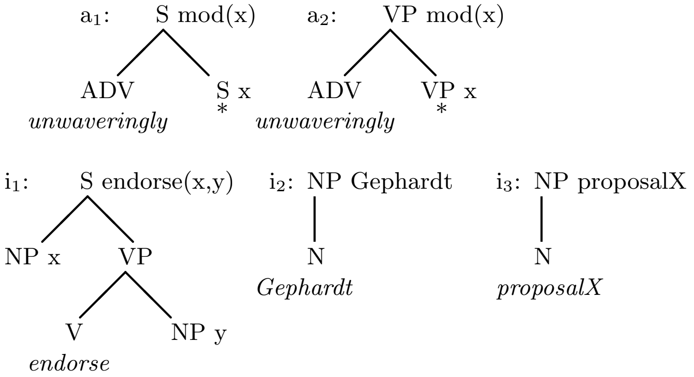
In an INLGS, first a direct parser for SU–TAGs has been developed and extended towards a *bidirectional parser/generator* because for testing the knowledge bases a parser is indispensable even in a generation system.

We deploy a *direct parser* because it explores schemata, i.e yields partially unspecified rules. An ordinary TAG parser presupposes the enumeration of all

[3] Since the evaluation of pre- and postconditions require further knowledge of VOTE (in the SU–TAG format) these features are skipped here (for more details see [Harbusch & Woch, 2000c]).

elementary trees according to schemata up to a maximum length (termination condition for infinite schemata). Hence, the condensation of the grammar gets lost. Instead, the direct parser exploits regular expressions in an Earley–based manner. This means, a new dot position ($\odot$) indicates whether a prefix of an alternative in a RX is already analysed. Traversing RXs replaces the analysis of branches in the six procedures of the TAG parser by Schabes [Schabes, 1990] in a straight–forward manner. For more details of the extensions of the individual procedures of the Schabes parser see [Harbusch & Woch, 2000a]. The worst–case time and space complexity remains the same as for ordinary TAGs ($O(n^6)$ and $O(n^4)$, respectively). In the average case, underspecification reduces the overall number of items.

This parser is *bidirectional.* For reasons of efficiency we make the generation process driven by the *semantic input structure* (indexing on meaning instead of indexing on string position). Generally speaking, such a generator predicts semantic heads. According to [Shieber *et al.*, 1990], two different procedures continue searching for connections to sub– and the super–derivation trees. The two search directions apply different rule sets. Here, the *pivot* is defined as the lowest node in the tree such that it and all higher nodes up to the root node or a higher pivot node have the same semantics. Accordingly, the set of *chain rules* consists of all rules in which the semantics of some right–hand side element is identical to the semantics of the left–hand side. All other rules belong to the set of *non–chain rules.* The traversal works top–down from the pivot node only using non–chain rules whereas the bottom–up steps, which connect the pivot with the root node, only use chain rules.



Adapting this mechanism to the generation of lexicalized SU–TAGs means that the chain rules are equal to the elementary tree schemata. Adjoining and substitution represent the application of non–chain rules. In order to illustrate this kind of processing let us assume the input structure *(unwaveringly(endorse (Gephardt, proposalX)))* provided according to the lexical choice in VOTE (see [Slade, 1994]:235) and the corresponding predicted chain rules. Here the foot nodes are marked with "*" in order to distinguish them from substitution nodes.

To keep the example simple, schemata are omitted here. For instance, i_1 is supposed to yield the extraposition of the object as well as a subject–verb–object word order. Constraints ensure that exactly one element is extraposed. Hence, a_1 does not coincide with the extraposition of the object (see figure above). In this example the semantics of the trees are informally annotated at the nodes where x and y are variables to be filled during the unification at that node.

In a first step all predictable chain rules are written to the one and only item set during processing. Due to this fact the structures can combine in any order. The bracketing structure of the logical form is achieved by evaluating the semantic expression associated with each elementary tree (e.g., for tree a_1 $mod(x)$ where x is a value filled by the subtree below the foot node). The processing is successful if a derived tree can be constructed where all elements of the logical form occur exactly once (since the bracketing structure is tested explicitly during the combination of elementary trees, the accept condition can be weaker without raising the logical form equivalence problem (cf. [Shieber, 1993]). Concerning the example two realizations for the input specification can be produced. The adjoining of the sentential adverb (a_1) is obvious whereas the semantic licensing of a_2 is not: Here, its variable x at the foot node is unified with the VP node of i_1 where according to the pivot definition the semantics on the spine from the root to the V node is identical. So, x contains the whole expression *(endorse(Gephardt,proposalX))*. The bracketing structure is correct regardless whether the unification checks the variables monotonically or not. Hence, the utterance is well-formed as well.

Let us now sum up and address future work.

5 Conclusions

In this paper we have given a sketch of the INLGS' architecture. Its integrated approach, i.e. all knowledge bases are specified in the same formalism (S-TAG with Unification) allows for

- the extension with additional KBs without restructuring the system's procedures,
- implicit communication between concept and formulation without an explicit communication language, and
- the parameterization of the rule-selection process, which in turn allows the adaption of the generation process to different psycho-linguistic models.

The knowledge bases are not developed from scratch but an automatic transformation from TAG– and plan–based formalisms into SU-TAGs is deployed. The one and only processing unit comprises a reversible direct SU–TAG parser/ generator which yields partially unspecified schemata in order to improve on the average case.

An open task is to test strategies to parameterize the generation unit to conduct a hierarchy of imposed knowledge bases as well as hierarchies within the knowledge bases. Further parameters to guide the exploitation of rules are under consideration, because a problem is that the specification must remain simple such that a user is able to keep track of the parameter.

Currently, the INLGS is tested as a generation component within a natural language driven automated help desk, which is kind of an automated information service, where customers can call to ask questions and so forth. Whithin this system the INLGS is responsible for the generation of the system's responses, further inquires, and solution descriptions.

References

Appelt, 1985. D. E. Appelt. *Planning English Sentences.* Cambridge University Press, Cambridge, MA, USA, 1985.

Becker *et al.*, 1998. T. Becker, W. Finkler, A. Kilger & P. Poller. An Efficient Kernel for Multilingual Generation in Speech–to–Speech Dialogue Translation. In [Isabelle, 1998], pp. 110–116.

Doran *et al.*, 1994. C. Doran, D. Egedi, B. A. Hockey, B. Srinivas & M. Zaidel. XTAG System — A Wide Coverage Grammar for English. In M. Nagao, ed., *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, vol. 2, pp. 922–928. Kyoto, Japan, 1994.

Gorn, 1967. S. Gorn. Explicit definitions and linguistic dominoes. In J. Hart & S. Takasu, eds., *Systems and Computer Science*, pp. 77–115. University of Toronto Press, Toronto, Canada, 1967.

Harbusch & Woch, 2000a. K. Harbusch & J. Woch. Direct Parsing of Schema–TAGs. In H. C. Bunt, ed., *Procs. of the 6th International Workshop on Parsing Technologies (IWPT)*, pp. 305–306. Institute for Scientific and Technological Research, Trento, Italy, 2000a.

Harbusch & Woch, 2000b. K. Harbusch & J. Woch. Modelling Communication between Conceptualisation and Formulation in an Integrated Generation System. In *Procs. of the 22nd Annual Conference of the Linguistic Association of Germany (DGfS).* Philipps–Universität Marburg, Marburg, Germany, 2000b.

Harbusch & Woch, 2000c. K. Harbusch & J. Woch. Reuse of Plan–Based Knowledge Sources in a Uniform TAG–based Generation System. In A. Abeillé & G. Satta, eds., *Procs. of the 5th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+5)*, pp. 245–248. University of Paris 7, Paris, France, 2000c.

Isabelle, 1998. P. Isabelle, ed. *Procs. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics.* Université de Montréal, Canada, Morgan Kaufmann, 1998.

Joshi & Schabes, 1997. A. K. Joshi & Y. Schabes. Tree Adjoining Grammars. In G. R. A. Salomaa, ed., *Handbook of Formal Languages*, vol. 3, pp. 69–214. Springer, Berlin, Heidelberg, Germany, 1997.

Meteer, 1990. M. W. Meteer. *The Generation Gap – The Problem of Expressibility in Text-Planning.* Ph.D. thesis, University of Massachusetts, Amherst, MA, USA, 1990.

Neumann, 1994. G. Neumann. *A Uniform Computational Model for Natural Language Parsing and Generation.* Ph.D. thesis, University of the Saarland, Saarbrücken, Germany, 1994.

Nicolov, 1998. N. Nicolov. Memoisation in Sentence Generation with Lexicalized Grammars. In A. Abeillé, T. Becker, O. Rambow, G. Satta & K. Vijay-Shanker, eds., *Procs. of the 4th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+4)*, pp. 124–127. Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA, USA, 1998. IRCS-Report 98–12.

Schabes, 1990. Y. Schabes. *Mathematical and Computational Aspects of Lexicalized Grammars.* Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA, 1990.

Shieber, 1993. S. M. Shieber. The Problem of Logical–Form Equivalence. *Computational Linguistics*, 19(1):179–190, 1993.

Shieber *et al.*, 1990. S. M. Shieber, G. van Noord, F. C. Pereira & R. C. Moore. Semantic Head-Driven Generation. *Computational Linguistics*, 16(1):30–42, 1990.

Slade, 1994. S. Slade. *Goal–Based Decision Making: An Interpersonal Model.* Lawrence Erlbaum Associates Inc., Hillsdale, NJ, USA, 1994.

Stone & Doran, 1997. M. Stone & C. Doran. Sentence Planning as Description Using Tree Adjoining Grammar. In *Procs. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL).* Universidad Nacional de Educacion a Distancia, Madrid, Spain, 1997.

Webber & Joshi, 1998. B. L. Webber & A. K. Joshi. Anchoring a Lexicalized Tree–Adjoining Grammar for Discourse. In [Isabelle, 1998].

Weir, 1987. D. Weir. Characterising Mildly Context–Sensitive Grammar Formalisms. PhD. proposal, University of Pennsylvania, Philadelphia, USA, 1987.

Yang, 1997. Q. Yang. *Intelligent Planning.* Springer-Verlag, Berlin, Germany, 1997.

Experiments with a Bilingual Document Generation Environment

Arantza Casillas[1] and Raquel Martínez[2]

[1] Dpt. de Electricidad y Electrónica, Facultad de Ciencias,
Universidad del País Vasco (UPV-EHU)
arantza@we.lc.ehu.es
[2] Escuela Superior de CC. Experimentales y Tecnología,
Universidad Rey Juan Carlos
r.martinez@escet.urjc.es

Abstract. This paper presents a bilingual generation enviroment and shows the results obtained after evaluating the enviroment with two different groups of documents.

1 Introduction

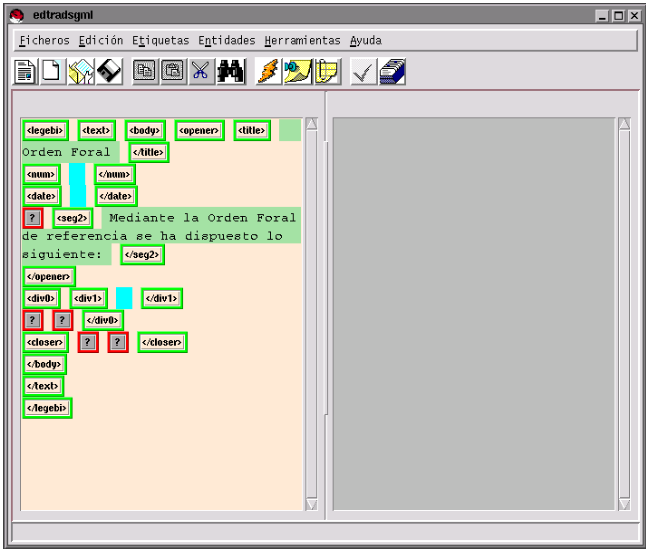
Producing bilingual documentation within specialised domains is a very time-consuming and expensive process. It is furthermore a relatively unautomated task, in spite of its potentialities. In the manual process it involves both human writers and translators, who devote endless efforts in a constant recycling of repetitive and reusable text chunks. The main desire of institutional writers as well as translators is to ascertain quickly how a recurrent text (whether memo, resolution, announcement, etc.) has been previously composed to save the effort of attempting a novel and a possibly problematic unseen version. Textual variations and divergences are not much appreciated in specialised documentation. When there is evidence that a similar document might have been previously written or translated, they take pains to find it in the normalised version. We have developed an authoring tool which directs the generation of both the source text and the target text.

In the next section we will describe the enviroment; the obtained results will be shown in section 3 and finally, we will sumarize the conclusions in section 4.

2 Enviroment Description

Our generation enviroment is based on the ideas of [1], [2]. The authoring environment directs the generation of both the source text and the target document through a planification process of the logical order of the document elements and their content. Two levels of text generation may be considered. There is a strategic level of decision which permits to organise the logical structure and content of document elements. The tactic level comes afterwards, whereby the syntax and words phrasing plan are selected.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 314–317, 2002.



```
<legebi> <text> <body> <opener> <title>
Orden Foral <title>
<num> </num>
<date> </date>
? <seg2> Mediante la Orden Foral
de referencia se ha dispuesto lo
siguiente: </seg2>
</opener>
<div0> <div1> </div1>
? ? </div0>
<closer> ? ? </closer>
</body>
</text>
</legebi>
```

Fig. 1. Example of the enviroment

From a bilingual aligned corpus we automatically extract: (1) Several translation memory databases (2) The logical structure of documents and theirs syntax in the form of a context freee grammar (this is called Document Type Definition or DTD). Every phase in the process is guided by the DTD and the content of the databases. The composition process follows two main steps which are the traditional source document generation and translation into target process. According to stored DTD models the logical structure and content are suggested to the user.

Before the user starts writing the source document, a DTD must be selected. There is a DTD for each type of document. The user selects the document type s/he wants to create and automatically the corresponding DTD is selected. This process has two consequences: on the one hand, the selected DTD produces a source document template that contains the logical structure of the document and some of its contents if any contents for the logical element is in the database (to know if one element has any content in the database we use the corresponding DTD element name). On the other hand, the selected source DTD triggers a target paired DTD, which will be used later to translate the document. Once the source document has been completed, the system derives its particular logical structure, which, with the aid of the target DTD, is projected into the resulting target logical structure. Starting from the source DTD in Spanish, institutional writers have a document scheme containing either the content of some of the elements or optional elements to choose from, in case there are more than one solution. These elements are the translation units. Fig. 1 shows an example of source document scheme. The enviroment uses different colours to make a distinction between types of elements.The user can modify the contents proposed

Table 1. Document % generated by our enviroment

Word/doc.	Num. doc.	Source	Target
0-500	378	24	36.5
500-1,000	25	11.9	21.3
More 1,000	16	2.4	11.2
Weighted Mean		22.5	34.6

by the enviroment or s/he can add new ones. We can update the translation memories with the new translation units detected. As well we can improve the accuracy of the translations memories with new bilingual documents or new bilingual aligned corpus. We think that the enviroment will work with documentation that: (1) shows regular structure, (2) is rich in recurrent textual patterns within specialized domain and (3) is multilingual or bilingual.
Memory-based systems facilitate the reutilization of aligned translation units, but they neglect any information refering to the logical structure of the text. Precisely, our guiding hypothesis has been the idea of providing the basic document scheme using DTDs. These DTDs determine the logical structure of documents.

3 Evaluation

To evaluate the efficency of the developed enviroment we have analyzed two different corpus:

- The first corpus has over 500,000 words of administrative publications in Spanish and Basque. Table 1 shows four columns: (1) size of document, (2) number of documents for each size, (3) percent of generated source document, (4) percent of target generated document. We can see that short documents (90.21%) have about 24% generated in the case of the source document and 36.5% generated for the target document. This figure goes down to 2.4% in source documents and to 11.2% in target documents containing more than 1,000 words. This is undestandable in the sense that the larger the document, the larger the proportion of textual divergences (or smaller the proportion of fixed sections) it will contains. If a content has not been stored in the translation memory, it cannot be generated or translated.
- A second corpus. The evaluation was based on analyzing how much corpus could be translated using the databases and DTDs created from the firts corpus (see table 2). This corpus has around 200,000 words of official publications. We suppose that the continuos updating of the different translation memory databases will increase the rate of translated document up to 21.92%.

We haven't compared the obtained results with standard translation memory system because: (1) On the one hand, there is not similar commercial software. Standard translation memory systems only translate the source document that

Table 2. Document % generated by our enviroment

Word/doc.	Num. doc.	Total
0-500	129	24.63
500-1,000	11	11.64
More 1,000	14	5.1
Weighted Mean		21.92

has been created by the user. Our enviroment generates part of the source document and part of the target document. (2) On the other hand, we detect many different translation units (such as terms, proper nouns, sentences, numbers, dates, segments that are frecuently used in administrative documentation, abbrebiations and so on) and standard translation memory systems only work with sentences and, in some cases, with terms. We can assert that with a standard translation memory system, which detects the same type of translation units as we do, the percent of text translated would be the same. The difference lies with logical elements position in the target document. A standard translation memory system doesn't know where is a logical target element. The DTD specifies the logical possition of the target element. DTDs contribute to locate the elements in the source and target document.

4 Conclusions

In this paper we have shown a generation enviroment for bilingual documents. This tool is capable of handling a substantial proportion of text both in the composition and translation of documents. On average, one fourth of the source document and one third of the target document can be automatically generated. Our enviroment, compared with standard translation memory systems, has this advantages: (1) Generates the source and target document structure (using DTDs). (2) Porposes contents for the source document. (3) Uses translation units that are not only sentences and terms. This approach permits increase the precentage of document translated.

Acknowledgements

This research is being partially supported by the Spanish Research Agency, project HERMES (TIC2000-0335-C03-03). We greatly appreciate the help given to us by Joseba Abaitua, Iñaki Alegria and Felisa Verdejo.

References

1. Kay M.: The Proper Place of Men and Machines in Language Translation. Machine Translation, 12:3–23, 1997.
2. Melby A. On human-machine interaction in translation. Machine Translation, 145–154, 1987.

A Computational Model of Change in Politeness with the Addition of Word Endings

Tamotsu Shirado and Hitoshi Isahara

Communications Research Laboratory,
4-2-1 Nukui-kita, Koganei city, Tokyo 184-8795, Japan
{shirado,isahara}@crl.go.jp

Abstract. Many polite expressions can be synthesized using simple expressions in Japanese. It is expected that such syntheses are dominated by psychological mechanisms in politeness. This study reveals one such mechanism by using a computational model for politeness in Japanese to describe changes in politeness through the addition of word endings. In this model, two stochastic features are assumed: (1) For each expression, a situation in which the expression would be used can be represented by a probability distribution of the politeness value in a psychological space, and (2) For each word ending **e**, a probability distribution exists in a one-dimensional psychometrical space of politeness, where the distribution represents the ideal distribution of the most suitable (or ideal) expression to which the word ending **e** would be added. The change in politeness that arise from the addition of word endings is calculated by the difference between these probability distributions. The information theory is utilized in this calculation. A linear relationship is expected to exist between the change in politeness that arise from the addition of word endings to expressions and the politeness of the original expressions. Psychological experiments were performed to verify the validity of the model. The degree of politeness of expressions was evaluated using Thurstone's scaling. Experimental results show the expected linearity to be true which qualitatively verifies the validity of the model. The results are also discussed in terms of linguistic intuition.

1 Introduction

In Japan, politeness plays an important role in social activities, particularly in conversation. Japanese speakers tend to choose different expressions with different levels of politeness, but with the same speech intentions, when the listeners and/or the topic persons change. This type of speech strategy for choosing expressions is dependent on social relationships such as power differences (eg. age difference), and/or social distances (eg. familiarity) amomg them. For example, the speakers tend to use polite expressions when the listener is older than them, and they tend to use impolite expressions when they are familiar with the listener. The latter is true if the listener is older than them. Let us call this type of social relationship "*politeness – relationship*," hereafter.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 318–326, 2002.

Many Japanese expressions that have different levels of politeness, but the same speech intentions, can be derived using a few simple expressions. One way to derivational expressions is the addition of word endings to simple expressions. For example, a simple expression with the speech intention "I know": "$shitte - ru$" becomes a less polite expression: "$shitte - ru - yo$" by simply adding the word ending "*yo*" (impolite). However, it becomes a more polite expression: "$shitte - masu$" by adding the word ending "*masu*" (polite). Furthermore, the expression: "$shitte-masu$" becomes a less polite expression: "$shitte-masu-yo$" by adding "*yo*."

We describe a computational model for qualitatively predicting changes in politeness through the addition of word endings to expressions. This model provides an important base for establishing a method that can be used to synthesize expressions with a desired level of politeness by computer.

In our previous study, numerical models of the strategy for choosing polite expressions were proposed[1]. It is expected that the present and the previous studies will contribute to realize some sort of flexible conversational systems.

2 Model for Changes in the Politeness of Expressions through the Addition of Word Endings

A $politeness - relationship$ in which there is only one speaker and one listener is assumed in our study. Although the assumed situation is a simple one, the model can be extended for use with more complicated situations in which more than two people are involved.

In our study, we assume that there are two types of probability distributions in the psychology of politeness.

2.1 The Psychological Probability Distributions of Politeness

Ogino revealed that the politeness of Japanese expressions with the same speech intention can be evaluated as scalar values in a one-dimensional psychometrical space of politeness [2]. Let us call the politeness of an expression $\mathbf{E}$ measured in a psychometrical space "politeness value $p(\mathbf{E})$." Psychometrical space is related to $politeness - relationship$s because the level of politeness of the expressions most suitable for the given $politeness - relationship$ can be evaluated as values in a one-dimensional psychometrical space of politeness [3]. Therefore, the most suitable expression $\mathbf{E}$ for a specific $politeness - relationship$ can be uniquely determined; however, the expression $\mathbf{E}$ is not solely restricted to one specific $politeness - relationship$. In other words, $\mathbf{E}$ can be used not only one, but for a variety of $politeness - relationship$s. Thus, there must be a probability distribution that corresponds to each $\mathbf{E}$ in a one-dimensional psychometrical space of politeness, where the politeness value $p(\mathbf{E})$ is the representative value (mean) of the distribution. Here, we introduce the following assumption.

Assumption 1: Corresponding to each expression $\mathbf{E}$, a normal distribution of the mean $p(\mathbf{E})$ exists in a one-dimensional psychometrical space of politeness.

Let us denote the normal distribution N (μ_E, σ_E^2), where $\mu_E(=p(\mathbf{E}))$ and σ_E^2 represent the mean and the variance of the distribution N.

Figure 1 shows examples of normal distributions that corresponds to "*shitte−masu*" and "*zonjite−masu.*" Both expressions have the same speech intention: "I know," but "*shitte − masu*" is less polite than "*zonjite − masu.*" The abscissa in Fig. 1 represents the politeness values of the most suitable (or ideal) expressions corresponding to *politeness−relationship*s. In general, variances do not necessarily coincide with these distributions.

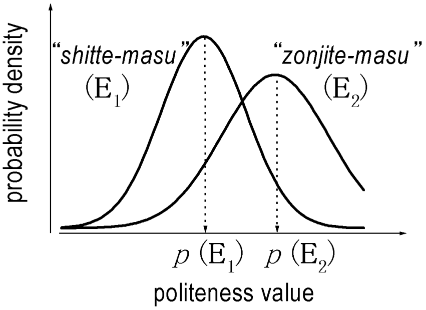


Fig. 1. An example of probability distributions that correspond to two expressions with the same speech intention but different levels of politeness.

2.2 Consistency between Expressions and Word Endings

Now let us suppose a situation in which we add the word ending **e** to an expression **E**. In some cases, we feel slightly uneasy even when the addition of the **e** to the **E** is grammatically legitimate. Such uneasiness can be explained by introducing the concept of "*politeness consistency*" between **E** and **e**. The following assumption is included in the concept.

Assumption 2: Corresponding to each word ending **e**, a normal distribution exists in a one-dimensional psychometrical space of politeness. The distribution represents the ideal distribution of the most suitable (or ideal) expression to which the word ending **e** would be added.

Let us denote the normal distribution N (μ_e, σ_e^2), where μ_e and σ_e^2 represent the mean and the variance of the distribution. The greater the difference between N($p(\mathbf{E})$,σ_E^2) and N(μ_e,σ_e^2), the greater the feeling of uneasiness caused by the addition. The degree of uneasiness can then be evaluated as "*consistency C*," which is defined as the area which N($p(\mathbf{E})$,σ_E^2) and N(μ_e,σ_e^2) overlap each other. Figure 2 shows an example of C, where **E** is "*shitte − masu*" and **e** is "*yo.*"

The consistency C between **e**(which distributes N(μ_e,σ_e^2)) and **E** (which distributes N($p(\mathbf{E})$,σ_E^2)) is defined as

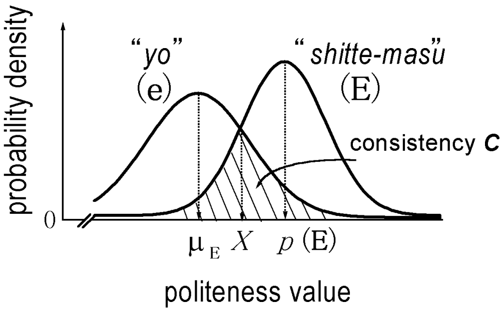


Fig. 2. An example of consistency C.

$$C \triangleq \frac{1}{\sigma_E\sqrt{2\pi}}\int_{-\infty}^{X} e^{-\frac{(x-p(\mathbf{E}))^2}{2\sigma_E^2}}\,dx + \frac{1}{\sigma_e\sqrt{2\pi}}\int_{X}^{\infty} e^{-\frac{(x-\mu_{\mathbf{e}})^2}{2\sigma_{\mathbf{e}}^2}}\,dx, \tag{1}$$

where X represents the horizontal value of the intersection between the probalility distribution functions.

The condition $\mu_e \le p(\mathbf{E})$ is presumed in Eq. (1); however, an equation for the condition in which $\mu_e > p(\mathbf{E})$ can be described in a similar form.

2.3 Model

Now let us consider the changes in politeness values ($\varDelta(\mathbf{E},\mathbf{e})$) through the addition of the word ending **e** to the expression **E**, where $\varDelta(\mathbf{E},\ \mathbf{e}) = p(\mathbf{Ee}) - p(\mathbf{E})$, and **Ee** represents the expression by combining **E** and **e**. From the viewpoint of politeness, the total amount of information included in an expression can be divided into two parts: the amount of information regarding the speech intention and the amount of information regarding the politeness. Therefore, it is reasonable to assume that $\varDelta(\mathbf{E},\ \mathbf{e})$ is proportion to the amount of information (I) obtained by the addition of **e** because it mainly affects the level of politeness of **E**, but hardly affects the speech intention of **E**. This relationship between $\varDelta(\mathbf{E},\ \mathbf{e})$ and I can be described as

$$\varDelta(\mathbf{E},\mathbf{e}) = \mathrm{k}_1 \cdot I + \mathrm{k}_2, \tag{2}$$

where coefficients k_1 and k_2 are constants that are dependent on psychometrical space.

Then, I (the amount of information obtained by the addition of **e** to **E**) can be evaluated by $\ln(1/C)$, because C represents the occurrence probability of **e** after we have established the probalility distribution $\mathrm{N}(p(\mathbf{E}),\sigma_E^2)$ that correspond to **E** [4]. Therefore, substituting $\ln(1/C)$ for I in Eq. (2) yields

$$\varDelta(\mathbf{E},\mathbf{e}) = \mathrm{k}_1 \cdot \ln(1/C) + \mathrm{k}_2. \tag{3}$$

2.4 Prediction of Changes in Politeness through the Addition of Word Endings

Now let us assume that the variances of the probability distributions N($p(\mathbf{E})$,σ_E^2) and N(μ_e,σ_e^2) coincide, that is to say, $\sigma_E^2 = \sigma_e^2 (= \sigma^2)$ in Eq. (1). Then, the Eq. (1) can be rewritten as

$$C(X) = \frac{2}{\sigma\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{(x-p(\mathrm{E}))^2}{2\sigma^2}} dx. \tag{4}$$

The standardization $t = (x\text{-}p(\mathbf{E}))/\sigma$ yields

$$C(X') = \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{X'} e^{-\frac{t^2}{2}} dt, \tag{5}$$

where $X' = (\mu_e\text{-}p(\mathbf{E}))/2\sigma$.

The function $C(X')$ cannot be analytically solved because it is a type of error function; however, C can be approximated by $e^{-f(X')}$, $f(X') = \mathrm{k}_3\ X'{+}\mathrm{k}_4$ when $|X'|{\le}2.5$, where k_3 and k_4 are constant. The restriction $|X'|{\le}2.5$ is equivalent to the restriction: "$C(X')$ occupies more than 1.2% of the area of $N(p(\mathbf{E}),\sigma^2)$." Therefore, this condition is fulfilled in the most combinatorial variation of **E** and **e**. Substituting $e^{-f(X')}$ for C in Eq. (3) yields

$$\Delta(\mathbf{E}, \mathbf{e}) = \mathrm{K}_1(\mu_e - p(\mathbf{E})) + \mathrm{K}_2, \tag{6}$$

where $\mathrm{K}_1 = \mathrm{k}_1\mathrm{k}_3/2\sigma$, $\mathrm{K}_2 = \mathrm{k}_2{+}\mathrm{k}_1\mathrm{k}_4$.

Equation (6) shows that Δ(**E, e**) (changes in politeness through the addition of **e** to **E**) is a linear of $(\mu_e - p(\mathbf{E}))$ [**predicted feature**].

3 Experiments

Psychological experiments were performed to verify the predicted feature described above. Ninety-seven subjects participated in the experiments.

3.1 Stimuli

Two types of expression groups corresponding to different speech intentions were utilized in the experiments. In addition, two types of word endings corresponding to each expression group were also used.

3.2 Expression Groups

Expression Group 1. Twenty-one expressions corresponding to the speech intention: "I know" were used as the original expressions (Table 1). The word ending was fixed to "*yo*," which is an impolite word ending. Therefore, Japanese expressions tend to reduce in politeness with the addition of "*yo*". Finally, forty-two (twenty-one expressions without "*yo*" and twenty-one expressions with "*yo*") expressions were used as stimuli for the experiments.

Table 1. Expression group 1

1: $waka - ru$	8: $shitte - ru$	15: $zonjite - ori - masu$
2: $wakari - masu$	9: $shitte - i - ru$	16: $zonji - agete - masu$
3: $wakkatte - ru$	10: $shitte - masu$	17: $zonji - agete - orimasu$
4: $wakatte - i - ru$	11: $shitte - i - masu$	18: $shouti - shite - ru$
5: $wakatte - masu$	12: $shitte - ori - masu$	19: $shouti - shite - i - masu$
6: $wakatte - i - masu$	13: $zonjite - masu$	20: $shouti - shite - i - masu$
7: $wakkatte - ori - masu$	14: $zonjite - i - masu$	21: $shouti - shite - ori - masu$

Table 2. Expression group 2

1: $iu?$	8: $oose - ni - naru?$	14: $noberu?$
2: $iwa - reru?$	9: $oose - ni - narareru?$	15: $nobe - rareru?$
3: $hanasu?$	10: $shaberu?$	16: $onobe - ni - naru?$
4: $hana - sareru?$	11: $shabe - rareru?$	17: $onobe - ni - narareru?$
5: $ohanashi - ni - naru?$	12: $oshaberi - ni - naru?$	18: $ossharu?$
6: $ohanashi - ni - narareru?$	13: $oshaberi - ni - narareru?$	19: $ossha - rareru?$
7: $ohanashi - suru?$		

Expression Group 2. Nineteen expressions corresponding to the speech intention: "Will you speak? (at the meeting)" were used as the original expressions (Table 2). The word ending was fixed to"*masu*." All Japanese expressions is expected to increase in politeness with the addition of "masu." Finally, thirty-eight (nineteen expressions without "*masu*" and nineteen expressions with "*masu*") expressions were used as stimuli for the experiments.

3.3 Procedure

Experiments for each expression group were performed using the paired-comparison method.

Paired Comparison Method.

Step 1: A pair of expressions from the expression group is presented to the subjects.
Step 2: Each subject is required to indicate which expression in the pair sounds more polite.
Step 3: Repeat Steps 1 and 2 until all pairs have been examined.

Thurstone's scaling [5] was applied to the experimental data in order to obtain the politeness values.

4 Results

Figures 3 and 4 show the experimental results for expression groups 1 and 2, respectively. The numbers next to the dots in the figures represent the identification numbers for expressions in Tables 1 and 2. The abscissa represents the politeness values: $p(\mathbf{E}_i)$ of the original expressions $\mathbf{E}_i$, while the ordinate represents the changes in politeness values: $\varDelta(\mathbf{E}_i,\ \mathbf{e})$ $(= p(\mathbf{E}_i\ \mathbf{e}) - p(\mathbf{E}_i)$, $i = 1,..,n$, where n is the number of original expressions). The arrow used in Fig. 4 will be explained in Section 4.2.

Simple regression analysis was applied to the data sets $(x_i, y_i) = (p(\mathbf{E}_i),$ $\varDelta(\mathbf{E}_i,\ \mathbf{e}))$, $i = 1,..,n$ in each figure. The regression parameters a and b of the regression line $y = ax{+}b$ can be estimated [6] as

$$a \triangleq \frac{\sum\limits_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum\limits_{i=1}^{n}(x_i - \bar{x})^2}, \tag{7}$$

$$b \triangleq \bar{y} - a\bar{x}, \tag{8}$$

where $\bar{x}$ and $\bar{y}$ represent the respective means of x_i and y_i.

Moreover, the coefficient of determination R^2 (representing the degree of fitness of data to the regression line) can be estimated [7] as

$$R^2 \triangleq \frac{\sum\limits_{i=1}^{n}(\bar{y} - (ax_i + b))^2}{\sum\limits_{i=1}^{n}(y_i - \bar{y})^2} \tag{9}$$

4.1 Regression Analysis Results for Expression Group 1

Simple regression analysis was applied to all data sets (dots) shown in Fig. 3. The estimated parameters a and b of the regression line $y = ax{+}b$ are $a = -0.28$, b = 0.71, and $R^2 = 0.84$.

4.2 Regression Analysis Results for Expression Group 2

Figure 4 seems to be divided around the expression $\mathbf{E}_{18}$, as indicated by the left-pointing arrow. Let us refer to the cluster to the left of $\mathbf{E}_{18}$ as “cluster-L,” and the cluster to the right of $\mathbf{E}_{18}$ as “cluster-R.” Cluster-L has a tendency to increase monotonically, while cluster-R has a tendency to decrease monotonically.

Simple regression analysis was separately applied to cluster-L and cluster-R. The estimated parameters are $a = 0.58$, b = 1.6, and $R^2 = 0.93$ for cluster-L, and $a = -0.54$, b = 8.47, and $R^2 = 0.75$ for cluster-R.

5 Discussion

5.1 Explanation of the Results for Expression Group 1

The experimental results described in Section 4.1 (for expression group 1) suggest that the dots in Fig. 3 fit well to the regression line ($R^2 = 0.84$). This result

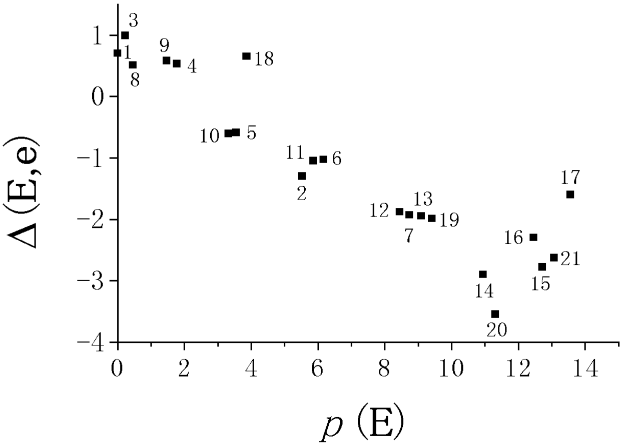


Fig. 3. Experimental results for expression group 1 (word ending: "*yo*")

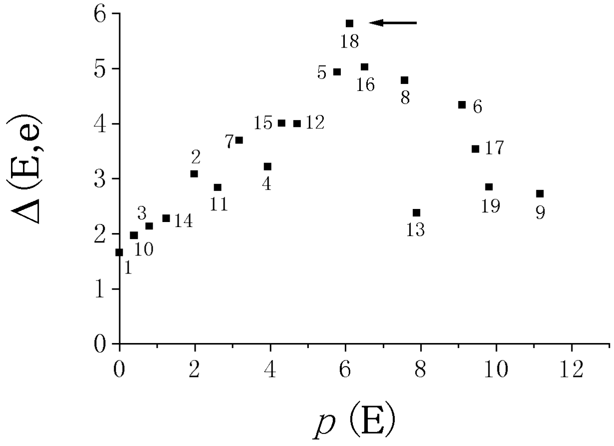


Fig. 4. Experimental results for expression group 2 (word ending: "*masu*")

supports the predicted feature: "$\varDelta(\mathbf{E}, \mathbf{e})$ is a linear of $(\mu_e - p(\mathbf{E}))$," which was described in Section 2.4.

These results also coincide with the following linguistic intuition: The more polite the original expressions, the more the politeness of the original expressions is reduced with the addition of "*yo*".

5.2 Explanation of the Results for Expression Group 2

The experimental results described in Section 5.2 (for expression group 2) suggest that the dots in cluster-L shown in Fig. 4, and those in cluster-R in Fig. 4 both fit well individually to the regression lines ($R^2 = 0.93$ for cluster-L, 0.75 for cluster-R). Therefore, the predicted feature described above is also supported by cluster-L and cluster-R.

The absolute values of the inclinations (a) of the regression line for cluster-L (0.58) and the regression line for cluster-R (−0.54) almost coincide; however, cluster-L has a positive inclination and cluster-R has a negative inclination. The reason the sign of the regression line changes near the expression $\mathbf{E}_{18}$ can be explained by assuming the following features.

1. $\mathrm{K}_1 > 0$ when $\mu_e < p(\mathbf{E})$,
2. $\mathrm{K}_1 < 0$ when $\mu_e > p(\mathbf{E})$,
3. $\mu_e \simeq p(\mathbf{E}_{18})$,

where K_1 represents a coefficient in Eq. (6). The first and the second assumption represent the general features of K_1.

The third assumption, a specific feature for expression group 2, represents that the expression $\mathbf{E}_{18}$ is assumed to be the most suitable expression to which the word ending "*masu*" would be added. Then, $\mathrm{K}_1 < 0$ for cluster-L because $\mu_e > p(\mathbf{E})$ for all $\mathbf{E}$ in cluster-L. Therefore, $\Delta(\mathbf{E}, \mathbf{e})$ in cluster-L must be a line with positive inclination, while $\Delta(\mathbf{E}, \mathbf{e})$ in cluster-R must be a line with negative inclination. This is because $\mu_e < p(\mathbf{E})$ for all $\mathbf{E}$ in cluster-L.

The first and the second assumptions described above are valid for expression group 1 (word ending "*yo*"). Here, $\mathrm{K}_1 > 0$ is assumed to be true for all expressions $\mathbf{E}$ because $\mu_e < p(\mathbf{E})$ is assumed to be true for word ending $\mathbf{e}$ "*yo*" which is impolite.

6 Conclusion

We have described a computational model to qualitatively predict changes in politeness when word endings are added to expressions. Experimental results support the validity of the model. This model provides a theoretical base for the generation of complex expressions by combining simple expressions. Therefore, this model is an important base for establishing a method to synthesize expressions with a desired level of politeness by computer.

References

1. Shirado T., and Isahara H.: Numerical models of the strategy for choosing polite expressions, LNCS 2004, pp. 98-109, Springer Berlin(2001).
2. Ogino T: Social linguistic research on polite expressions, Nihongo-gaku Vol. 5(1986). (in Japanese).
3. Brown P., and Levinson S.: Politeness - Some universals of language usage -, Cambridge London(1987).
4. Abramson, N.: Information theory and coding, McGraw-Hill New York(1963).
5. Baird J.C. and Noma E.: Fundamentals of scaling and psychophysics, Wiley New York(1978).
6. Montgomery, D.C. et al.: Introduction to linear regression analysis, Wiley New York(1992).
7. Siegel, A.F. et al.: Statistics and data analysis, Wiley New York(1996).

Tartar Morphology Implementation

Dj.Sh. Suleymanov

Kazan State University, Tatarstan Republic, Russia
dvdt@telecet.ru

Abstract. A model for representation of different strata of information necessary for morphological analysis of the agglutinative Tatar language is presented.

1 Introduction

Tartar language is spoken in Tatarstan, one of the largest states of Russian Federation, and is, along with Russian, one of the two official languages of this state.

It is a typical agglutinative language similar to Turkish. An important characteristic of the morphology of such languages is that the word structure is very regular, practically without exceptions. A word can have many (and a variable number of) grammatical morphemes, not a small fixed number as in inflective languages. This leads to a potentially infinite number of wordforms for each lexeme. For such languages, morphological analyzer plays a crucial role in text analysis, comparable with that of syntactic analyzer. In particular, simple word bag models of morphological analysis cannot be used for such languages. Other well-known models developed mainly for inflective languages [1–4] also are not suitable for agglutinative morphology. Thus the necessity for further development of morphological models of Tatar and other agglutinative languages.

We have developed a declarative model of Tartar morphology and implemented the corresponding software that allows analyzing and generating Tartar wordforms. It is used in speech synthesis, in publishing systems, and in text processing systems, as well as in computer-aided language teaching.

2 The Model

Each morpheme in the model is supplied with the information on various language levels. There are 7 top-level features describing each morpheme.

Morphological feature has the following sub-features.

Functional sub-feature represents the grammatical properties of the morpheme. It includes the following sub-features. Binary *Morpho-functional* sub-feature indicates whether the morpheme is synthetic or analytical. For example, *-daj* ‘similar to’ is a synthetic morpheme (e.g., *uramdaj* ‘similar to a street’) while *syman* ‘similar to’ is an analytic one (*uram syman* ‘similar to a street’). Ternary *Syntactic-Functional* sub-

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 327–329, 2002.

feature indicates the category of the morpheme (derivational, inflectional, or modal). For example, *-dan* ‘ablative case’ is an inflectional morpheme since the lexeme having this morpheme preserve its lexical meaning. *Semantic-functional* sub-feature describes the meaning of the morpheme.

Sub-feature *Order* defines the relative order in the wordform of other morphemes with respect to the given one.

Binary sub-feature *Recursion* indicates whether the morpheme can occur multiple times within the same wordform forming regularly built meanings. For example, *-dagy* ‘something that is in’: *avyldagy* ‘something that is in the village’, *avyldagydagy* ‘something that is in something that is in the village’, *avyldagydagydagy* ‘something that is in something that is in something that is in the village’, etc.

Syntactic feature specifies the syntactic properties of the affixal morpheme. It includes the following sub-features: *Word Order*, describing how this morpheme affects word order; *Idioms*; *Morphological Ellipsis.*

Semantic feature specifies the semantic properties of the affixal morpheme. It includes the following sub-features: binary sub-features *Preserves Lexical Semantics* and *Meaning*. The latter sub-features depends on both the specific morpheme and on the context.

To describe the meanings of the morphemes in a uniform manner, we had to develop a logical language based on predicate calculus to define the relationships between the semantic properties of the morpheme and the context. We have defined 77 standard contexts defined by 37 predicative constructions. For example, in this system the morpheme *-ga* ‘directon’ is described as follows:

$$\begin{bmatrix} \mathit{object}_1\mathit{:} & \begin{bmatrix} \text{action - local:} & \begin{bmatrix} \text{old - location :} & \mathit{object}_2 \\ \text{name - action :} & \mathit{name} \\ \text{route :} & \mathit{object}_4 \\ \text{global - location :} & \mathit{location} \\ \text{new - location :} & \mathit{object}_3 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

which is instantiated in the context *urmanga baru* ‘go to forest’ as

$$\begin{bmatrix} \mathit{object}_1\mathit{:} & \begin{bmatrix} \text{action - local:} & \begin{bmatrix} \text{old - location :} & \mathit{object}_2 \\ \text{name - action :} & \mathit{baru}\text{ ’go’} \\ \text{route :} & \mathit{object}_4 \\ \text{global - location :} & \mathit{location} \\ \text{new - location :} & \mathit{urman}\text{ ’forest’} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

which corresponds to the following role hierarchy: *relation of action*: *action of the change of the spatial location*: *new location.*

Morphonological feature specifies the phonetic characteristics of the morpheme, possibly depending on the context. Its sub-feature *Allomorph Table* describes all allomorphs of the morpheme, i.e., its phonetic representations defined by phonological rules.

Synonymy, **Homonymy**, and **Antonymy** are the other three features, which we do not discuss here.

3 Conclusions

Detailed specification of the meaning and syntactics of Tatar morpheme has required the development of a complex descriptive system based on predicate calculus, comparable in its functioning and complexity with syntactic descriptions. This system has been used for the implementation of a general-purpose morphological analyzer for Tatar.

Acknowledgements

The work was done under partial support of the Academy of Sciences of Tatarstan Republic, Russia. The author thanks Prof. A. Gelbukh and Prof. G. Sidorov for useful discussion.

References

1. Gelbukh, Alexander (1992) *Effective implementation of morphology model for an inflectional natural language* (translation from Russian). J. Automatic Documentation and Mathematical Linguistics, Allerton Press, vol. 26, N 1, 1992, pp. 22 - 31.
2. Gelbukh, Alexander (2002) A data structure for exact and approximate prefix search in very large dictionaries under access locality requirements and its application to morphological analysis and spelling correction. J. *Computación y Sistemas*, 2002, to appear.
3. Koskinniemi, Kimmo (1983) Two-level morphology: A general computational model for word-recognition and production. University of Helsinki, 160 pp.
4. Sidorov, Grigori (1996) *Lemmatization in automatized system for compilation of personal style dictionaries of literature writers*. In: *Word of Dostoyevsky*. Moscow, Russia, Russian Academy of Sciences, 1996. pp. 266-300.

Automatic Generation of Pronunciation Lexicons for Spanish

Esmeralda Uraga and Luis Pineda

Universidad Nacional Autónoma de México
Instituto de Investigación en Matemáticas Aplicadas y en Sistemas
Mexico, D.F.
{euraga,luis}@leibniz.iimas.unam.mx

Abstract. In this paper a method for the automatic generation of pronunciation lexicons including multiple word pronunciation is presented. This method is based on the application of a set of rules for grapheme-to-phone conversion for Mexican Spanish. The generation of multiple word pronunciations is based on the introduction of pronunciation variants by applying allophonic rules. The performance of a speech recognition system using a pronunciation lexicon with and without multiple word pronunciations is also presented.

1 Introduction

Words are almost always pronounced differently. This variation in pronunciation is a major problem in Automatic Speech Recognition (ASR) [9]. The production of pronunciation lexicon for large vocabularies in which pronunciation variants are considered is a very costly and time consuming process. To facilitate this task, phonological rules to introduce allophonic pronunciation variants for a selected number of phonemes out of canonical pronunciation models have been used [4]; however, in such approach it is still required to produce canonical pronunciation models manually or use a pronunciation dictionary. In this paper a method for the automatic generation of multiple word pronunciations for Mexican Spanish is presented. This approach is based on the definition and application of a set of rules for grapheme-to-phone conversion. A phonetic alphabet called Mexbet, which is based on Worldbet symbols [5], including representations for all Spanish phones is also proposed. The rules cover most phonetic types of Spanish contexts and map orthographic representations to sequences of Mexbet symbols. The performance improvement of a continuous speech recognition system in the phone call domain using a multiple pronunciation lexicon is also presented.

2 Grapheme-Phoneme Relations

According to the Spanish Royal Academy [8], there are 29 letters to represent the 25 phonemes that form the Spanish phonological system: 20 consonant phonemes

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 330–338, 2002.

(p, b, t, d, k, g, f, T[1], s, j, ch, r, rr, l, ll, m, n, ñ, y, w) and 5 vowel phonemes (a, e, i, o, u).

The relation between graphemes and phonemes in Spanish is not one-to-one, as almost all graphemes can be pronounced in more than one way depending on contextual factors, the syllabic boundaries in the words and the geographical area. In addition, different phonemes can be represented by the same letter. Following [3,7,8,11] we have found nine kinds of relations in which one or more graphemes have zero, one or more than one phoneme realization, and viceversa. These are some examples:

1. A grapheme represents a phoneme (eg. grapheme *t* represents only phoneme */t/*).
2. The same grapheme can represent different phonemes in different contexts. (eg. *x*[2] can represent */j/, /s/* or */ch/* as they occur in the words *México, excepción* and *mixiotle*).
3. A grapheme can represent a sequence of two different phonemes (eg. *x* in the word *excelente* is pronounced as the sequence */k/ /s/*).
4. A sequence of two graphemes can represent a single phoneme (eg. *gu* before *e* or *i* represents the phoneme */g/* as it occurs in the word *águila*).
5. Different graphemes can represent the same phoneme. For example, in the words *zumo, sala, excepción* and *cero,* all the graphemes *z, s, x*, and *c* (before *e* or *i*) are pronounced as */s/.*
6. Some graphemes can represent no phoneme. A mute *h* is omitted in the pronunciation if the letter *c* do not occur before *h,* as it happens in the words *hola* or *zanahoria.*
7. The same grapheme, in the same context, can represent by different phonemes in different words depending on the syllable and morpheme boundaries. The word *sobra*, for example, is formed by syllables *so* + *bra* and *r* is pronounced with */r/*, while the word *subraya* is formed by syllables *sub* + *ra* + *ya* (*b* and *r* in different syllables) and *r* is pronounced with */rr/*.
8. The presence of some special graphemes like dieresis (¨) and accent (´) determines whether a grapheme is pronounced in one way or another. Dieresis in vowel *u* (*ü*) is used to indicate the pronunciation of *u* between *g* and *e* or *i* (eg. antigüedad, pingüino); the accent in vowels (*á, é, í, ó, ú*) determines the syllabic boundaries of words and its stressed syllable. The word *rio* for instance, is pronounced as a monosyllabic word, while the word *río* is pronounced with two syllables *rí-o* where the first syllable is stressed. Similar effects happens in the words *cambie, bofeteo* and *licuo* because their stressed syllable is different to the pronunciation of *cambié, bofeteó* and *licúo*

[1] The phoneme */T/* is used only in Castillian Spanish to pronounce the grapheme *c* before *e* (i.e., cero */T e r o/*) or *i* (i.e., cielo */T i e l o/*), and the grapheme *z* (i.e., zeta */T e t a/*). In American Spanish, it is used */s/* instead of */T/*.

[2] Many words that have the grapheme *x* in Mexican Spanish were adopted from Mexican languages like Nahuatl.

9. The same word is pronounced in different ways by different people. For example: *área* is pronounced with two syllables (*á-rea*) or with three syllables (*á-re-a*). The word *cae* is pronounced with one syllable (*cai*) or by two syllables (*ca-e*).

3 The Phonetic Alphabet Mexbet

To represent all Spanish sounds an adequate scheme is required. Current phonetic alphabets for Spanish consider information regarding the pronunciation of common phones only, despite that dialects differ in phonology, phonetics and vocabulary. Here, we propose a phonetic alphabet based on Worldbet [5] in which symbols for all Spanish phonemes are included. We refer to this phonetic alphabet as **Mexbet** (Table 1). Mexbet includes symbols for allophones in complementary distribution and free variation. Additionally, due to the contextual effect produced between vowels forming a diphthong, a representation for each of these diphthongs is formed by concatenating the corresponding vowels symbols. Finally, Mexbet also considers symbols for some kinds of cross-word variations, affected pronunciations and regional dependent sounds.

Table 1. Symbols of the phonetic alphabet Mexbet

Phoneme	Sym	Word	Phoneme	Sym	Word	Phoneme	Sym	Word
/p/	**pc**	paso	/rr/	**r**	honra	/l/	**l**	habla
	p	paso		**9r**	Israel		**l_T**	alzar
/b/	**bc**	vaso	/s/	**s**	sol		**l[**	alto
	b	vaso		**s[**	hasta		**L**	malhiere
	V	iba		**s_T**	ascender	/ll/	**L**	silla
	V_0	obtener		**s_h**	dos	/y/	**y**	suyo
	v	virtud		**z**	transbordo		**dZc**	yema
/t/	**tc**	tasa		**z[**	desde		**dZ**	yema
	t	tasa	/x/	**x**	jamás	/w/	**w(**	huésped
	t_T	azteca		**X**	ojo	/i/	**i**	imán
/d/	**dc**	dos		**h**	Ajá!		**I**	río
	d	dos	/ch/	**tSc**	chal		**j**	rio
	D	hada		**tS**	chal		**i(**	ley
	D_0	adjetivo		**S**	leche	/e/	**e**	ella
/k/	**kc**	cal	/m/	**m**	mesa		**E**	piel
	k	cal		**M**	inferior	/a/	**a**	casi
/g/	**gc**	gana	/n/	**n**	nube		**A**	pausa
	g	gana		**n_T**	encima	/u/	**u**	uno
	G	viga		**n[**	canto		**U**	rudo
	G_0	zigzag		**N**	mango		**w**	puente
/f/	**f**	foco		**m**	inmóvil		**u(**	deuda
/T/	**T**	zumo		**ñ**	conlleva	/o/	**o**	oro
/r/	**r(**	pero	/ñ/	**ñ**	año		**O**	soy

4 Automatic Generation of Canonical Pronunciations

Canonical pronunciations are represented by sequences of phonetic symbols. Since the pronunciation of Mexican Spanish words is rather regular, it was possible to define a set of rules for grapheme-to-phone conversion. The rules were defined manually according to linguistic knowledge and observations found in a hand-labeled speech corpus [11]. This kind of rules have also been used for text-to-speech synthesis in Spanish as is shown in [1,2].

4.1 Rule Formalism

Following [3] we define a **class** as a sequence of graphemes or symbols having a common property. Classes are used to generalize the rules. The class of vowels, for instance, is represented through the symbol V as V : a, e, i, o, u, á, é, í, ó, ú, ü.

A rewriting **rule** is defined as $< ls > \rightarrow < rs > / < lc > \_ < rc >$

where $< ls >$ (left string) is the string to be replaced, $< rs >$ (right string) is the string replacing ls, $< lc >$ (left context) represents the string to be found on the left side of ls, $< rc >$ (right context) represents the string to be found on the right side of ls.

Next, we present some instances of rules:

- A context-free rule: $p \rightarrow [p]$
- A context-dependent rule: $n \rightarrow [N]$ / V _ +$[k]$,+$[g]$,+$[x]$

where the string in square brackets is a phonetic sequence, + represents syllabic boundaries and different options are separated with ",".

For example, c is pronounced $[s]$ if it is followed by e or i as in *cero* and *cien.* The rule to replace c by $[s]$ in this context is as follows:

$c \rightarrow [s]$ / _ e, i

4.2 Grapheme-to-Phone Conversion

For the conversion process, rules were ordered relative to the context and length of graphemes: context-dependent rules were listed before the context-free ones, and rules were also ordered in relation to the largest input grapheme length (e.g., ch is processed before c). Rule validation was performed by comparing automatic and manual phonetic transcriptions.

The input of the grapheme-to-phone conversion is a list of words (eg. the vocabulary of the speech recognition system) and the output is a pronunciation lexicon including orthographical, syllabic, phonological and phonetic information. The grapheme-to-phone conversion is performed by a process of substitution of exception words followed by the insertion of word and syllabic boundaries and, finally, the creation of canonical pronunciations. Next, we illustrate each of these processes.

4.3 Substitution of Exception Words

There are words that have an exceptional form of pronunciation which we refer to as **exception words**. This is often the case for words adopted from foreign or indigenous languages. Words of this kind are substituted by their normalized form stored in a dictionary table which is looked up before phonological rules are applied. The word *México*, for instance, should be canonically pronounced $[m][e][k][s][i][k][o]$, but in México it is pronounced $[m][e][x][i][k][o]$. So, the word *México* is replaced by the string *Méjico,* that produces the desired pronunciation. The corresponding rule is as follows:

$México \rightarrow Méjico$

4.4 Insertion of Word and Syllabic Boundaries

The pronunciation of phonemes depends often on syllable boundaries. Consider, for instance, the pronunciation of */r/* in *sobra* and *subraya* (see Section 2). Accordingly, before applying a phonological rule, words and syllables must be individuated. To mark **word and syllabic boundaries** the symbols # and + are respectively inserted.

Considering the classes of vowels and consonants there are three kinds of syllabic boundaries in Spanish words; these are as follows:

- Syllabic boundary V+C (e.g., *ca+sa* (*house*))
- Syllabic boundary C+C (e.g., *al+to* (*tall*))
- Syllabic boundary V+V (e.g., *tra+er* (*to bring*))

The rules to mark word and syllabic boundaries were manually designed from linguistic information on syllabic boundaries reported in [8]. In Spanish there are many conditions to demarcate syllabic boundaries that were expressed in 22 syllabic rules. One common and simple instance of these V+C conditions is that any sequence of the form VCV is split off as V+CV. This condition is expressed by the **syllabification rule**:

$C \rightarrow +C\ /\ V\ \_\ V$

For instance, the string *#raya#* is separated as *#ra+ya#*.

One condition for the kind C+C, for instance, is if the sequence is VCCV and the first consonant is one of {*p, b, f, t, d, k, g*} and the second consonant is *r* (eg. libro, subrigadier, prensa) then the sequence is split off as V+CCV (eg. *li+bro, su+bri+ga+dier*) else it is split off as VC+CV (eg. *pren+sa*). The generic rules to formalize this knowledge are as follows:

$X : \{p,b,f,t,d,k,g\}$
$R : \{r\}$
$X \rightarrow$ +X /V _ RV
$C \rightarrow$ C+ /V _ RV

These rules have wide applicability but there are exceptions; consider for instance the word *subraya* would be split off as *su+bra+ya* instead of its correct form which is *sub+ra+ya.* To handle the exceptions specific rules are defined.

These rules are ordered before the generic ones. For this particular case, the next syllabification rules were defined:

$b \rightarrow +b\ /\#su\ \_\ rigadier, ranquial$
$b \rightarrow b+\ /\#su\ \_\ rV$
$C \rightarrow C+\ /\#sub\ \_\ V$

Where the second rule splits off the string *#subra+ya#* as *#sub+ra+ya#*.

4.5 Canonical Pronunciation Models

We turn now to the construction of canonical pronunciations. These consist of sequences of Mexbet symbols representing the standard pronunciation of words, and they are produced through the application of **phonological rules** to the strings resulting from syllabification.

Canonical pronunciation models are obtained by a grapheme-to-phone conversion process, wich is performed through phonological rules. The creation of these rules was based on linguistic knowledge extracted from many kinds of information sources as phonetic alphabets [5], text-to-speech systems [1] and literature about Spanish, phonology and phonetics [8].

One or more phonological rules were defined to cover each of the 9 grapheme-phoneme relations discussed previously in Section 2. For the simplest case (case 1), a set of context-independent rules have been designed. The plosives /*p*/, /*t*/ and /*k*/, for instance, are always pronounced in the same way in Mexican Spanish and their corresponding phones are produced by the following rules:

$p \rightarrow [p]$
$t \rightarrow [t]$
$k \rightarrow [k]$

The situation is more complex for the other kinds of grapheme-phoneme relations. Next we consider the case of a grapheme that can be realized with more than one pronunciation (i.e., Case 2 in Section 2). The grapheme *g*, for instance, can represent the phone [*x*] (whenever its right context is *e* or *i* as in *gel* and *gis*), *g* can also represent the phones [*g*] (as in *mango*) and [*G*] (as in *agua*). The corresponding phonological rules are:

$g \rightarrow [x]\ /\ \_\ e, i$
$g \rightarrow [g]\ /\ \#, [N]\ \_$
$g \rightarrow [g]\ /\ \_\ r,\ l$
$g \rightarrow [G]$

The set of phonological rules for Mexican Spanish consists of about 300 rules. These rules consider about 600 phonetic combinations corresponding to the 9 cases mentioned in Section 2.

5 Introduction of Multiple Pronunciation Variants

Allophonic variation for all phonemes is specified through the definition of a set of **allophonic rules**. Multiple pronunciation variants are introduced through the application of allophonic rules to canonical pronunciation models. A general set of 16 allophonic rules were defined to introduce 25 kinds of inter-word pronunciation variations and 10 allophonic rules for introducing cross-word pronunciation variations were also defined. The information on pronunciation variation is not limited to a specific corpus and can easily be applied to other lexicons.

Adding pronunciation variants to the lexicon usually introduces new speech recognition errors due to the acoustic confusability within the lexicon increases. This problem has been minimized by making an appropiate selection of the pronunciation variants [9]. In this work, a domain-dependent and partially data-driven approach was used. Only the most frequent inter-word pronunciation variants in the phone-call corpus [11] were added. Additionally, information of a domain-dependent bigram language model was used to select the cross-word pronunciation variations to be added.

Consider, for instance, the canonical pronunciation of the words in the name *David Rosas*:

$\#[d][a]+[V][i][D]\#$
$\#[r][O]+[s][a][s]\#$

Some of the phonetic units in this model have allophonic variations due to the different ways these can be articulated. Each of these inter-word variations can be captured through an allophonic rule. For the phonetic unit $[d]$ the next rule can be applied:

V : a, e, i, o, u, á, é, í, ó, ú
$\#[d] \rightarrow \#([dc]\ [d]|[D])\ /\ \_\,V$

For instance, the application of this rule to the sequence $\#[d][a]+[V][i][D]\#$ results in the sequence $\#([dc][d]|[D])[a]+[V][i][D]\#$.

To introduce the pronunciation variations of *Rosas*, we consider cross-word coarticulation information of the bigram *David Rosas.* In this case, the allophonic rule $\#[r] \rightarrow \#([r] \mid [9r])$ changes the canonical pronunciation model of *Rosas* into $\#([r]|[9r])[O]+[s][a][s]\#$.

The final pair of rules deletes the boundary symbols # and +:

$\# \rightarrow \phi$
$+ \rightarrow \phi$

6 Experiments and Results

A read continuous speech corpus formed by 900 phrases to request a phone call with a specific person was used. The corpus was recorded at 8000 Hertz via microphone by 14 people. This corpus was hand-labeled at ortographic and phonetic level. Three-fifths of the available data were chosen for training, and the rest was used for testing [11].

Two feed-forward neural networks were trained to estimate the probability of 41 context-independent (CI) acoustic-phonetic categories (24 phonemes, 16 most frequent and domain-dependent allophonic units plus 1 silence unit) and more than 270 context-dependent (CD) acoustic-phonetic categories. The acoustic features were 12 Mel-Frequency Cepstrum Coefficientes (MFCC) + 12 deltaMFCC

+ 1 energy(MFCC) + 1 energy(deltaMFCC) computed at non-overlapping 10-msec frames. At each frame, a 130 dimensional vector is constructed using five surrounding frames. These acoustic models were trained and tested with the CSLU Toolkit [6], [10].

A finite-state grammar was used to model the language in the phone-call domain. The multiple pronunciation lexicon and the acoustic models based on neural networks were evaluated through a continuous-speech recognition system. The size of the vocabulary was 107 words with 230 pronunciations.

The criteria to evaluate the performance of the speech recognition system using the single-pronunciation lexicon and the multiple-pronunciation lexicon is based on reducing the Word Error Rate (WER). The results obtained are summarized in Table 2:

Table 2. Word Error Rates obtained in the evaluation of a speech recognition system with and without pronunciation variations in the lexicon.

Speech Recognition System	**CI**	**CD**
Without pronunciation variations	11.34%	11.17%
With pronunciation variations	4.91%	2.88%

The results show the performance had a significant improvement of 6.43% (CI) and 8.29% (CD) using the lexicon with multiple pronunciation variants.

7 Conclusions

This paper describes a definition, application and validation of a set of rules for translating graphemes into phones. A phonetic alphabet based on Worldbet symbols for representing Spanish, which is called Mexbet, is proposed. A method to create multiple-pronunciation lexicon consisting of substitution of exception words, insertion of syllabic and word boundaries, creation of canonical word pronunciations using a set of rules and a strategy to introduce multiple pronunciation variants were described. The pronunciation lexicons were evaluated through a simple continuous speech recognition system with very promising results, wich show clearly that the propperly selection and insertion of pronunciation variations in a pronunciation lexicon permits to improve the performance of a speech recognition system.

Acknowledgment

The authors gratefully acknowledge the useful discussions and feedback from the following people: Antonio Massé, Miguel Salas, Luis Villaseñor and Erick Schwarz of our research group, Enrique Sucar from ITESM- Campus Morelos, Eduard Hovy, James Allen and the Natural Language Group at the University of Rochester. This research was funded by grant 499316-5-C092A from the National Council of Science and Technology of Mexico (CONACYT).

References

1. Barbosa, A. : Desarrollo de una nueva voz en Español de México para el Sistema de Texto a Voz Festival, Master Thesis, Universidad de las Américas - Puebla, Departamento de Ingenieria en Sistemas Computacionales, (1997).
2. Bonafonte, A., Esquerra, I., Febrer, A., Fonollosa, J. A. R., Vallverdú, F. : The UPC Text-to-Speech System for Spanish and Catalan. Proceedings of the 5th International Conference on Spoken Language Processing, ICSLP'98, Sydney, Australia, (1998).
3. Divay, M.: Phonological Rules for Speech Synthesis. Human Factors and Voice Interactive Systems Daryle. Gardner-Bonneau (ed.). Kluwer Academic Publishers (1999) 99–121.
4. Ferreiros: Introducing Multiple Pronunciations In Spanish Speech Recognition. Proceedings of the Workshop Modeling Pronunciation Variation for Automatic Speech Recognition (1998).
5. Hieronymus, J. L.: ASCII Phonetic Symbols for the Worlds Languages: Worldbet. Technical report, AT&T Bell Laboratories (1993).
6. Hosom, J., Cole, R., Fanty, M., Schalkwyk, J., Yan, Y., Wei, W.: Trainning Neural Networks for Speech Recognition. Center for Spoken Language Understanding (CSLU). Oregon Graduate Institute of Science and Technology (1999) URL: cslu.cse.ogi.edu/tutordemos/.
7. Onieva, J.L.: Cómo Dominar la Gramática Estructural del Español. Editorial Playor, Madrid, Spain (1995).
8. RAE: Real Academia de la Lengua Española. Esbozo de una Nueva Gramática del Español. Espasa-Calpe Editorial, Madrid, Spain (1999) 9–63.
9. Strik, H., Cucchiarini, C., Modeling Pronunciation Variation for ASR: Overview and Comparison Methods. Proceedings of the Workshop Modeling Pronunciation Variation for Automatic Speech Recognition (1998).
10. Sutton, S., Cole, R., De Villiers, J., Schalkwyk, J., Vermeulen, P., Macon, M., Yan, Y., Kaiser, E., Rundle, B., Shobaki, K., Hosom, P., Kain, A., Wouters, J., Massaro, M., and Cohen, M.: Universal Speech Tools: the CSLU Toolkit. Proceedings of the International Conference on Spoken Language Processing (ICSLP). Sydney, Australia, November (1998) 3221–3224.
11. Uraga, E.: Modelado Fonético para un Sistema de Reconocimiento de Voz Continua en Español. Master's thesis. Instituto Tecnológico y de Estudios Superiores de Monterrey - Campus Morelos. México (1999).
12. Yan, Y., Fanty, M., Cole, R.: Speech Recognition Using Neural Networks with Forward-Backward Probability Generated Targets. Center for Spoken Language Understanding (CSLU). Oregon Graduate Institute of Science and Technology. Proceedings of the International Conference of Acoustical Speech Signal Processing, ICASSP-97. Munich (1997).
13. Young, S., Jansen, J., Odell, J., Ollason, D., Woodland, P.: The HTK Book Cambridge University, Entropic Cambridge Research Laboratory (1995) URL:www.entropic.com/HTK/HTKBook/.

Diacritics Restoration:
Learning from Letters
versus Learning from Words

Rada F. Mihalcea

Southern Methodist University,
Computer Science and Engineering Department,
Dallas, TX, 75275-0122
rada@seas.smu.edu

Abstract. This paper presents a method for diacritics restoration based on learning mechanisms that act at letter level. This technique is new to our knowledge, and we compare it with the well known techniques for diacritics restoration that learn from words. Our method is particularly useful for languages that lack large electronic dictionaries and where means for generalization beyond words are required. Accuracies of over 99% at letter level are reported.

1 Introduction

Diacritics restoration is the problem of inserting diacritics into a text where they are missing. With the continuously increasing amount of texts available on the Web, tools for automatic insertion of diacritics become an essential component in many important applications such as Information Retrieval, Machine Translation, Corpora Acquisition, construction of Machine Readable Dictionaries, and others. Spelling correction has a direct impact on the processing quality in many of these applications. For instance, in the absence of a tool for diacritics recovery, a search for the Romanian word *peşte*(*fish*) retrieves *peste*(*over*) as well, *paturi* can be wrongly translated as *beds*, where the intended meaning was *blankets* (the translation of *pături*), and so forth.

The problem as such is not very difficult, and previous work has demonstrated that a good dictionary can lead to over 90% accuracy in accent restoration for French and Spanish [9], [11], [5]. The method described by Michael Simard in [9] is an improvement over a similar method proposed by El-Bèze [4]. It relies on Hidden Markov Models and learns from surrounding words for an overall reported accuracy of 99%. Tufiş and Chiţu [10] propose a similar approach for diacritics insertion in Romanian texts. Yarowsky gives in [11] a comprehensive overview of accent restoration techniques. Most of the algorithms he presents rely on dictionaries and surrounding words in deciding whether to select a form or another for a given ambiguous word. He mentions, in addition to the baseline constituted by the dictionary based approach, N-gram taggers, Bayesian classifiers and decision lists, all of them relying on contexts, and eventually on

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 339–348, 2002.

additional morphological and syntactic information. A different approach is proposed by Nagy et. al in [7], where strings extracted from texts are used to derive statistics, with high precision reported on French texts. Their work is similar with the approach proposed in [1], where trigram similarity measures are employed for automatic spelling correction.

The majority of studies performed so far in this field have addressed well known and widely spread languages such as French or Spanish, and very few studies have emphasized less popular languages like Czech, Slovene, Turkish or other languages that employ diacritics in their spelling. Table 1[1] lists the diacritics encountered in European languages. As seen in the table, a large number of languages face the problem of diacritics restoration. From the entire set of 36 languages listed in the table, English seems to be the most "lucky" one from this point of view, as it is the only one with no diacritics. However, because of this distinction its semantic ambiguity is higher than the average language[2].

Table 1. Diacritics in European languages with Latin based alphabets.

Language	Diacritics	Language	Diacritics
Albanian	ç ë	Italian	à é è í ì ï ó ò ú ù
Basque	ñ ü	Lower Sorbian	ć č ě ł ń ŕ ś š ź ž
Breton	â ê ñ ù ü	Maltese	ċ ġ ℏ ż
Catalan	à ç è é í ï l· ò ó ú ü	Norwegian	å æ ø
Czech	á č d' é í ň ó ř š t' ý ž	Polish	a¸ ć e¸ ł ń ó ś ź ż
Danish	å æ ø	Portuguese	â ã ç ê ó ô õ ü
Dutch	á à â ä é è ê ë i í ì î ï ó ò ô ö ú ù û ü	Romanian	â ă î ş ţ
English	none	Sami	á ï č d- ń n¸ š t- ž
Estonian	ä č õ ö š ü ž	Serbo-Croatian	ć č d- š ž
Faroese	á æ d- í ó øú ý	Slovak	á ä č d' é í Ĺ ñ ó ô ŕ š t' ú ý ž
Finnish	ä å ö š ž	Slovene	č š ž
French	à â æ ç è é ê ë î ï ô œ ù û ÿ	Spanish	á é í ó ú ü ñ
Gaelic	á é í ó ú	Swedish	ä å ö
German	ä ö ü ß	Turkish	ç ğ i ı ö ş ü
Hungarian	á ó ö ő ú ü ű	Upper Sorbian	ć č ě ł ń ó ř š ž
Icelandic	á æ∂ é í ó ö ú ý ♭	Welsh	â ê î ô û ŵ ŷ

We have started to think about this problem when faced with diacritics restoration in an electronic Romanian dictionary. No context is available in this case, and we deal with the dictionary itself and therefore methods relying on information encoded in a dictionary are not useful for this task. The role of a dictionary could be played by an *ad-hoc* vocabulary built from online corpora.

[1] The table lists only lower case letters. There is a corresponding upper case diacritic letter for each lower case letter. The information in this table was compiled from lists of diacritics in European languages available at http://www.tiro.com/di_intro.html

[2] Studies performed on bilingual parallel corpora have shown that the vocabulary built from an English text is about half the size of the vocabulary build for the same text written in a different language. Senseval competition [6] has also reported significantly lower precision for English with respect to other languages, in a word sense disambiguation task.

Nonetheless, for some languages the availability of online data is quite limited, especially when we place the constraint that the texts should contain diacritics.

It turns out that the applicability of previous methods is limited when:

(1) Electronic dictionaries are not available, or only limited size dictionaries are made public. Moreover, when the dictionary itself lacks diacritics, methods relying on diacritics restoration from dictionaries become useless.

(2) Tools for morphological and/or syntactic analysis, which are considered to be helpful for the problem of diacritics restoration, are inexistent or are not publicly available.

(3) Size of usable corpora containing diacritics is limited. The size of the corpora available on the Web or in other public forms influences the size of the vocabulary that can be built *ad-hoc* out of these texts. Moreover, Web publishers choose in many cases to avoid diacritics, for reasons of simplicity, uniformity or just the lack of means for diacritics encoding.

We propose in this paper a technique for diacritics restoration based on learning performed at letter level, rather than word level. The strongest advantage of this method is that it provides the means for generalization beyond words. The method is particularly useful for languages that lack large electronic dictionaries with diacritics. Well studied and widespread languages such as French and Spanish can benefit as well from this methodology in dealing with unknown words.

We have experimented this algorithm on diacritics restoration in Romanian texts, and a precision of over 99% at letter level was observed. Moreover, this method does not require any preprocessing steps, only a small size corpus of raw text with diacritics. Due to the simplicity of the algorithm, the processing speed is very high, about 20 pages of text per second, measured on a Pentium III running at 500MHz, with 250MB memory.

Specifically, instead of learning rules that apply at word level, such as *"anuncio should change to anunció when it is a verb"*, we are interested in learning rules at letter level, like *"s followed by i and preceded by white space should change to ş"*. This latest type of rules are more general and they have higher applicability when only small dictionaries are available, when many unknown words are encountered in the input text, or when there are no usable tools for morphological or syntactic analysis.

It is obvious that letters constitute the smallest possible level of granularity in language analysis, and therefore have the highest potential for generalization. Instead of having about 150,000 units that are potential candidates for the algorithm (the approximate size of the vocabulary of a language), we have more or less 26 characters that will constitute the entry to the disambiguation mechanism[3].

[3] The actual numbers depend on the language considered. It was shown, for instance, that about 85% of the French words do not have any spelling that includes diacritics, and hence only about 20,000 words are potentially ambiguous. On the other hand, only 7 letters are ambiguous in French.

2 Experimental Setup

The purpose of the experiments reported in this paper is to see whether learning at letter level is possible to the end of solving the problem of diacritics re-insertion. Besides providing an additional method for diacritics restoration, the purpose of doing learning at such a low level is to supply a methodology for languages that have only few lexical and semantic resources and for which diacritics restoration via learning at word level is hard to perform.

2.1 Data

During our experiments, we have considered the Romanian language. First, Romanian is not a widely spread language, and consequently it does not have many publicly available tools for preprocessing, and only small electronic dictionaries are available. Secondly, we had to solve a specific problem that required diacritics restoration. We have an electronic dictionary that we plan to use in further development of tools for Romanian. The size of the dictionary is fairly large, about 75,000 entries, but it has the disadvantage of containing no diacritics. Instead of relying on smaller dictionaries with diacritics, we chose to further study the problem of diacritics restoration and make use of our large dictionary. Furthermore, for the tools we plan to develop we need Romanian corpora, which usually lack diacritics, and once again diacritics restoration is required. Moreover, we have the means of comparison with learning at word level performed on the same language, through the experiments and results reported in [10].

We needed therefore a corpus of Romanian texts with diacritics. To this end, we downloaded articles from "România Literară"[4], which is a Romanian newspaper published weekly, with publications related mostly to literature. The newspaper started to have a version including diacritics beginning with year 2000. The entire collection available online at the date of the download (August 2001) adds up to 2,780 articles.

Next, we converted the HTML files into text files. We have paid particular attention only to characters specific to the Romanian language. Other characters such as ê, ç, etc., have been transformed into their equivalents, since we are interested in Romanian characters, rather than French or other languages. After this step, we were left with a corpus of about 3 million words.

Upper case letters have been converted into lower case. It is worth mentioning the case of the â and î letters in the Romanian language. Practically, the two letters have the same pronunciation but their spelling depends on their position within the word. At the beginning of a word, î should be used, whereas â spelling is employed inside the word. The spelling of this letter has been controversial over the years. A law from the sixties changed the spelling from â to î, with the only exception being the words with the root *Român*. In early nineties the old spelling was reintroduced, and so we ended up having inconsistent texts. It so happens that one can encounter different spellings for the same word. For

[4] Available at http://www.romlit.ro

instance, *cîntec* and *cântec*, both meaning *song*, can be sometimes found in the same source text. The "România Literară" newspaper is still applying the î spelling, with few exceptions (i.e. articles written by invited writers who chose to use â instead of î).

2.2 Learning Algorithms

We decided to use an instance based learning algorithm for our diacritics restoration task. The reasons for this decision are twofold. First, it was demonstrated that forgetting exceptions is harmful in Natural Language applications, and instance based learning algorithms are known for their property of taking into consideration every single training example when making a classification decision [2]. Secondly, this type of algorithms are efficient in terms of training and testing time. We have used the Timbl [3] implementation to run our learning experiments.

Additionally, we have performed similar experiments with a decision tree classifier, namely C4.5 [8]. The results obtained were similar with the cases when instance based learning is employed, but C4.5 has the capability of generating expressive rules, which are useful for practical implementations.

As we work at the low level of letters, the target attribute to be learned is constituted by the ambiguous letters. It can be therefore any of the *ambiguous* characters listed in Table 1. For Romanian, for instance, we have four pairs of ambiguous letters: *s* - *ş*, *t* - *ţ*, *a* - *ă* and *i* - *î*. Upper case diacritics are not considered as they have been previously converted to lower case. Due to the fact that the source data we are using applies the î spelling, as mentioned in Section 2.1, we do not have an *a* - *â* ambiguity, instead we have an *i* - *î* ambiguity. This fact does not imply any loss in generality. The conversion between the two spelling modes is very simple, using merely the position of the letter within the word, and thus different spellings do not affect in any way the final outcome of the algorithm.

2.3 Features

The features used in any learning algorithm have tremendous influence over the final accuracy. As mentioned in the introduction, we do not have the possibility of using part of speech taggers or any other morphological or syntactic analyzers. Furthermore, we do not want to rely on surrounding words, because the data we have is limited, and we would therefore encounter many cases of unknown words.

Hence, we decided for very simple features, for the extraction of whom no particular processing is required. We are using surrounding letters, with a special notation assigned to white spaces, commas, dots and colons (these characters may affect the learning process, as they are considered special characters by C4.5 and/or Timbl). This set of features performs surprisingly well in terms of accuracy, as shown later in the paper.

3 Results

For Romanian, there are four pairs of ambiguous letters. As mentioned earlier, we did not want to rely on any tags obtained with pre-processing tools, but simply on the information that can be extracted from raw text. Also, we are interested to find means for generalization, such that limited size corpora can be used to derive rules for diacritics re-insertion. Rather than learning from words, as it was the case with previous approaches, we want to learn rules from letters, as they constitute the smallest language units and enable learning from very small corpora.

For each ambiguous pair of letters, we scan the text and generate all possible examples encountered in the corpus. The attributes in an example are formed by N letters to the left and right of the ambiguous letter, and the target attribute is the ambiguous letter itself. We present below samples of feature vectors that were fed to the learning algorithm for the *s* - *ş* ambiguous pair. CO, DO and SP are the replacement codes we use to denote comma, dot or space.

l , i , n , SP , (, u , b , SP , i , n , s.
e , CO , SP , r , o , - , g , a , r , d , ş.
g , a , r , d , i , t , u , l , CO , SP , s.
e , SP , o , r , a , DO , SP , t , o , t , ş.

The number of examples extracted from the corpus depends on the pair of letters. From the entire set of three million words, we obtained 2,161,556 examples for the ambiguous pair *a* - *ă*, 2,055,147 for the pair *i* - *î*, 1,257,458 examples for *t* - *ţ*, and finally 866,964 examples for the *s* - *ş* pair.

The best accuracy was observed for an window size of ten surrounding letters (i.e. $N = 5$). We have therefore studied in more detail this case, including learning rates for the four pairs. Nevertheless, results are provided for various window sizes for comparative purposes.

Table 2 shows the results obtained for $N = 5$. The precision figures reported in this table are obtained using the instance based learning algorithm. We have performed tests with various sizes for the training set, ranging from 2,000,000 examples to as few as 10 examples, to the end of finding the learning rate and the minimum size of a corpus required for a satisfactory precision. All the experiments are performed with a test set size of 50,000 examples. A 10-fold cross validation scheme was used for more accurate results. The table also shows the baseline, defined here as the precision obtained when the most frequent letter is used out of the two letters found in a pair.

The results shown in Table 2 are plotted in Figure 1. It is interesting to observe that the most important part of the learning process is achieved with the first 10,000 examples. We have measured that about 100,000-250,000 running characters (approx. 25-60 pages of text) are needed to generate 10,000 examples with diacritics, which is a small corpus. From there on, a significant number of examples is required for every single percent of improvement in accuracy. We also show in bold the first precision figure that exceeds the baseline, as an indicative

Table 2. Results obtained in solving diacritics ambiguity, using an instance based learning algorithm and an window size of ten surrounding letters

	Ambiguous pair				
	a-*ă*	*a*-*ă*(2)	*i*-*î*	*s*-*ş*	*t*-*ţ*
Data set size	2,161,556	1,369,517	2,055,147	866,964	1,157,458
Baseline	74.70%	85.90%	88.20%	76.53%	85.81%
Training size	Precision obtained with a test set of 50,000 examples				
2,000,000	95.56%	-	99.69%	-	-
1,000,000	95.10%	99.14%	99.58%	-	98.75%
750,000	94.83%	98.97%	99.53%	99.07%	98.63%
500,000	94.57%	98.79%	99.46%	98.86%	98.40%
250,000	94.00%	98.37%	99.28%	98.87%	98.26%
100,000	93.03%	97.56%	98.96%	98.54%	97.81%
50,000	92.10%	96.86%	98.57%	98.13%	97.40%
25,000	90.99%	95.75%	98.11%	97.58%	96.92%
10,000	88.99%	93.75%	97.31%	96.53%	96.20%
5,000	87.56%	92.76%	96.65%	95.61%	95.10%
4,000	86.91%	91.86%	96.49%	94.99%	94.53%
3,000	86.39%	90.99%	96.19%	94.18%	94.30%
2,000	85.81%	89.93%	95.49%	93.47%	93.56%
1,000	83.49%	**88.36%**	93.78%	92.31%	91.85%
500	80.61%	85.66%	93.07%	90.75%	89.74%
250	77.89%	83.17%	92.75%	87.41%	**87.23%**
100	**74.80%**	84.04%	**91.41%**	82.13%	84.46%
50	72.79%	82.73%	88.05%	86.53%	77.54%
25	72.45%	81.34%	88.15%	**78.26%**	78.52%
10	73.38%	85.90%	88.20%	75.88%	85.81%

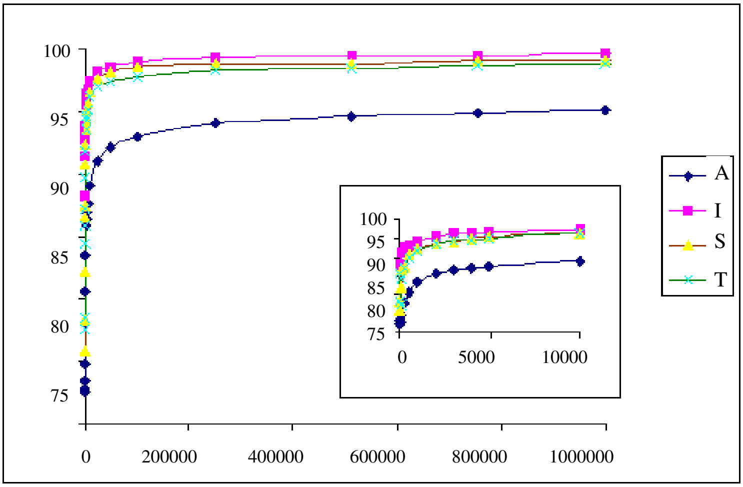


Fig. 1. Learning rates for the four ambiguous diacritics. The chart in the middle represents a zoom of the 0-10,000 range area.

of the smallest size of training set where a form of learning is observed. Notice that as few as 1,000 examples are enough to perform *some* learning.

Using the entire set of examples extracted from the corpus, the disambiguation of the *i-î* pair is almost 100% correct. For this diacritic letter, we now have one instance wrong out of 300 instances, whereas the baseline implies one instance wrong for every eight instances, therefore a significant improvement.

The worst precision is achieved in the case of *a-ă* pair. From a simple error analysis, it turns out that the main reason for this is the fact that many Romanian nouns have their base form ending in *ă*, whereas their articulated form ends in *a*. For instance, *masă* and *masa* are two forms, one articulated and one not, for the same noun *table.* Also, some verbs have two different tenses with the only difference standing in an *a* - *ă* ending letter. The learner is therefore tricked by many identical usages for these letters. A simple solution for this is to avoid in the learning process those examples that contain an *a* or *ă* letter at the end of a word. The results obtained under this simplifying assumption are reported in Table 2 under the heading *a-ă(2)*[5]. As shown in the table, more than four percents are gained in precision with this simple condition (this translates into 87% error reduction).

We have also employed C4.5 on the same training data, but no improvements were observed with respect to the results from Table 2. The disadvantage of using C4.5 for this task is that the learning phase is slower than with the Timbl implementation. On the other hand, C4.5 has the capability of generating expressive rules. *"L_1=e and L_2=space then s"*(99.5%), *"L_1=t and L_2=space then s"* (98.7%), *"$L_{-4} = p$ and L_{-1}=v and L_1=t L_2=e then ş"*(95.5%), are examples of such rules, where L_i denotes a surrounding letter at the relative position i with respect to the ambiguous letter. Notice that these rules do not say anything about whether or not the letters belong to one single word. The learning algorithm simply relies on letters, regardless of the word they belong to. Consequently, pseudo-homographs words (as in *peste* and *peşte* - see Section 1) are equally addressed by this method, as the algorithm has the capability of going across words.

3.1 Different Window Sizes

We have experimented various window sizes to determine the size of the context that would best model our problem. We considered window sizes of two, six, ten, fourteen and eighteen surrounding letters (i.e. $N = 1, 3, 7, 9$). Comparative results are reported in Table 3. These figures should be compared with the uppermost row in Table 2 (the N=5 column in the current table).

When no context is available, window sizes of N=3 can be used without losing much in precision. Nevertheless, as stated earlier, the best accuracy is attained for a window of ten surrounding letters (N=5).

[5] Generality is not affected in by our assumption that ending *a* or *ă* letters are not considered during the learning process. This case of ambiguity can be easily solved by finding words articulation, if any, which is a fairly simple task.

Table 3. Comparative results for various window sizes

Ambiguous pair	Window size				
	N=1	N=3	N=5	N=7	N=9
a-ă(2)	88.63%	98.79%	99.14%	99.10%	99.10%
i-î	94.18%	99.13%	99.69%	99.68%	99.43%
s-ş	88.09%	99.06%	99.07%	99.02%	99.00%
t-ţ	89.45%	98.57%	98.75%	98.67%	98.25%

3.2 Comparison with Related Work

These results are best compared with the work reported by Tufiş and Chiţu [10], who employed the same language as in our experiments.

According to Tufiş and Chiţu, the task of diacritics recovery in Romanian is harder than with other languages, as Romanian makes more intensive use of diacritics. As reported in their experiments, only about 60% of the Romanian words are diacritics free, compared to the studies reported in [9] which show that about 85% of the French words are spelled with no accents.

The approach presented by Tufiş and Chiţu uses dictionaries, a tokenizer and part of speech tagger, and learning is performed at word level, for an overall performance of 97.4%. We cannot directly compare our results, as both methods and evaluations are fundamentally different. The average precision of 99% we have obtained is measured at letter level, whereas the accuracy they report is determined at word level.

Our methodology overcomes previous approaches in that very high precisions and processing speeds are obtained without any preprocessing tools or dictionaries being required, and therefore this algorithm is applicable to any language, with the only requirement being a relatively small corpus of texts with diacritics.

4 Conclusions

We have presented a method for diacritics restoration based on learning mechanisms that act at letter level. This technique is new to our knowledge, and its strongest advantage stands in its capability of generalization beyond words. No preprocessing steps are required, and no tools or dictionaries are employed. The only requirement is a relatively small corpus of texts with diacritics.

The method is particularly useful for languages that lack large electronic dictionaries and morphological or syntactic tools. Raw texts are fed to the learning mechanism, and accuracies of over 99% at letter level are reported. Moreover, due to its simplicity, processing speeds of about 20 pages of text per second can be attained.

References

1. Angell, R., Freund, G., and Willett, P. Automatic spelling correction using a trigram similarity measure. *Information Processing and Management 19*, 4 (1983), 255–261.

2. Daelemans, W., van den Bosch, A., and Zavrel, J. Forgetting exceptions is harmful in language learning. *Machine Learning 34*, 1-3 (1999), 11–34.
3. Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosch, A. Timbl: Tilburg memory based learner, version 4.0, reference guide. Tech. rep., University of Antwerp, 2001.
4. El-Bèze, M., Mérialdo, B., Rozeron, B., and Derouault, A. Accentuation automatique des textes par des méthodes probabilistes. *Techniques et sciences informatique 16*, 6 (1994), 797–815.
5. Galicia-Haro, S., Bolshakov, I., and Gelbukh, A. A simple Spanish part of speech tagger for detection and correction of accentuation error. In *Text, Speech and Dialogue - Second International Workshop, TSD'99, September 1999, Proceedings* (Plzen, Czech Republic, 1999), vol. 1692 of *Lecture Notes in Computer Science*, Springer, pp. 219–222.
6. Kilgarriff, A., Ed. *SENSEVAL-2* (to appear).
7. Nagy, G., N., N., and Sabourin, M. Signes diacritiques: perdus et retrouvés. In *Actes du 1er Collque International Francophone sur l'Écrit et le Document CIFED '98* (Queébec, Canada, 1998), pp. 404–412.
8. Quinlan, J. *C4.5: programs for machine learning.* Morgan Kaufman, 1993.
9. Simard, M. Automatic insertion of accents in French text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-3* (Granada, Spain, 1998).
10. Tufiş, D., and Chiţu, A. Automatic diacritics insertion in Romanian texts. In *Proceedings of the International Conference on Computational Lexicography COMPLEX'99* (Pecs, Hungary, June 1999).
11. Yarowsky, D. *Corpus-based techniques for Restoring accents in Spanish and French Text.* In *Natural Language Processing Using Very Large Corpora.* Kluwer Academics Publisher, 1999, pp. 99–120.

A Comparative Study of Information Extraction Strategies

Ronen Feldman, Yonatan Aumann, Michal Finkelstein-Landau,
Eyal Hurvitz, Yizhar Regev, and Ariel Yaroshevich

ClearForest Ltd.
Or Yehuda, Israel
Tel. 972 3 7350000
{ronen,yonatan,michal,eyal,ryizhar,ariel}@clearforest.com

Abstract. The availability of online text documents exposes readers to a vast amount of potentially valuable knowledge buried therein. The sheer scale of material has created the pressing need for automated methods of discovering relevant information without having to read it all. Hence the growing interest in recent years in Text Mining.

A common approach to Text Mining is Information Extraction (IE), extracting specific types (or templates) of information from a document collection. Although many works on IE have been published, researchers have not paid much attention to evaluate the contribution of syntactic and semantic analysis using Natural Language Processing (NLP) techniques to the quality of IE results.

In this work we try to quantify the contribution of NLP techniques, by comparing three strategies for IE: naïve co-occurrence, ordered co-occurrence, and the structure-driven method – a rule-based strategy that relies on syntactic analysis followed by the extraction of suitable semantic templates. We use the three strategies for the extraction of two templates from financial news stories. We show that the structure-driven strategy provides significantly better precision results than the two other strategies (80-90% for the structure-driven compared with about only 60% for the co-occurrence and ordered co-occurrence). These results indicate that a syntactical and semantic analysis is necessary if one wishes to obtain high accuracy.

Keywords: Information Extraction, Natural Language Processing

1 Introduction

The need for automated methods to extract information from online text documents is constantly growing in this age of information overload. Since most of the information available in digital format is unstructured, there is a growing interest in Text Mining that focuses on extracting data from textual sources.

One of the most common processes of Text Mining is known as Information Extraction (IE).

In Information Extraction, key concepts (facts or events concerning entities or relationships between entities discussed in the text) are defined in advance and then the text is searched for concrete evidence for the existence of such concepts. For example,

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 349–359, 2002.

in financial news documents we may be interested in information about acquisitions of a certain company by another company. Such information may be typically given by a sentence such as:

ABC Inc, the leading manufacturer of electronic toys, has successfully completed the acquisition of DFG Corporation.

The above would be converted into a fact, or an instance of the template:

Table 1. Acquisition template

Acquisition:	Company1	Company2
	ABC Inc.	DFG Corporation

Thus, structured information is created from the unstructured text.

Another example is what we will call Person-Left-Position: information about the fact that a certain employee in a company left the company (willingly or unwillingly).

An example of a sentence delivering this information is:

Andrx Group (Nasdaq:ADRX) today announced that Chih-Ming Chen, Ph.D. has resigned from his position as the Company's Chief Scientific Officer.

The corresponding template is:

Table 2. Person-Left-Position template

Person-Left-Position:	Company	Person	Position
	Andrx Group	Chih-Ming Chen	Chief Scientific Officer

There are several different algorithms and methods to perform Information Extraction. These are based on various levels of semantic and syntactic analysis of the text.

Existing IE systems include systems based on hand-crafted rules that "understand" the text and manage the filling of the template slots, as ([1], [4]), as well as trainable systems (WAVE [2], CRYSTAL ([5],[6]).

Trainable systems have the advantage over hand-crafted systems that they can be extended more easily and require less domain knowledge. However, the performance of the trainable systems is usually not as good as the hand-crafted ones (precision and recall-wise).

In this paper we compare the performance of three strategies for Information Extraction. We show a general method for performing semantic and syntactic analysis of the text that enables constructing of "structure-driven rules" that achieve high levels of precision (80%-90%).

We compare it to two other strategies: Co-occurrence strategy and Ordered Co-occurrence strategy. The co-occurrence strategy is much simpler than the structure-driven strategy, seeking only the existence of relevant keywords in the text, without reference to their syntactic or semantic role therein. The Ordered Co-occurrence strategy is similar to the Co-occurrence strategy, but here constraints regarding the position of the keyword within the sentence, relatively to the entities involved in the template are applied. We show that, while these two strategies allow rapid constructing of rules, the precision of such rules is consistently relatively low (50%-60%). That is:

although the structure-driven rules require more labor, the precision results clearly justify the additional work.

The remainder of this paper is organized as follows. In Section 2 we describe the three strategies in details. In Section 3 we describe the experimental evaluation of the three strategies in the DIAL language. In Section 4 we describe the comparison performed and its results. We discuss the results in Section 5.

2 Information Extraction Strategies

An advanced task in Information Extraction systems involves filling up predefined templates after identifying semantic relationships between different entities in the text.

In this section we will describe a hierarchy of three strategies, evaluated and compared in this study, for extracting semantic relationships between entities in an Information Extraction system. The hierarchy ranges from a simple co-occurrence method to a sophisticated rule-based method for extracting information from a single sentence.

All strategies are based on an underlying system for entity recognition, namely a system that extracts proper names and classifies them according to a predefined set of categories, such as Company, Person, Location and so forth. The identified entities are utilized by the different methods as-is without any further analysis.

For illustration purposes, we will use a binary relation between companies called ACQUIRED:

ACQUIRED(Company1, Company2) means that an acquisition event took place between Company1 and Company2, namely either Company1 acquired Company2 or the opposite.

2.1 The Co-occurrence Strategy

This method follows the simple definition of the term co-occurrence: "an event or situation that happens at the same time as or in connection with another" (http://www.dictionary.com). It seeks only the existence of the relevant entities and keywords in the same sentence.

This method is implemented by a simple pattern matching mechanism without referring to any syntactic or semantic role of the searched entities and keywords.

For identifying the ACQUIRED relationship, this strategy searches for sentences that contain the following elements:

- Two different companies: identified at an earlier stage of entity recognition, as described above
- Acquisition keyword: a keyword taken from a lexicon of acquisition nouns and verbs; e.g. acquire, bought, acquisition and so on.

The following sentences were extracted as candidates for the ACQUIRED relationship by the Co-occurrence strategy.

The first one is correct; the second is incorrect. Co-occurrence elements (companies and acquisition keywords) are bolded.

- *Recently,* ***Sovereign*** *entered into a definitive agreement with* ***Main Street Bancorp, Inc.*** *("Main Street") for* ***Sovereign*** *to* ***acquire Main Street****.*
- ***Ask Jeeves*** *deploys its solutions on* ***Ask Jeeves*** *at* ***Ask.com****,* ***Ask Jeeves for Kids*** *at* ***AJKids.com****,* ***DirectHit.com*** *and* ***Jeeves Tours****, to help companies target and* ***acquire*** *qualified prospects online and to provide consumers with real-time access to information, products and services.*

2.2 The Ordered Co-occurrence Strategy

This method is an enhancement of the naïve co-occurrence method. We choose to enhance the simple co-occurrence results by adding order constraints on the matched pattern. Such constraints are intended to heuristically preclude the extraction of syntactically invalid or semantically unreasonable events.

The simple co-occurrence strategy might extract sentences where the searched elements by no means form a valid structure for correct semantic relationships. For example, if the keyword searched for is a transitive verb, it should be located between the entities, neither precede them nor follow them.

Within the ACQUIRED relationship, forcing the keyword "acquire" to separate one company from the other helps in eliminating trivial precision errors like in the following sentence, where the transitive verb precedes both companies:

The building was vacant at the time it was ***purchased*** *and is now 100% leased to* ***Deltek Systems*** *and* ***Perot Systems*** *70,524 square feet executed in late July of 2001).*

Defining the appropriate constraints requires a shallow linguistic understanding of the domain in order to determine the appropriate order between the searched elements.

2.3 The Structure-Driven Rule-Based Strategy

This strategy is based on noun phrase and verb phrase identification augmented by linguistic and semantic constraints.

In this strategy, the extraction of the predefined semantic relationships is performed in the means of deep syntactic and semantic analysis of the sentences. Naturally, this method involves more human effort, but we will show that it consistently achieves higher precision rates.

For example, for the ACQUIRED relationship we search for a Subject-Verb-Object structure, requiring the Subject and Object to be companies and the Verb to be tensed where its head belongs to the Acquisition lexicon (e.g. *acquire*, *purchase*). The constraints require, for example, different Subject and Object (i.e. two different companies - a semantic constraint) and verb-preposition agreement (syntactic constraint).

As indicated by this example, this method requires a skilled developer and entails a fairly elaborate development effort. The advantage, as will be discussed later in this document, is that its qualitative results are by far better than the two simpler methods.

The implementation of the Structure-Driven processing is based on a general multi- level NLP system. We give here a brief description of its different layers:

Layer 0 - POS (Part of Speech) Tagger: Assigning POS tags (noun, proper noun, verb, adjective, adverb, preposition, and so on.) to each word.

Layer 1 - Noun Phrase and Verb Phrase Grouper: Grouping together the head noun with its left modifiers (for example: "*massive payment agreement*") and, for verbs, chunking a main verb with its auxiliaries, like in "*has been acquired*" or "*is already being incorporated*".

Layer 2 - Verb and Noun Pattern Extractor: Extracting larger verb and noun phrases, on the basis of semantic requirements. Examples: "*said Monday it has acquired*" and "*announced plans to acquire*".

In general, this mechanism matches verbs and nouns with their complements, as specified in their sub-categorization properties. This level is semantically-oriented: it keeps track of the semantic features of a pattern, as expressed by various elements such as adverbs, tense and voice of the verb group and certain syntactic structures. This way, the system can identify complex patterns that still express a basic relation given by the rightmost element of the pattern. For example, in "*SignalSoft has expanded its application portfolio with the acquisition of mobilePosition(R)*", "*has expanded its application portfolio with the acquisition*" is a Verb Pattern based on the keyword "*acquisition*", that is used to extract acquirer-acquired relations.

Layer 3 - Named Entity Recognizer: recognition of companies, persons, products, and so forth.

Layer 4 - Nominal Expression Extractor: Matching nominal phrases that contain entities as arguments, such as "*Microsoft's acquisition of Visio*", or "*The acquisition by Microsoft of Visio*".

Layer 5 - Template ("Event") Extractor: Rule-based extraction of patterns at a full sentence or phrase level.

For example, the full sentence "*Microsoft announced Monday it has acquired Visio*" is matched using the Verb Pattern of Layer 2 "*announced Monday it has acquired*". This layer uses a lexicon of keywords, nouns and verbs that are relevant to the specific template. (For example, in the case of the Acquisition template, verbs such as "*acquire*", "*buy*", "*bid*"). This layer includes extraction of other elements that are needed to shallow parse sentences and additional information regarding a template (such as adverbial phrases, appositive clauses, dates, and so forth.).

3 Implementation in the DIAL Extraction Language

In this section we will briefly describe the framework used for building our IE system, a rule-based general IE language developed at ClearForest (DIAL).

DIAL is a declarative, rule-based language, designed specifically for IE. The complete syntax of DIAL is beyond the scope of this paper. In the next item we present of the key elements relevant to this work. Further details and examples are presented in [3].

DIAL enables the user to implement separately the different operations required for performing IE: tokenization, zoning (recognizing paragraph and sentence limit), and morphological and lexical processing, parsing and domain semantics. DIAL has built-in modules that perform the general tasks of tokenization and part-of-speech tagging. In addition, we have developed a general library of rules that perform Noun Phrase and Verb Phrase grouping and separate libraries for recognizing relevant Entities, such as *companies* or *persons*.

3.1 Survey of DIAL's Basic Elements

As stated above, DIAL is a rule-based language. DIAL "program" is phrased as a logic program - a Rule Book.

A Rule Book, Γ, is a conjunction of Definite clauses ("rules") C_i : $H_i \leftarrow B_i$, where C_i is a clause label, H_i ("the head") is a literal and B_i = (B_{i1},B_{i2} ...) = (P_i,N_i) (the clause's body), where P_i = (p_{ij}) is a series of Pattern Matching Elements and N_i = {n_{ij}} is a set of constraints operating on P_i.

The clause C_i : $H_i \leftarrow B_i$ represents the assertion that H_i is implied (or, in our context, that an instance of H_i is defined) by the conjunction of the literals in P_i while satisfying all the constraints in N_i.

Typically, the H_i is the template (event) sought by the Information Extraction process (such as Acquisition or Person-Left-Position). The practical meaning of the above formal definition is that whenever the series of pattern matching elements P_i is found in the text and the constraints set N_i is fulfilled , deduce that the template H_i occurs in that text fragment.

A Pattern Matching Element p_{ij} may be:

- An explicit token (String) found in the text - e.g. *"announces"*
- A word class element: a phrase from a predefined set of phrases that share a common semantic function. Example: the word class *wcResignation* includes the words: *"resignation"*, *"retirement"* and "*departure*".
- A predicate call - e.g. Company(C)

See [3] for a more complete list of DIAL elements.

A constraint n_{ij} may be used for carrying out on-the-fly Boolean checks on relevant segment of texts matched by the pattern matching elements. A constraint is typically implemented by using a suitable Boolean function, for example: InWC, which returns TRUE if the tested text segment is a member of the tested word class.

For example, verify(InWC(*P*, @wcAnnounce)) means that the *P* pattern matching element must be a member of the word class wcAnnounce.

3.2 DIAL Rule – An Example

Below we give an example of a rather simple DIAL rule for extracting a common Person-Left-Position template:

```
PersonLeftPosition(Person_Name, Position, Company_Name) :-
Company(Company_Name)
Verb_Group(V_Stem,V_Tense,V_Modifiers)

Noun_Group(N_Determiner,N_Head,N_Stem,N_Modifiers)
"of"
Person(Person_Name)
[ "as" ]
wcCompanyPositions
verify(InWC(V_Stem,@wcAnnounce))
verify(InWC(N_Stem,@wcResignation)) ;
```

The rule above corresponds to a common pattern in financial news announcing resignation or retirement, as in: "*International Isotopes Inc Announces the Resignation of Dr. David Camp As President and CEO*".

The meaning of the code above is as follows: Extract a *Person-Left-Position* template from this text segment if a *Company* was identified, followed by a Verb Group whose stem is included in the *wcAnnounce* word class (that includes verb such as "*announce*" or "*report*"), followed by a Noun Group (that may include a determiner such as "*the*") whose head is a member of the word class *wcResignation* (This word class includes the terms "*resignation*", "*retirement*" and "*departure*"), followed by the word "*of*", followed by a *person* name, followed by the optional word "*as*" and a term from *wcCompanyPosition*, a word class that includes common positions of executives such as "*President*", "*CEO*", "*CFO*" and so forth.

The Company and Person predicates are implemented in a separate module that is executed before the *Person-Left-Position* module.

4 Experimental Evaluation

In order to test the three strategies we have conducted two separate experiments. In each experiment we tested the results of the three strategies for the extraction of one concept (template) – in one experiment the *Acquisition* template was extracted, and in the second experiment – the *Person-Left-Position* template.

4.1 The Data Source

For each experiment, we created a news article collection by downloading documents from the *NewsAlert* site (www.newsalert.com) using a suitable set of keywords:

For the *Acquisition* template, the keyword set included terms such as "acquisition", "acquire", "buy", "bid" and "purchase". The collection included 500 document published in September 2001.

For the *Person-Left-Position,* the keyword set included terms such "resign", "retire", "resignation", "fire" and "step down". The collection included 1725 documents published in August-September 2001. (The number of required documents for this template was bigger than the number required for the *Acquisition* template, because this template is less frequent).

The *NewsAlert* site aggregates news document from a number of sources, including, among others, *Reuters*, *PRNewsWire* and *BusinessWire.*

4.2 Evaluating the Different Strategies

It is important to note that the set of structure-driven rules were written prior to downloading the test collection, using the DIAL NLP libraries described in section 2.3 above. These rules were written based on a small set of financial news documents we had previously downloaded. We have also created sets of rules implementing the two other strategies.

For each of the two templates, we executed separately the rules written according to each of the three strategies using the ClearStudio environment developed at ClearForest. The ClearStudio environment creates as a result a file of all the instances found and enables viewing the location within the original document from which the instance was extracted and to classifying that instance (for example, as correct or incorrect). See Figure 1 below.

In the *Acquisition* template, all rules were required to extract the two companies involved in the Acquisition relationship. For the purpose of the experiment, we ignore modalities, so that an extracted Acquisition event could be either an actual, possible, pending or even a cancelled acquisition. In the *Person-Left-Position* template, an instance was extracted if it included the person name and the company from which she or he retired, or, the person name and the position she or he had held.

For each of the two templates, we executed separately the rules written according to each of the three approaches using the ClearStudio environment developed at ClearForest. The ClearStudio environment creates as a result a file of all the instances found and then, to view the location within the original document from which the instance was extracted and to classify that instance (For example, as correct or incorrect). See Figure 1 below.

4.3 The Results

The results for the two templates are given in tables 3 and 4 below.

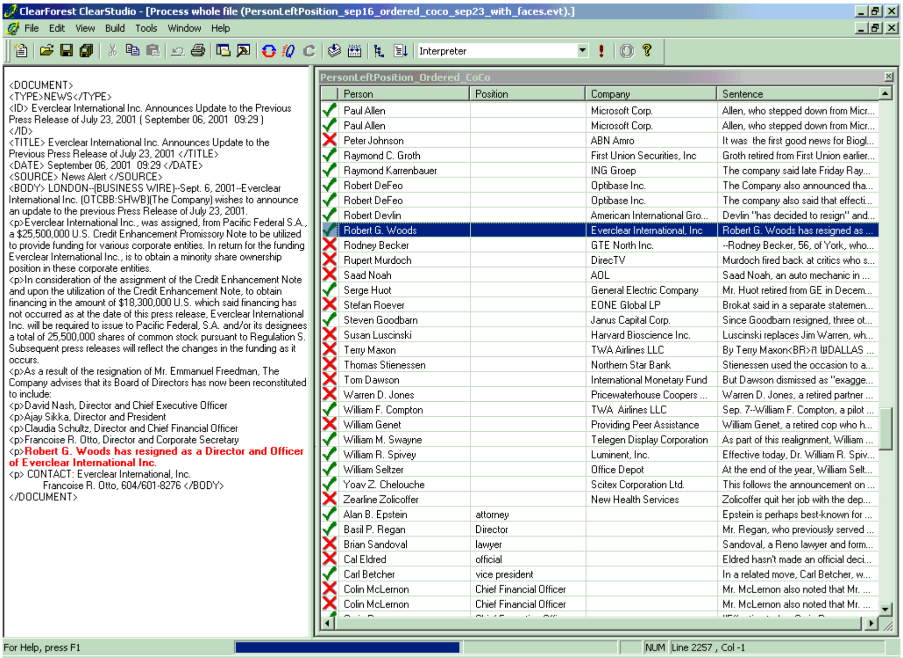
Table 3. Acquisition template results (recall is relative to a total of 260 events)

Method	Co-occurrence	Ordered Co-occurrence	Structure-Driven
Correct Instances	244	246	135
Incorrect Instances	201	132	16
Total Instances	445	396	151
Precision	54.8%	62.1%	89.4%
Recall	93.8%	94.6%	51.9%

Table 4. Person-Left-Position template results (recall is relative to a total of 363 events)

Method	Co-occurrence	Ordered Co-occurrence	Structure-Driven
Correct Instances	353	266	174
Incorrect Instances	250	165	44
Total Instances	603	431	218
Precision	58.5%	61.7%	79.8%
Recall	97.2%	73.2%	47.9%

Remark: The recall rates given in tables 3 and 4 are relative to the total number of correct instances found during the assessment of the extraction results. Recall of naïve co-occurrence does not reach 100% because this method sometimes picks up the wrong pair of companies, missing out the correct pair in the same sentence. The alter-


ClearForest ClearStudio - [Process whole file (PersonLeftPosition_sep16_ordered_coco_sep23_with_faces.evt).]
File
Edit
View
Build
Tools
Window
Help
Interpreter
PersonLeftPosition_Ordered_CoCo
For Help, press F1
NUM
Line 2257 , Col -1

```
<DOCUMENT>
<TYPE>NEWS</TYPE>
<ID> Everclear International Inc. Announces Update to the Previous
Press Release of July 23, 2001 ( September 06, 2001  09:29 )
</ID>
<TITLE> Everclear International Inc. Announces Update to the
Previous Press Release of July 23, 2001 </TITLE>
<DATE> September 06, 2001  09:29 </DATE>
<SOURCE> News Alert </SOURCE>
<BODY> LONDON--(BUSINESS WIRE)--Sept. 6, 2001--Everclear
International Inc. (OTCBB:SHWB)(The Company) wishes to announce
an update to the previous Press Release of July 23, 2001.
<p>Everclear International Inc., was assigned, from Pacific Federal S.A.,
a $25,500,000 U.S. Credit Enhancement Promissory Note to be utilized
to provide funding for various corporate entities. In return for the funding
Everclear International Inc., is to obtain a minority share ownership
position in these corporate entities.
<p>In consideration of the assignment of the Credit Enhancement Note
and upon the utilization of the Credit Enhancement Note, to obtain
financing in the amount of $18,300,000 U.S. which said financing has
not occurred as at the date of this press release, Everclear International
Inc. will be required to issue to Pacific Federal, S.A. and/or its designees
a total of 25,500,000 shares of common stock pursuant to Regulation S.
Subsequent press releases will reflect the changes in the funding as it
occurs.
<p>As a result of the resignation of Mr. Emmanuel Freedman, The
Company advises that its Board of Directors has now been reconstituted
to include:
<p>David Nash, Director and Chief Executive Officer
<p>Ajay Sikka, Director and President
<p>Claudia Schultz, Director and Chief Financial Officer
<p>Francoise R. Otto, Director and Corporate Secretary
<p>Robert G. Woods has resigned as a Director and Officer
of Everclear International Inc.
<p> CONTACT: Everclear International, Inc.
          Francoise R. Otto, 604/601-8276 </BODY>
</DOCUMENT>
```

	Person	Position	Company	Sentence
✓	Paul Allen		Microsoft Corp.	Allen, who stepped down from Micr...
✓	Paul Allen		Microsoft Corp.	Allen, who stepped down from Micr...
✗	Peter Johnson		ABN Amro	It was the first good news for Biogl...
✓	Raymond C. Groth		First Union Securities, Inc	Groth retired from First Union earlier...
✓	Raymond Karrenbauer		ING Groep	The company said late Friday Ray...
✓	Robert DeFeo		Optibase Inc.	The Company also announced tha...
✓	Robert DeFeo		Optibase Inc.	The company also said that effecti...
✓	Robert Devlin		American International Gro...	Devlin "has decided to resign" and...
✓	Robert G. Woods		Everclear International, Inc	Robert G. Woods has resigned as ...
✗	Rodney Becker		GTE North Inc.	--Rodney Becker, 56, of York, who...
✗	Rupert Murdoch		DirecTV	Murdoch fired back at critics who s...
✗	Saad Noah		AOL	Saad Noah, an auto mechanic in ...
✓	Serge Huot		General Electric Company	Mr. Huot retired from GE in Decem...
✗	Stefan Roever		EONE Global LP	Brokat said in a separate statemen...
✓	Steven Goodbarn		Janus Capital Corp.	Since Goodbarn resigned, three ot...
✗	Susan Luscinski		Harvard Bioscience Inc.	Luscinski replaces Jim Warren, wh...
✗	Terry Maxon		TWA Airlines LLC	By Terry Maxon<BR>[illegible]DALLAS ...
✗	Thomas Stienessen		Northern Star Bank	Stienessen used the occasion to a...
✗	Tom Dawson		International Monetary Fund	But Dawson dismissed as "exagge...
✗	Warren D. Jones		Pricewaterhouse Coopers ...	Warren D. Jones, a retired partner ...
✓	William F. Compton		TWA Airlines LLC	Sep. 7--William F. Compton, a pilot ...
✗	William Genet		Providing Peer Assistance	William Genet, a retired cop who h...
✓	William M. Swayne		Telegen Display Corporation	As part of this realignment, William ...
✓	William R. Spivey		Luminent, Inc.	Effective today, Dr. William R. Spiv...
✓	William Seltzer		Office Depot	At the end of the year, William Selt...
✓	Yoav Z. Chelouche		Scitex Corporation Ltd.	This follows the announcement on ...
✗	Zearline Zolicoffer		New Health Services	Zolicoffer quit her job with the dep...
✓	Alan B. Epstein	attorney		Epstein is perhaps best-known for ...
✓	Basil P. Regan	Director		Mr. Regan, who previously served ...
✗	Brian Sandoval	lawyer		Sandoval, a Reno lawyer and form...
✗	Cal Eldred	official		Eldred hasn't made an official deci...
✓	Carl Betcher	vice president		In a related move, Carl Betcher, w...
✗	Colin McLernon	Chief Financial Officer		Mr. McLernon also noted that Mr. ...
✗	Colin McLernon	Chief Financial Officer		Mr. McLernon also noted that Mr. ...

Fig. 1. Sample Screen from the ClearStudio Environment as used for the Experiment

native to picking up only a single pair would be to extract *all* pairs, but that would clearly result in much poorer precision rates, as typically *½n(n-1)-1* incorrect instances would be automatically extracted from each sentence with an Acquisition keyword containing *n* companies.

5 Discussion and Conclusions

For both templates, the precision results for the structure-driven strategy were significantly better than the two other simpler methods. For both methods, the ordered co-occurrence strategy performed only slightly better than the naïve co-occurrence. Precision results, under the structure-driven method were better for the *Acquisition* template than for the *Person-Left-Position* template.

The structure-driven method performs always better since it filters many noisy instances, in which the searched keyword may have a totally different meaning. For example, the *Person-Left-Position* co-occurrence rules produced as an instance the following sentence ("*quit*" was one of the keywords*):*

David Oxlade, chief executive of Xenova Group Plc, the company behind the project, said the vaccine could eventually have an important role to play in helping smokers ***quit.***

Similarly, the Acquisition co-occurrence rules produced the sentence:

ZAMBA's clients have included Aether Systems, Best ***Buy****, CompuCom, GE Medical Systems, BellSouth, Hertz, General Mills, Symbol Technologies and Towers Perrin.*

The above sentence was extracted because of the "*buy*" keyword, although, clearly none of the companies mentioned have an Acquisition relationship between them.

Rather than being an isolated case, the problem exhibited by the last example of Acquisition is very common in the domain of business news, as Acquisition keywords and particularly "acquisition" is a part of the name or description of many (acquisition) companies.

We believe that the lower precision rate for the *Person-Left-Position* template is due to the fact that this concept is more complex and can be phrased in many ways, and as a result requires more rules (patterns). Sentences that discuss a retirement of a person mention several persons and / or companies, making it more difficult to extract the correct ones. Specifically, we observed that while the rules beginning with the Company (such as in the example in Section 1 above) have very good precision, the rules beginning with the Person have worse precision.

Rather surprisingly, the ordered co-occurrence strategy proved little better than the naive co-occurrence. Several reasons may be given for this result.

Regarding the Acquisition event, ordered co-occurrence only improved results in case the pattern was based on a subject-verb-object pattern. For nominal patterns no such ordering is relevant[1].

Another reason is that, while ordered co-occurrence may rule out sentences in which the relevant keyword has a different *syntactical* function, it fails to handle more complex *semantic* patterns. A common problem we encountered is patterns involving a more complex relationship between more than two entities. For example:

John F. Hoffner, 54, replaces Charles W. Duddles, 61, who announced in March that he would retire from Jack in the Box this year.

Both the co-occurrence and the ordered co-occurrence strategies fail to find that the first person (*Mr. Hoffner*) is *not* the retiring one, but rather *succeeds* Mr *Duddles*. The ordered co-occurrence strategy checks only that the verb group (*would retire)* follows the entity. But it cannot observe that this verb actually refers to the second person.

The above results indicate that if we are interested in all the references to an Acquisition or *Person-Left-Position* template, then the co-occurrence strategy is better recall-wise, since it extracts significantly more instances

One of the main recall problems of the structure-driven method occurs in sentences in which some of the entities can only be anaphorically resolved, since it is not explicitly mentioned within the pattern, as exemplified by the sentence below ("*acquisition of Pifco Holding*"). The naïve co-occurrence method is not sensitive to sentence structure, so it can identify the company as long as it appears somewhere in the sentence. For a structure-driven method to overcome this problem, it has to employ an anaphora resolution mechanism. We actually do employ such a mechanism, but restrict it to the sentences that contain an explicit anaphoric expression.

[1] While "Microsoft's acquisition of Visio" has the order subject-predicate-object (like in the case of verbs), in "the acquisition by Microsoft of Visio", for instance, both arguments follow the predicate.

The usage of a simplistic anaphoric resolution mechanism in sentences that lack such expressions may lower the precision rate.

Salton, Inc. (NYSE: SFP), today reported its fiscal 2001 fourth quarter and year-end results for the period ended June 30, 2001, which includes operating results from June 1, 2001 through June 30, 2001 resulting from the previously announced acquisition of Pifco Holding PLC.

Although the co-occurrence strategy produces more instances, our analysis shows that many of those instances occur within clauses and refer to information already known from other parts of the same document or from other documents. Note that many anaphoric cases, in this domain, fall under this category as well.

To illustrate this, consider the following sentence, which may be extracted only using the co-occurrence strategy:

"My advice is to listen to Jim Kelly," he tells a colleague, referring to their boss, the retiring UPS chairman who announced Thursday he will step down at the end of the year.

The retirement of Jim Kelly was discussed several times in the financial news in August 2001, and at least once it was within a clear template that was extracted by our structure-driven rules:

"UPS Chairman and Chief Executive Jim Kelly said Thursday he will retire".

Clearly, the structure-driven method requires much more extensive initial work. However, the results of this paper indicate that this is necessary if one wishes to obtain high accuracy. In any given case, the cost-benefit tradeoff must be weighed, in order to decide on the best strategy for the given application.

Besides the fact that the structure-driven strategy is clearly superior over the co-occurrence strategy in precision, it also achieves a high recall rate for recent events and thus is preferable for practical applications that aim to extract precise information from news.

References

1. Appelt D. E., Hobbs J., Bear J., Israel D. and Tyson M., 1993. "FASTUS: A Finite-State Processor for Information Extraction from Real-World Text", Proceedings. *IJCAI-93*, Chambery, France, August 1993.
2. Aseltine J., 1999. "WAVE: An Incremental Algorithm for Information Extraction". *In Proceedings of the AAAI 1999 Workshop on Machine Learning for Information Extraction.*
3. Feldman R., Liberzon Y, Rosenfeld B., Schler J. and Stoppi J., 2000. "A Framework for Specifying Explicit Bias for Revision of Approximate Information Extraction Rules". *KDD 2000*: 189-199.
4. Lin D. 1995. University of Manitoba: Description of the PIE System as Used for MUC-6 . In *Proceedings of the Sixth Conference on Message Understanding (MUC-6)*, Columbia, Maryland.
5. Soderland S., 1996. "Learning Text Analysis Rules for Domain-specific Natural Language Processing". *Ph.D. thesis, technical report UM-CS-1996-087* University of Massachusetts, Amherst.
6. Soderland S., Fisher D., and Lehnert W., 1997. "Automatically Learned vs. Hand-crafted Text Analysis Rules". *CIIR Technical Report.*

Answer Extraction in Technical Domains

Fabio Rinaldi[1], Michael Hess[1], Diego Mollá[2], Rolf Schwitter[2], James Dowdall[1], Gerold Schneider[1], and Rachel Fournier[1]

[1] University of Zurich, Department of Computer Science,
Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
{rinaldi,hess,dowdall,gschneid,fournier}@ifi.unizh.ch
[2] Department of Computing, Macquarie University,
Sydney, Australia
{diego,rolfs}@ics.mq.edu.au

Abstract. In recent years, the information overload caused by the new media has made the shortcomings of traditional Information Retrieval increasingly evident. Practical needs of industry, government organizations and individual users alike push the research community towards systems that can exactly pinpoint those parts of documents that contain the information requested, rather than return a set of relevant documents. Answer Extraction (AE) systems aim to satisfy this need. In this article we discuss the problems faced in AE and present one such system.

1 Introduction

Traditional Information Retrieval (IR) techniques provide a very useful solution to a classical type of *information need*, which can be described with the scenario of "Essay Writing". The user needs to find some information and backup material on a particular topic, and she will sift through a number of documents returned by the IR system. This assumes that the user has sufficient time to elaborate and extract the relevant information from a number of documents[1]. However, a different type of information need is becoming increasingly more common, namely one where the user has to solve a specific problem in a technical domain, which requires finding precise information of a limited size. This could be called a "Problem Solving" scenario. A very fitting example is that of technical manuals. Imagine the situation of an airplane maintenance technician who needs to operate on a defective component which is preventing an airplane from starting. He needs to swiftly locate in the maintenance manual the specific procedure to replace that component. What users really need in this situation are systems capable of analyzing a question (phrased in Natural Language) and searching for a precise answer in document collections.

[1] It has been often observed that traditional Information Retrieval should rather be called "Document Retrieval".

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 360–369, 2002.

In this paper we discuss the general problem of Question Answering (QA) and focus on a simpler task, Answer Extraction (AE), as a building block towards the more ambitious goal of QA. We also briefly describe an AE system that is used to solve a real world problem. In section 2 we compare QA, as defined in the TREC competitions [26,28], with AE. Section 3 presents the ExtrAns system whereas section 4 evaluates it. In section 5 we survey the related work.

2 Question Answering and Answer Extraction

There are different levels of performance that can be expected from a Question Answering system, and a classification is not easy. However, a first broad distinction can be made on the basis of the type of knowledge that the system employs, which ultimately determines which questions the system can answer.

An ideal system would return a grammatically well-formed surface string generated from a non-linguistic knowledge base in response to a natural language query. Unfortunately, many problems in the Knowledge Representation field are still to be solved and a comprehensive repository of world knowledge is not available[2]. What is achievable are systems that acquire their knowledge only from the target data (the documents to be queried). Such a system may allow inferences at the local/linguistic level or across multiple or single texts, depending on the task at hand. Systems employing only knowledge found explicitly in the documents should be called, in our opinion, “Answer Extraction Systems” whereas the term “Question Answering” should be reserved for systems making use of wider inferential capabilities.

The complexity of an Answer Extraction system could be defined in terms of the kind of transformations that it allows over the user query. The most simple approach would be to allow only syntactic variants (such as active/passive), while more sophisticated approaches would gradually include detection of synonyms and of more complex lexical relations among words such as thesaurus relationships like *“subdirectory* ***is a subtype of*** *directory”* as well as textual references (pronouns, definite noun phrases), and finally the use of meaning postulates (such as *“if something is* ***installed*** *in some place, then it* ***is*** *there”*).

The focus of the TREC competitions has been predominantly factual (non-generic, extensional) questions about events, geography and history, such as *“When was Yemen reunified?”* or *“Who is the president of Ghana?”*. It has been observed repeatedly that many such questions would better be directed at encyclopedias rather than at newspaper articles. Questions concerning rule-like or definitional knowledge (generic, intensional questions), such as *“How do you stop a Diesel engine?”* or *“What is a typhoon?”* have received less attention[3]. As technical documents consist almost exclusively of generic statements it is this type of question on which we have focused our attention.

[2] Despite some commendable efforts in this direction [17].

[3] Although a small number of them were included in the QA track of TREC-9 and TREC-10.

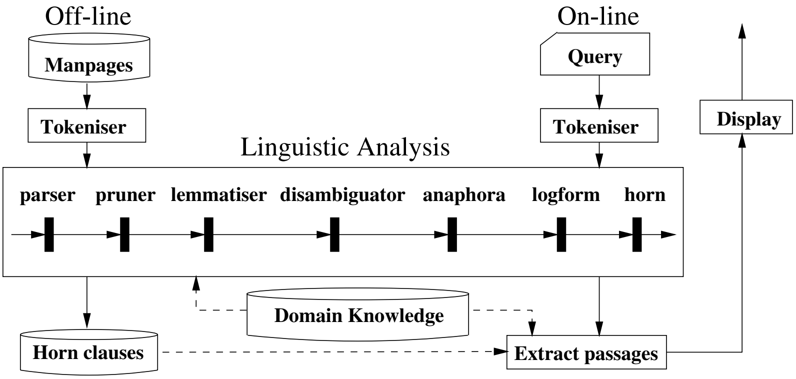


Fig. 1. Architecture of the ExtrAns system

3 A Brief Presentation of ExtrAns

Over the past few years our research group has developed an Answer Extraction system (ExtrAns) that works by transforming documents and queries into a semantic representation called Minimal Logical Form (MLF) [21] and derives the answers by logical proof from the documents. A full linguistic (syntactic and semantic) analysis, complete with lexical alternations (synonyms and hyponyms) is performed. While documents are processed in an off-line stage, the query is processed on-line (see Fig. 1).

Two real world applications have so far been implemented with the same underlying technology. The original ExtrAns system is used to extract answers to arbitrary user queries over the Unix documentation files ("man pages"). A set of 500+ unedited man pages has been used for this application. An on-line demo of ExtrAns can be found at the project web page[4].

More recently we tackled a different domain, the Airplane Maintenance Manuals (AMM) of the Airbus A320, which offered the additional challenges of an SGML-based format and a much larger size (120MB)[5]. Despite being developed initially for a specific domain, ExtrAns has demonstrated a high level of domain independence.

As we work on relatively small volumes of data we can afford to process (in an off-line stage) all the documents in our collection rather than just a few selected paragraphs. Clearly in some situations (e.g. processing incoming news) such an approach might not be feasible and paragraph indexing techniques would need to be used. At the moment we have a preselection mechanism which is based on a loose matching of question concepts against the stored semantic representations of the documents. Our current approach is particularly targeted to small and medium sized collections. For larger collections an initial preselection module would be unavoidable.

[4] `http://www.ifi.unizh.ch/cl/extrans/`

[5] Still considerably smaller than the size of the document collections used for TREC.

In the present section we will briefly describe the ExtrAns system and provide examples from the two applications. Further details can be found in [20,21,22].

3.1 Lexical and Syntactic Analysis

The document sentences (and user queries) are syntactically processed with the Link Grammar (LG) parser [25] which uses a dependency-based grammar. A corpus-based approach [3] is used to deal with ambiguities that cannot be solved with syntactic information only, in particular attachments of prepositional phrases, gerunds and infinitive constructions.

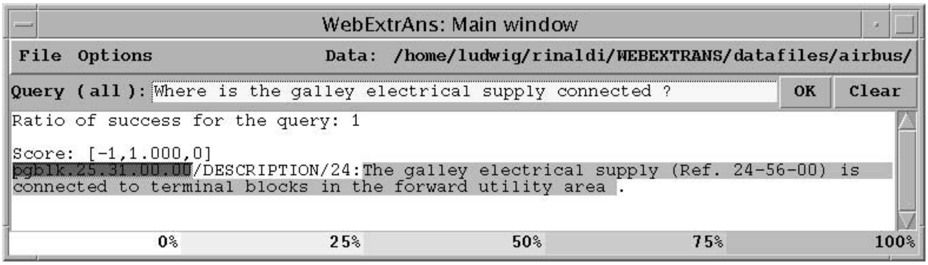
ExtrAns adopts an anaphora resolution algorithm [16] that was originally applied to the syntactic structures generated by McCord's Slot Grammar [18]. So far the resolution is restricted to sentence-internal pronouns but the same algorithm can be applied to sentence-external pronouns too.

A small lexicon of nominalizations is used for the most important cases. The main problem here is that the semantic relationship between the base words (mostly, but not exclusively, verbs) and the derived words (mostly, but not exclusively, nouns) is not sufficiently systematic to allow a derivation lexicon to be compiled automatically. Only in relatively rare cases is the relationship as simple as with "to edit <a text>" ↔ "editor of <a text>"/"<text> editor", as the effort that went into building resources such as NOMLEX [19] also shows.

Recently, we have integrated a new module which is capable of identifying previously detected multi-word domain-specific terminology (stored in a separate external DB) and processing them as single syntactical units. One of the positive effects is that the complexity of parsing the manual is considerably reduced (in some instances by as much as 50%).

User queries are processed on-line and converted into MLFs (possibly expanded by synonyms) and proved by refutation over the document knowledge base. Pointers to the original text attached to the retrieved logical forms allow the system to identify and highlight those words in the retrieved sentence that contribute most to that particular answer [22]. An example of the output of ExtrAns can be seen in Fig. 2. When the user clicks on one of the answers provided, the corresponding document will be displayed with the relevant passages highlighted.

When no direct proof for the user query is found, the system is capable of relaxing the proof criteria in a stepwise manner. First, hyponyms of the query terms will be added, thus making it more general but still logically correct. If that fails, the system will attempt approximate matching, in which the sentence with the highest overlap of predicates with the query is retrieved. The (partially) matching sentences are scored and the best fits are returned. In the case that even this method does not find sufficient answers the system will attempt keyword matching, in which syntactic criteria are abandoned and only information about word classes is used. This last step corresponds approximately to a traditional passage-retrieval methodology with consideration of the POS tags. It is important to note that, in the strict mode, the system finds only logically



```
Where is the galley electrical supply connected ?
```

```
Ratio of success for the query: 1

Score: [-1,1.000,0]
pgblk.25.31.00.00/DESCRIPTION/24:The galley electrical supply (Ref. 24-56-00) is
connected to terminal blocks in the forward utility area .
```

Fig. 2. An example of the output of ExtrAns

correct proofs (within the limits of what MLFs can represent; see below), i.e. it is a high precision AE system.

3.2 Semantic Analysis

The meaning of the documents and of the queries produced by ExtrAns is expressed by means of Minimal Logical Forms (MLFs) [24]. The MLFs are designed so that they can be found for *any* sentence (using robust approaches to treat very complex or ungrammatical sentences), and they are optimized for NLP tasks that involve the semantic comparison of sentences, such as AE.

The main feature of the MLFs is the use of reification to achieve flat expressions. As opposed to Hobb's ontologically promiscuous semantics [12], where every predicate is reified, for the time being we apply reification to a very limited number of types of predicates, in particular to objects, eventualities (events or states), and properties[6]. That way we can represent event modifiers, negations, higher order verbs, conditionals, and higher order predicates.

The expressivity of the MLFs is minimal in the sense that the main syntactic dependencies between the words are used to express verb-argument relations, and modifier and adjunct relations. However, complex quantification, tense and aspect, temporal relations, plurality, and modality are not expressed. One of the effects of this kind of underspecification is that several natural language queries, although slightly different in meaning, produce the same logical form.

The MLFs are expressed as conjunctions of predicates with all the variables existentially bound with wide scope. For example, the MLF of the sentence *"cp will quickly copy the files"* is:

(1) holds(e4), object(cp,o1,x1), object(s_command,o2,x1), evt(s_copy,e4,[x1,x6]), object(s_file,o3,x6), prop(quickly,p3,e4).

In other words, there is an entity $x1$ which represents an object of type *cp* and of type *command*, there is an entity $x6$ (a file), there is an entity $e4$, which represents a copying event where the first argument is $x1$ and the second

[6] Another related approach is that taken in Minimal Recursion Semantics [6].

argument is $x6$, there is an entity $p3$ which states that $e4$ is done quickly, and the event $e4$, that is, the copying, holds. The entities $o1$, $o2$, $o3$, $e4$, and $p3$ are the result of reification. The reification of the event, $e4$, has been used to express that the event is done quickly. The other entities are not used in this MLF, but other more complex sentences may need to refer to the reification of objects (non-intersective adjectives) or properties (adjective-modifying adverbs).

ExtrAns' domain knowledge determines that *cp* is a command name, and the words defined in the thesaurus will be replaced with their synset code (here represented as s_command, s_copy, and s_file). We have developed a small domain-specific thesaurus based on the same format as WordNet [7].

The MLFs are derived from the syntactic information produced by Link Grammar (LG) [25]. The methodology to produce the MLFs is relatively simple, one only needs to follow the main dependencies produced by the LG. However, as has been said elsewhere [21], the internal complexities of the dependency structures produced by the LG must be taken into account when producing the MLFs. The LG has a robust component that makes it possible to return structures even if the sentences are too complex or ungrammatical. The resulting structures can still be processed by ExtrAns and the corresponding MLFs are produced, possibly extended with special predicates that mark the unprocessed words as "keywords".

ExtrAns finds the answers to the questions by forming the MLFs of the questions and then running Prolog's default resolution mechanism to find those MLFs that can prove the question. Thus, the logical form of the question *"which command can duplicate files?"* is:

(2) object(s_command,O1,X1), evt(s_copy,E1,[X1,X2]), object(s_file,O2,X2)

The variables introduced in a question MLF are converted into Prolog variables. The resulting MLF can be run as a Prolog query that will succeed provided that the MLF of the sentence *"cp will quickly copy the files"* has been asserted. A sentence identifier and a pointer (indicating the tokens from which the predicate has been derived) are attached to each predicate of a MLF in the knowledge base. This information matches against additional variables attached to the predicates in the question (not shown in the example above) and is eventually used to highlight the answer in the context of the document (see Fig. 2). The use of Prolog resolution will find the answers that can logically prove the question, but given that the MLFs are simplified logical forms converted into flat structures, ExtrAns will find sentences that, logically speaking, are not exact answers but are still relevant to the user's question, such as: *"cp copies files"*, *"cp does not copy a file onto itself"*, *"if the user types y, then cp copies files"*.

In our view MLFs open up a potential path to a stepwise development of a question answering system by allowing monotonically incremental refinements of the representation without the need to destruct previous partial information [24]. While MLFs specify the core meaning of sentences they leave underspecified those aspects of semantics that are less relevant or too hard to analyse, for the time being.

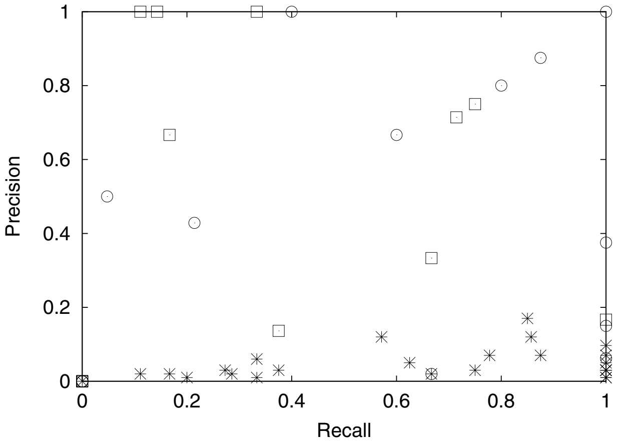


Fig. 3. Recall against precision for 30 queries and the top 100 hits per query. Prise's results are displayed with a star ($*$), and ExtrAns' results with circles ($\odot$) for the default search and with squares ($\boxdot$) for the approximate matching.

4 Evaluation

We conducted two different kinds of evaluation, one designed to compare the original ExtrAns system against a standard IR system, and one designed to give us a feeling for the portability of ExtrAns to a new domain. For the initial evaluation we used a set of 30 queries over 500 manual pages. The system chosen for the comparison was Prise, a system developed by NIST [10]. Since Prise returns full documents, we used ExtrAns' tokenizer to find the sentence boundaries and to create independent documents, one per sentence in the manual pages. Then Prise was run with our set of queries, which lead to an average of 908 hits per query. The set of all correct answers was compiled mainly by hand. As Prise provides a ranked output, in order to compute precision and recall one has to select a cut-off value (n). The combined plot of pairs computed for each n did not show significant differences with the plot for $n = 100$: the values for ExtrAns were nearly the same, and for Prise, the number of recall and precision pairs increased but the area with the highest density of points remains the same. We will therefore concentrate on the plot for $n = 100$.

Fig. 3 shows that precision is in general higher for ExtrAns than for Prise, and that Prise has better recall values. In the upper right corner, we can see a higher density of ExtrAns' values which is likely to shift to the left if we use a less restricted set of queries. The fact that ExtrAns never stopped at the hyponym and keyword search is also related to the actual query set. If the queries were more complex, we would have some recall and precision pairs corresponding to the keyword search, and this would probably cause a lower overall precision.

When we started to work on a new domain we designed a simple evaluation framework, to test the domain independence of the system. Our porting to the domain of Airbus Maintenance manuals did not involve modification of any linguistic component, but simply of the I/O interfaces of the system and the development of a new tokenizer capable to deal with SGML/XML markup. We selected semi-randomly 100 sentences from the data documents and prepared 100 questions to which those sentences could be an answer. When the port was complete we tested the questions and we obtained the expected answer on 84% of them, in another 9% of cases we obtained a correct answer, different from the one we expected, in 7% of cases we did not obtain a correct answer. We are now in the process of analyzing these results. We are also planning a similar type of evaluation with questions directly formulated by the potential users of the system (which might not have a straightforward answer in the manual).

5 Related Work

IR techniques can be used to implement QA/AE systems, by applying them at the passage or sentence level. Portions of text with the maximum overlap of question terms contain, with a certain probability, an answer. The relevance of the passages is almost invariably determined on the basis of the weights assigned to individual terms, and these weights are computed from term frequencies in the documents (or passages) and in the entire document collection (the *tf/idf* measure). Since this measure is blind to syntactic (and hence semantic) relationships it does not distinguish between hits that are logically correct and others that are purely coincidental. "Bag of words" approaches will never be able to distinguish different strings that contain the same words in different syntactic configurations, such as "absence of evidence" and "evidence of absence".

Results from the two first TREC Question Answering Tracks [26,28] showed clearly that traditional IR techniques are not sufficient for satisfactory Answer Extraction. When the answer is restricted to a very small window of text (50 bytes), systems that relied only on those techniques fared significantly worse than systems that employed some kind of language processing.

More successful approaches employ special treatment for some terms [8] (e.g. named entity recognition [14]) or a taxonomy of questions [13]. There appears to be some convergence towards a common architecture which is based on four core components [1,23]. Passage Retrieval [4] is used to identify paragraphs (or text windows) that show similarity to the question (according to some system specific metric), a Question Classification module is used to detect possible answer types [11], an Entity Extraction module [15] analyzes the passages and extracts all the entities that are potential answers and finally a Scoring module [2] ranks these entities against the question type, thus leading to the selection of the answer(s).

The systems that obtained the best results in the QA track of TREC have gradually moved into NLP techniques, such as semantics and logical forms. Falcon [9] (the best performing system in TREC-9) performs a complete analysis of a set of preselected paragraphs for each query and of the query itself and creates, after several intermediate steps, a logical representation inspired by the notation proposed by Hobbs [12]. Another similarity between ExtrAns and Falcon is that both build a semantic form starting from a dependency-based representation of the questions, although the syntactic analysis in Falcon is based on a statistical parser [5] while we use a dependency parser. As for the type of inferencing used, while ExtrAns uses standard deduction (proving questions over documents), Falcon uses an abductive backchaining mechanism to exclude erroneous answers.

6 Conclusion

In this article we have proposed as a first step towards Question Answering a more restricted kind of task for which we suggest the use of the term "Answer Extraction". We have described the differences between QA and AE and have presented an example of an AE system.

If Answer Extraction is to perform satisfactorily in technical domains over limited amounts of textual data with very little redundancy it must make maximal use of the information contained in the documents. This means that the meaning of both queries and documents must be taken into account, by syntactic and semantic analysis. Our fully functioning AE system, ExtrAns, shows that such applications are within the reach of present-day technology.

References

1. Steven Abney, Michael Collins, and Amit Singhal. Answer extraction. In Sergei Nirenburg, editor, *Proc. 6th Applied Natural Language Processing Conference*, pages 296–301, Seattle, WA, 2000. Morgan Kaufmann.
2. Eric Breck, John Burger, Lisa Ferro, Warren Greiff, Marc Light, Inderjeet Mani, and Jason Rennie. Another sys called qanda. In Voorhees and Harman [28].
3. Eric Brill and Philip Resnik. A rule-based approach to prepositional phrase attachment disambiguation. In *Proc. COLING '94*, volume 2, pages 998–1004, Kyoto, Japan, 1994.
4. C.L.A. Clarke, G.V. Cormack, D.I.E. Kisman, and T.R. Lynam. Question answering by passage selection (MultiText experiments for TREC-9). In Voorhees and Harman [28].
5. Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34st Annual Meeting of the Association for Computational Linguistics, ACL-96*, pages 184–191, 1996.
6. Ann Copestake, Dan Flickinger, and Ivan A. Sag. Minimal recursion semantics: an introduction. Technical report, CSLI, Stanford University, Stanford, CA, 1997.
7. Christiane Fellbaum. Wordnet: Introduction. In Christiane Fellbaum, editor, *WordNet: an electronic lexical database*, Language, Speech, and Communication, pages 1–19. MIT Press, Cambrige, MA, 1998.
8. Olivier Ferrett, Brigitte Grau, Martine Hurault-Plantet, and Gabriel Illouz. Qualc - the question-answering system of limsi-cnrs. In Voorhees and Harman [28].
9. Sanda Harabagiu, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Gîrju, Vasile Rus, and Paul Morarescu. FALCON: Boosting knowledge for answer engines. In Voorhees and Harman [28].

10. Donna K. Harman and Gerald T. Candela. A very fast prototype retrieval using statistical ranking. *SIGIR Forum*, 23(3/4):100–110, 1989.
11. Ulf Hermjakob. Parsing and question classification for question answering. In *Proc. of the ACL'01 workshop "Open-Domain Question Answering"*, pages 17–22, 2001.
12. Jerry R. Hobbs. Ontological promiscuity. In *Proc. ACL'85*, pages 61–69. University of Chicago, Association for Computational Linguistics, 1985.
13. Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. In Voorhees and Harman [28].
14. Kevin Humphreys, Robert Gaizauskas, Mark Hepple, and Mark Sanderson. University of Sheffield TREC-8 Q&A System. In Voorhees and Harman [27].
15. Harksoo Kim, Kyungsun Kim, Gary Geunbae Lee, and Jungyun Seo. Maya: A fast question-answering system based on a predicate answer indexer. In *Proc. of the ACL'01 workshop "Open-Domain Question Answering"*, pages 9–16, 2001.
16. Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.
17. D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 11, 1995.
18. Michael McCord, Arendse Bernth, Shalom Lappin, and Wlodek Zadrozny. Natural language processing within a slot grammar framework. *International Journal on Artificial Intelligence Tools*, 1(2):229–277, 1992.
19. Adam Meyers, Catherine Macleod, Roman Yangarber, Ralph Grishman, Leslie Barrett, and Ruth Reeves. Using NOMLEX to produce nominalization patterns for information extraction. In *Proceedings: the Computational Treatment of Nominals, Montreal, Canada, (Coling-ACL98 workshop)*, August 1998.
20. Diego Mollá. ExtrAns: An answer extraction system for unix manpages – on-line manual. Technical report, Computational Linguistics, University of Zurich, 1998. `http://www.ifi.unizh.ch/CL/`.
21. Diego Mollá, Gerold Schneider, Rolf Schwitter, and Michael Hess. Answer Extraction using a Dependency Grammar in ExtrAns. *Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammar*, 41(1):127–156, 2000.
22. Diego Mollá, Rolf Schwitter, Michael Hess, and Rachel Fournier. Extrans, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*, 2000.
23. Marius Pasca and Sanda Harabagiu. Answer mining from on-line documents. In *Proc. of the ACL'01 workshop "Open-Domain Question Answering"*, pages 38–45, 2001.
24. Gerold Schneider, Diego Mollá Aliod, and Michael Hess. Inkrementelle minimale logische formen für die antwortextraktion. 1999. Proceedings of 4th Linguistic Colloquium, University of Mainz, FASK, September 7-10, 1999.
25. Daniel D. Sleator and Davy Temperley. Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292, 1993.
26. Ellen M. Voorhees. The TREC-8 Question Answering Track Report. In Voorhees and Harman [27].
27. Ellen M. Voorhees and Donna Harman, editors. *The Eighth Text REtrieval Conference (TREC-8)*. NIST, 2000.
28. Ellen M. Voorhees and Donna Harman, editors. *Proceedings of the Ninth Text REtrieval Conference (TREC-9), Gaithersburg, Maryland, November 13-16, 2000. Preliminary on-line version: http://trec.nist.gov/pubs/trec9/t9_proceedings.html*, 2001.

Automatic Extraction of Non-standard Lexical Data for a Metalinguistic Information Database

Carlos Rodríguez

Pompeu Fabra University, Barcelona
crodri@jazzfree.com

Abstract. A modular Information Extraction system is proposed to exploit special domain texts by processing metalinguistic segments where information about the rules or units of a technical sublanguage is put forward. Final output results in a Metalinguistic Information Database with computationally tractable data that can be useful not only for lexicographers and terminologists, but also for AI systems that need unorthodox information not readily available in existing semantic networks, lexicons or traditional ontologies.

Traditional entries of lexical resources like dictionaries and glossaries contain high level, default information on word usage, and by their very nature miss a lot of the more specific, context-dependent, linguistic knowledge accessible through specialized texts in which terms are being proposed, discussed, defined, modified, or evaluated within the complex dynamics of an community of experts. Such relevant information is made prominent in discourse by various cognitive means, precisely because it cannot be presupposed to be available solely based on linguistic or domain competence. We have called *Explicit Metalinguistic Operations* (or EMOs) those textual segments in which this knowledge is negotiated. This last sentence is one instance of an EMO, as it has served to introduce this very term into our common knowledge space.

I have also proposed in previous publications [1] a tripartite nature for these discourse operations that is reflected in the following examples taken from our corpus.[1] For the sake of clarity in this discussion, I present those examples in a table where the first full length row contains a complete EMO, the first column in the following row shows the lexical item(s) figuring in it as *autonyms* [2] (or self-referential elements).[2] The next column contains the lexical, pragmatic or paralinguistic elements that help flag and articulate these discourse operations, while the last column presents the actual informative segments where something is stated about the lexical item.

To exploit such information embedded in text, we have conceived a modular Information Extraction system (**MOP**, for *Metalinguistic Operation Processor*) that

[1] A 100,000-word corpus of sociological research articles and a 10,000-sentence subcorpus extracted from highly specialized written portions of the British National Corpus.

[2] That is, those elements that, regardless of their original grammatical category, appear as names for themselves and as the logical subjects about whom the information is being provided [2].

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 370–372, 2002.

extracts candidate EMOs from running text, then parses and interprets them after standard corpus preprocessing techniques. To identify ***terms*** and ***marker/operator*** elements, MOP then applies to those sentences a set of hand-coded heuristics derived from analysis and markup of our Corpus. The data, demo and other related material can be found at http://iling.iingen.unam.mx/MOP.

Table 1.

<table>
<tr><td colspan="3">• This means that they ingest oxygen from the air via fine hollow tubes, known as tracheae.</td></tr>
<tr><th>Term</th><th>Markers/Operators</th><th>Informational segments</th></tr>
<tr><td>Tracheae</td><td>known as | <i>Apposition</i></td><td>fine hollow tubes</td></tr>
<tr><td colspan="3">• Computational Linguistics could be defined as the study of computer systems for understanding, generating and processing natural language [Grishman, 1986].</td></tr>
<tr><th>Term</th><th>Markers/Operators</th><th>Informational segments</th></tr>
<tr><td>Computational Linguistics</td><td>defined as | <i>Caps</i></td><td>the study of computer systems for understanding, generating and processing natural language [Grishman, 1986].</td></tr>
<tr><td colspan="3">• In 1965 the term soliton was coined to describe waves with this remarkable behaviour.</td></tr>
<tr><th>Term</th><th>Markers/Operators</th><th>Informational segments</th></tr>
<tr><td>Soliton</td><td>coined | the term</td><td>to describe waves with this remarkable behaviour</td></tr>
</table>

The algorithms implemented in Python identify informative segments and obtain non-standard lexical data by using a final Predicate Processing Module that populates a METALINGUISTIC INFORMATION DATABASE (MID). Predicate processing is done through partial (or chunk) parsing and a series of hand-coded rules that hopefully will be superseded in the future by applying machine-learning techniques to leverage empirical data with a fairly large training Corpus. In order to enhance the Recall and Precision of manually-coded systems, this kind of data-driven fine-tuning has been shown [3,4] to be the best way to improve both the portability and the general performance of Information Extraction systems created for well defined domains or subjects, like those tested at the series of Message Understanding Conferences sponsored by DARPA in the Nineties.

Though most Information Extraction systems are limited in practice to a single subject, to a single domain, because our specific "domain" of language usage and terminological conventions is present in all areas of knowledge and in all disciplines[3] metalinguistic information extraction has the potential of being applied to any kind of technical text.

[3] Even sharing some features across languages due to the international scientific community and the way terminology is controlled and transmitted

The MID will efficiently store and make lexical and pragmatic information available to update terminological Knowledge Bases or machine-readable dictionaries, but can also be used by AI systems that need unorthodox information (not readily found in semantic networks, lexicons or traditional ontologies) to drive inferences or disambiguate.

An MID can be conceptualized as a veritable *anti-dictionary*, since it contains exceptions, special contexts, specific usages: instances where meaning, value or contextual conditions have been spotlighted for epistemic and cognitive reasons. A MID with computationally tractable data can either override or enrich the default information of a lexical database. Its role would not be to replace, but to complement terminological Knowledge Bases or computational lexicons. A small prototype of a MID can be viewed at the project's URL.

We have implemented the Database using XML standards and resources to ensure transparency, portability and accessibility across platforms and applications. XML is flexible enough to transfer the responsibility for processing data to querying applications, instead of forcing some kind of interpretation by its very nature or structure. In that sense, a database that encourages further processing lies in between the raw possibilities of pure corpus text, and the (sometimes excessively) structured data of traditional lexical resources that are anchored in fixed theoretical frameworks.

The real challenge facing this research lies not in retrieving EMOs from text to populate a MID, but the successful formalization of heterogeneous linguistic information into a robust and manageable data structure. This objective might require redefinition, within this context, of such notions as *lexical meaning, semantic content, sense restriction, contextual conditions, community consensus,* etc. An effective and efficient computational representation of such diverse information is not trivial. A *Metalinguistic Information Database* must integrate the best features from diverse (and perhaps conflicting) lexical representation systems.

References

1. Rodriguez, C.: "Extraction of knowledge about terms from metalinguistic activity in texts". In: A. Gelbukh (ed.): *Proceedings of the Conference on Intelligent text processing and Computational Linguistics, CICLING-2000.* Instituto Politécnico Nacional, Mexico City, Mexico. (2000)
2. "Explicit Metalinguistic Operations in specialized discourse: The construction of lexical meaning in theoretic science". In: P. Sandrini (ed.): *Terminology and Knowledge Engineering TKE'99*, Innsbruck, Austria. (1999)
3. Rey-Debove, J.: Le Métalangage, Le Robert, Paris (1978)
4. Lehnert, W. C. Cardie, D. Fisher, J. McCarthy, E. Riloff, & S. Soderland.: Evaluating an Information Extraction System. Journal of Integrated Computer-Aided Engineering. 1(6) (1994)
5. Riloff, E.: Automatically Generating Extraction Patterns from Untagged Text. Proc. 13th. National Conference on Artificial Intelligence, p.1044-1049 (1996)

Text Segmentation for Efficient Information Retrieval[*]

Fernando Llopis, Antonio Ferrández, and José Luis Vicedo

Departamento de Lenguajes y Sistemas Informáticos
University of Alicante
Alicante, Spain
{llopis,antonio,vicedo}@dlsi.ua.es

Abstract. Previous works in Information Retrieval show that using pieces of text obtain better results than using the whole document as the basic unit to compare with the user's query. This kind of IR systems is usually called *Passage Retrieval (PR)*. However, there is not a general agreement about how one should define those pieces of text (also known as *passages*), in order to obtain an optimum performance. This paper proposes a PR system based on a novel selection of variable size passages. It presents an evaluation that shows better results than a standard IR system and several well-known PR systems.

1 Introduction

Information Retrieval (IR) systems are defined as tools capable of extracting a ranked list of relevant documents for a user's query. These systems are based on measuring the similarity between each document and the query, by means of several formulas that typically use the frequency of query terms in the documents. This way of measuring causes that bigger documents could have more chances to be considered relevant, because of its higher number of terms that could coincide with those of the query.

In order to solve this problem, some IR systems measure the similarity in accordance with the relevance of the pieces of adjoining text that form the documents, where these pieces of text are called passages. This kind of IR systems, which are usually called Passage Retrieval (PR), allows that the similarity measure is not affected by the size of the document. Moreover, PR systems obtain better accuracy than IR systems, and they also return the precise piece of text where it is supposed to find the answer to the query, a fact that is especially important when big documents are returned.

PR systems are more complex than IR, since the number of textual units to compare is higher (each document is formed by several passages) and the number of modules is higher (above all when the passage splitting is accomplished after processing

[*] This paper has been partially supported by the Spanish Government (CICYT) project number TIC2000-0664-C02-02.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 373–380, 2002.

each query as it is proposed in 4). Nevertheless, PR better results are higher than complexity that adds these systems. For example, in 1 the improvement reaches a 20%, and in 4 it does a 50%.

The PR system presented in this paper is called IR-n. It defines a novel passage selection model, which forms the passages from sentences in the document. IR-n has been used in the last Cross-Language Evaluation Forum (CLEF-2001) and in the last Text REtrieval Conference (TREC-2001) in the Question Answering track. Moreover, it has been compared with two standard IR systems, the first one based on cosine similarity measure by Salton 8, and the second one the well-known IR system called Z/Prise 12. The test has been accomplished on the same set of documents and questions. Furthermore, our system is compared with other PR systems.

The following section presents the backgrounds in PR. Section 3 describes the architecture of our proposed PR system. In section 4, we give a detailed account of the test, experiments and obtained results. Finally, we present the conclusions of this work.

2 Backgrounds in Passage Retrieval

The most frequent similarity measures between documents and queries are cosine 8, the pivoted cosine 10 and the okapi system 11. These models are mainly based on counting the number of terms that documents and queries are sharing and applying a normalization process.

The main differences between different PR systems are the way that they select the passages, that is to say, what they consider as a passage and the size of them. According to the taxonomy proposed in 1, the following PR systems can be found: discourse-based model, semantic model and window model. The first one uses the structural properties of the documents, such as sentences or paragraphs (e.g. the one proposed in 7, 9) in order to define the passages. The second one divides each document in semantic pieces, according to the different topics in the document (e.g. those in 2). The last one uses windows with a fixed size to form the passages 1, 3.

On the one hand, it looks coherent that discourse-based models are more effective since they are using the structure of the document itself. However, the greater problem of them is that the results could depend on the writing style of the document author. On the other hand, window models have the main advantage that they are simpler to accomplish, since the passages have a previously known size, whereas the remaining models have to bear in mind the variable size of each passage. Nevertheless, discourse-based and semantic models have the main advantage that they return logic and coherent fragments of the document, which is quite important if these IR systems are used for other applications such as Question Answering. Finally, it should be mentioned that semantic and window models can partially or full overlap pieces of text can overlap pieces of text in order to fine tune.

The passage extraction model that we are proposing allows us to benefit from the advantages from discourse-based models, since logic information units of the text, such as sentences, form the passages. Moreover, another novel proposal in our PR system is the relevance measure, which unlike other discourse-based models is not calculated from the number of passage terms, but the fixed number of passage sentences. It allows a simpler calculation of this measure unlike other discourse-based or semantic models. Although we are using a fixed number of sentences for each passage, we consider that our proposal differs from the window models since our passages does not have a fixed size (i.e. a fixed number of words) because they are using sentences with a variable size.

3 Overview of the System

In this section, we are briefly describing the architecture of the proposed PR system, namely IR-n. We are focusing on its two main modules: the indexation and the document-extracting module.

3.1 Indexation Module

The main aim of this module is to generate the dictionaries that contain all the required information for the document-extracting module. It requires the following information for each term:

- The number of documents that contain the term.
- For each document:
  - The number of times that the term appears in the document.
  - Position of the term in the document: the number of sentence and position in the sentence.

Where we are considering as terms, the stems produced by the Porter stemmer on those words that do not appear in a list of stop-words, list that is similar to those used in IR systems. For the query, the terms are also extracted in the same way, that is to say, their stems and positions in the query for each query word that does not appear in the list of stop-words. In principle, it supposes that we need more information than for standard IR systems. But, as it will be shown with the results, the benefits exceed this storage increase.

3.2 Document-Extracting Module

This module extracts the documents according to its similarity with the user's query. The scheme in this process is the following:

1. Query terms are sorted according to the number of documents in which they appear, where the terms that appear in fewer documents are processed firstly.
2. The documents that contain some query term are extracted.
3. The following similarity measure is calculated for each passage p with the query q:

$$\text{Similarity\_measure (p, q)} = \sum_{t \in p \wedge q} \mathrm{W}_{\mathrm{p,t}} * \mathrm{W}_{\mathrm{q,t}}$$

Where:

$W_{p,t} = \log_e(\mathrm{f}_{\mathrm{p,t}} + 1)$.
$f_{p,t}$ is the number of times that the term *t* appears in the passage *p*.
$W_{q,t} = \log_e(\mathrm{f}_{\mathrm{q,t}} + 1) * \mathrm{idf}$.
$f_{q,t}$ is the number of times that the term *t* appears in the query *q*.
$idf = \log_e(\mathrm{N} / \mathrm{f}_{\mathrm{t}} + 1)$.
N is the number of documents in the collection.
f_t is the number of documents that contain the term *t*.

4. Each document is assigned the highest similarity measure from its passages.
5. The documents are sorted by their similarity measure.
6. The documents are presented according to their similarity measure.

As it is noticed, the similarity measure is similar to cosine measure presented in 8. The only difference is that the size of each passage (the number of terms) is not used to normalise the results. This difference makes the calculation simpler than other discourse-based PR systems or IR systems, since the normalization is accomplished according to a fixed number of sentences per passage. Another important detail to notice is that we are using *N* as the number of documents in the collection, instead of the number of passages. That is because in 4 it is not considered relevant for the final results.

The optimum number of sentences to consider per passage is experimentally obtained. It can depend on the genre of the documents, or even on the type of the query as it is suggested in 3. We have experimentally considered a fixed number of 20 sentences for the collection of documents in which we are going to work 6. Table 1 presents the experiment where the 20 sentences per passage obtained the best results.

Table 1. Precision results obtained on *Los Angeles Times* collection with different number of sentences per passage

		Precision IR-n					
		5 Sent.	**10 Sent.**	**15 Sent.**	**20 Sent.**	**25 Sent.**	**30 Sent.**
Recall	**0.00**	0.6378	0.6508	0.6950	*0.7343*	0.6759	0.6823
	0.10	0.5253	0.5490	0.5441	*0.5516*	0.5287	0.5269
	0.20	0.4204	0.4583	0.4696	*0.4891*	0.4566	0.4431
	0.30	0.3372	0.3694	0.3848	*0.3964*	0.3522	0.3591
	0.40	0.2751	0.3017	0.2992	*0.2970*	0.2766	0.2827
	0.50	0.2564	0.2837	0.2678	*0.2633*	0.2466	0.2515
	0.60	0.1836	0.1934	0.1809	*0.1880*	0.1949	0.1882
	0.70	0.1496	0.1597	0.1517	*0.1498*	0.1517	0.1517
	0.80	0.1213	0.1201	0.1218	*0.1254*	0.1229	0.1279
	0.90	0.0844	0.0878	0.0909	*0.0880*	0.0874	0.0904
	1.00	0.0728	0.0722	0.0785	*0.0755*	0.0721	0.0711

As it is commented, the proposed PR system can be classified into discourse-based models since it is using variable-sized passages that are based on a fixed number of sentences (but different number of terms per passage). The passages are overlapping each other, that is to say, let us suppose that the size of the passage is *N* sentences, then the first passage will be formed by the sentences from 1 to N, the second one from 2 to N+1, and so on. We have decided to overlap just one sentence based on the following experiment, where several numbers of overlapping sentences have been tested. In this experiment, Table 2, can be observed that only one overlapping sentence obtain the best results.

Table 2. Experiments with a different number of overlapping sentences

		IR-n with 1 overlap.	**IR-n with 5 overlap.**	**IR-n 10 overlap.**
Recall	**0.00**	*0.7729*	0.7211	0.7244
	0.10	*0.7299*	0.6707	0.6541
	0.20	*0.6770*	0.6072	0.6143
	0.30	*0.5835*	0.5173	0.5225
	0.40	*0.4832*	0.4144	0.4215
	0.50	*0.4284*	0.3704	0.3758
	0.60	*0.3115*	0.2743	0.2759
	0.70	*0.2546*	0.2252	0.2240
	0.80	*0.2176*	0.1914	0.1918
	0.90	*0.1748*	0.1504	0.1485
	1.00	*0.1046*	0.0890	0.0886
Medium		*0.4150*	0.3635	0.3648

4 Evaluation

In this section, the evaluation is presented, in which the obtained results show the improvement that introduce our proposal.

4.1 Experiments

Some experiments have been carried out to measure the improvement of our proposal. These experiments have been run on a TREC collection: *Los Angeles Times*. This collection is formed by 113.005 documents, where the medium number of words per sentence is about 29, and the medium number of terms per sentence is about 9. However it should be noticed that this collection has a heterogeneous format and size, where the longest document has 807 sentences, and the smaller just 1 sentence.

A set of 47 queries has been used for the evaluation. These queries have been previously used in the last Cross-Language Evaluation Forum (CLEF) in which the authors have participated.

The evaluation measures in this paper are those used in TREC conferences, namely recall and precision:

$$\text{Recall} = \frac{\text{Number of relevant documents extracted}}{\text{Number of relevant texts in the collection}}$$

$$\text{Precision} = \frac{\text{Number of relevant documents extracted}}{\text{Number of extracted documents}}$$

In order to compare our proposal, IR-n, with other proposals, two IR systems have been run on the same set of documents and queries. The first one is the vectorial standard model defined in 8. The second one is the Z/Prise system 12.

Although, in CLEF conference, all the 47 test queries are formulated in three different ways: *title, narrative* and *description*, in this paper, the experiments have only been accomplished on their *title* form. The title form usually have between 2 to 4 words, which is the more usual length of the queries in Internet.

4.2 Obtained Results

Table 3 presents the interpolated precision for standard levels of recall. The second column in this table presents the results obtained by the vectorial model 8, the third one does by Z/Prise 12, and the last column does for our system, IR-n.

From Table 3, it can be observed that our proposal obtains better precision results than the vectorial model for all the different passage size. For 20 sentences length and recall 0.10, the benefit reaches a 29%. With reference to Z/Prise, similar results are obtained, although it should be mentioned that Z/Prise uses query expansion, and we are using *title* form for the queries, that suppose a medium of 3 words.

Table 3. Obtained results on *Los Angeles Times* collection and 47 CLEF title-queries

	Precision		
Recall	**Vectorial model**	**Z/Prise**	**IR-n**
0.00	0.5184	0.7583	0.7729
0.10	0.4379	0.7278	0.7299
0.20	0.3862	0.6476	0.6770
0.30	0.3424	0.5632	0.5835
0.40	0.3171	0.4904	0.4832
0.50	0.2774	0.4389	0.4284
0.60	0.2201	0.3315	0.3115
0.70	0.1718	0.2825	0.2546
0.80	0.1350	0.2343	0.2176
0.90	0.1180	0.1925	0.1748
1.00	0.0878	0.1317	0.1046
Medium	0.2563	0.4208	0.4150

Table 4 presents the results obtained when 5, 10, 15, 20, 30 and 200 documents are extracted. Again, the results show that IR-n is quite superior to the vectorial model, and quite similar to Z/Prise, although in this case, our system is superior to Z/Prise when fewer documents are extracted, in spite of not using query expansion techniques as it does Z/Prise.

Table 4. Precision obtained with different number of extracted documents

		Vectorial model	Z/Prise	IR-n
Precision	**At 5 docs**	0.2638	0.4638	0.5021
	At 10 docs	0.2511	0.3872	0.4021
	At 15 docs	0.2184	0.3135	0.3319
	At 20 docs	0.2053	0.2851	0.2840
	At 30 docs	0.1716	0.2383	0.2411
	At 100 docs	0.0943	0.1215	0.1177
	At 200 docs	0.0591	0.0705	0.0691
	At 500 docs	0.0287	0.0318	0.0318
	At 1000 docs	0.0154	0.0167	0.0165
	R-Precision	0.2353	0.4009	0.3899

Kaskinoel and Zobel 3 made a comparison between different PR models and the vectorial model, whose results are presented in Table 5. In this table, the vectorial model is compared with the best PR system: those with a fixed window of 350 words per passage, and those with 250 words. Since these results have been obtained on a different set of documents and queries, the direct comparison is not possible with our IR-n system. However, the per cent differences between these PR systems and the vectorial model can be relatively compared with our differences with the same vectorial model, where the improvement in IR-n is higher with reference to these window models.

Table 5. Comparison with other PR systems

	Precision	%Difference
Results obtained in this paper		
Vectorial model	0.2563	
IR-n	0.4150	+52.16
Kaskiel and Zobel results		
Vectorial model	0.2434	
350 words per passage	0.3391	+39.31
250 words per passage	0.3432	+41.00

5 Conclusions and Future Work

In this paper, a novel passage extraction model has been presented. This model can be included in the discourse-based models since it is using the sentences as the logical unit to divide the document into passages. The passages are formed by a fixed number of sentences, which does not mean that it could be included in the window models, since our passages does not have a fixed number of words. In this paper, a similarity measure is also proposed, which allows us to calculate the similarity between documents and queries in a simpler way than other discourse-based models. Finally, the proposed system, namely IR-n, has been evaluated and compared with other IR and PR-systems: vectorial model, Z/Prise and window models, where IR-n obtains better precision except for the Z/Prise (the precision is similar between them), although IR-n leads Z/Prise when less documents are extracted, in spite of not using query expansion techniques as it does Z/Prise. As future works, we pretend to work in several topics about PR. For example, we intend to deeply study the influence of the passage length with reference to different set of documents and different queries. Moreover, we intend to incorporate to IR-n several query expansion techniques in order to set a real comparison with other IR systems as Z/Prise.

References

1. Callan, J. *Passage-Level Evidence in Document Retrieval*. In Proceedings of the 17 th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 1994, pp. 302-310.
2. Hearst, M. and Plaunt, C. *Subtopic structuring for full-length document access.* Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, June 1993, Pittsburgh, PA, pp 59-68
3. Kaskiel, M. and Zobel, J. *Passage Retrieval Revisited* SIGIR '97: Proceedings of the 20th Annual International ACM July, 1997, Philadelphia, PA, USA, pp 27-31
4. KaszKiel, M. and Zobel, J. *Effective Ranking with Arbitrary Passages.* Journal of the American Society for Information Science, Vol 52, No. 4, February 2001, pp 344-364.
5. Kaszkiel, M.; Zobel, J. and. Sacks-Davis, R.. *Efficient Passage Ranking for Document Databases.* ACM transactions on Information Systems, Vol 17, Nº 4, October 1999, pp 406-439
6. Llopis, F. and Vicedo, J. *Ir-n system, a passage retrieval system at CLEF 2001* Working Notes for the Clef 2001 Darmstdt, Germany , pp 115-120
7. Namba, I *Fujitsu Laboratories TREC9 Report.* Proceedings of the Tenth Text REtrieval Conference, TREC-10. Gaithersburg,USA. November 2001, pp 203-208
8. Salton G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer,* Addison Wesley Publishing, New York. 1989
9. Salton, G.; Allan, J. Buckley *Approaches to passage retrieval in full text information systems.* In R Korfhage, E Rasmussen & P Willet (Eds.) Prodeedings of the 16 th annual international ACM-SIGIR conference on research and development in information retrieval. Pittsburgh PA, pp 49-58
10. Singhal, A.; Buckley, C. and Mitra, M. *Pivoted document length normalization.* Proceedings of the 19th annual international ACM-SIGIR conference on research and development in information retrieval, 1996.
11. Venner, G. and Walker, S. *Okapi '84: 'Best match' system.* Microcomputer networking in libraries II. Vine, 48,1983, pp 22-26.
12. Zprise developed by Darrin Dimmick (NIST) Available on demand at http://itl.nist.gov./iaui/894.02/works/papers/zp2/zp2.html

Using Syntactic Dependency-Pairs Conflation to Improve Retrieval Performance in Spanish*

Jesús Vilares, Fco. Mario Barcala, and Miguel A. Alonso

Departamento de Computación, Universidade da Coruña
Campus de Elviña s/n, 15071 La Coruña, Spain
{jvilares,barcala}@mail2.udc.es, alonso@udc.es
http://coleweb.dc.fi.udc.es/

Abstract. This article presents two new approaches for term indexing which are particularly appropriate for languages with a rich lexis and morphology, such as Spanish, and need few resources to be applied. At word level, productive derivational morphology is used to conflate semantically related words. At sentence level, an approximate grammar is used to conflate syntactic and morphosyntactic variants of a given multi-word term into a common base form. Experimental results show remarkable improvements with regard to classical indexing methods.

1 Introduction

For Information Retrieval (IR) tasks, documents are frequently represented through a set of index terms or representative keywords. This can be accomplished through operations such as the elimination of *stopwords* (too frequent words or words with no apparent significance) or the use of *stemming* (which reduces distinct words to their supposed grammatical root). These operations are called *text operations*, providing a *logical view* of the processed document.

In effect, current IR systems conflate the documents before indexing to decrease their linguistic variety by grouping together textual occurrences referring to similar or identical concepts by exploiting graphical similarities, thesaurus, etc. [1,7]. However, most classical IR techniques for such tasks lack solid linguistic grounding. Even operations with an apparent linguistic basis (e.g. stemming) which obtain good results for English, perform badly when applied to languages with a very rich lexis and morphology, such as Spanish. For these languages, we must employ more and better linguistic resources with Natural Language Processing (NLP) techniques, all of which involves a greater complexity and a higher computational cost. At this point, we must face one of the main problems of NLP in Spanish, which is the lack of available resources: large tagged corpora, treebanks and advanced lexicons are not freely available.

In this context, we propose to extend classical IR techniques to avoid such obstacles.

* This research has been partially supported by Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (Grant TIC2000-0370-C02-01), FEDER of EU (Grant 1FD97-0047-C04-02) and Xunta de Galicia (Grants PGIDT99XI10502B and PGIDT01PXI10506PN).

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 381–390, 2002.

2 Single Word Term Conflation

In English, single word term conflation can be accomplished through a *stemmer* [9], a simple tool from a linguistic point of view, with a low computational cost. The results obtained are satisfactory enough since the inflectional morphology of English being very simple. The situation for Spanish is completely different, because inflectional modifications exist at multiple levels[1] with many irregularities. Therefore, we must apply NLP techniques, thus increasing the complexity and the computational cost of the system. As a first step, we have employed a lemmatizer to obtain the lemma of each word, thereby solving the problems derived from inflection in Spanish. As a second step, we have developed a new approach based on morphological families.

2.1 Morphological Families as a Text Operation

Spanish has a great productivity and flexibility in its word formation mechanisms by using a rich and complex productive morphology, preferring derivation to other mechanisms [2]. We define a *morphological family* as a set of words obtained from the same morphological root through derivation mechanisms. It is expected that a basic semantic relationship will remain between the words of a given family[2]. Regular word formation patterns in Spanish can be obtained through the 'rules of word formation' [8] defined by generative phonology and transformational-generative grammars. Though this paradigm is not complete, it can be used to implement an automatic system for generation of morphological families with an acceptable degree of completeness and correction [10].

In order to use morphological families for document conflation, the first step is to obtain the part of speech and the lemmas of the text to be indexed. Next, we replace each of the lemmas obtained by the representative of its morphological family. In this way we are using the same index term to represent all words belonging to the same morphological family; therefore, semantic relations that exist between these words remain in the index because related terms are conflated to the same index term.

We have compared the accuracy of lemmatization and morphological families as text operations with respect to the classical technique of stemming. We have studied the behaviour of different stemmers specifically designed for Spanish, and the best results we obtained were for the stemmer used by the open source search engine Muscat[3], based on Porter's algorithm [1]. However, such results were poor. The employment of a lemmatizer allowed us to reach an approximate accuracy of 96%, whereas the Muscat stemmer only reached 37% overall. Furthermore, the behaviour of a lemmatizer is uniform for all grammar categories,

[1] Gender and number for nouns and adjectives, and person, mood, time and tense for verbs.

[2] Relations of the type process-result, e.g. *producción* (production) / *producto* (product), process-agent, e.g. *manipulación* (manipulation) / *manipulador* (manipulator), etc.

[3] `http://open.muscat.com`

whereas stemmers obtain an accuracy of 46% for nouns, 36% for adjectives and 0% for verbs[4]. A noticeable extra advantage of lemmatizers in relation to stemmers is their capability to disambiguate using word context. Moreover, comparing stemmers with respect to morphological families, we find that the Muscat stemmer is able to identify 27% of the families, 95% of which are families formed by only one lemma, 3% by two lemmas, and less than 2% by three lemmas.

With regard to computational cost, morphological families and their representatives are computed *a priori*, so they do not affect the final indexing and querying cost. The running cost of a stemmer is linear in relation to the length of the word. The running cost of a lemmatizer-disambiguator is only slightly greater: linear in relation to the length of the word and cubic in relation to the size of the tagset, which is a constant. As will be detailed in Sect. 3.3, our system only needs to know the grammatical category of the word, so the tagset will be very small. Therefore, the increase in cost becomes negligible.

3 Multi-word Term Conflation

A *multi-word term* is a term containing two or more content words (nouns, verbs and adjectives) [5]. Several techniques are described in the literature to obtain them. One of the most frequently used is *text simplification* [5]: as a first step, we make a single word stemming, after which stopwords are deleted; in the final step, terms are extracted and conflated by means of pattern matching [3], statistical criteria [4], etc. As we can see, most operations lack solid linguistic grounding[6], which often results in incorrect conflations. Nevertheless, this is the easiest and least costly method. At the other extreme, we find the *morpho-syntactic analysis* of the text, which uses a parser that produces syntactic trees which denote dependency relations between involved words. As a result, structures with similar dependency relations are conflated in the same way. At the mid point, we have *syntactic pattern matching*, which is based on the hypothesis that the most informative parts of the texts correspond to specific syntactic patterns [6]. In this article we take an approach that combines these two last solutions, trying to obtain the concepts of a text by means of the syntactic relations that exist between the terms of the document. These syntactic relations will be identified through *syntactic patterns* of noun syntagmas and their *syntactic and morpho-syntactic variants.*

A syntactic or morpho-syntactic variant of a multi-word term is a textual utterance in which:

- Syntactic variants result from the inflection of individual words and from modifying the syntactic structure of the original term. E.g. *chicos gordos y altos* (fat and tall boys) is a variant of *chico gordo* (fat boy).

[4] This is due to the complexity of the verbal paradigm in Spanish, which is not treated in depth by any stemmer.

[5] E.g. *el perro grande del vecino* (the neighbour's big dog)

[6] For example, stopwords such as determiners and prepositions are key components of the syntactic structure.

- Morpho-syntactic variants differ from syntactic variants in that at least one of the content words of the original term is transformed into another word derived from the same morphological stem. E.g. *medir el contenido* (to measure the content) is a variant of *medición del contenido* (measurement of the content).
- The original term can substitute the variant in a task of information access.

¿From a morphological point of view, syntactic variants refer to inflectional morphology, whereas morpho-syntactic variants also refer to derivational morphology. In the case of syntax, syntactic variants have a very restricted scope, i.e. a noun syntagma, whereas morpho-syntactic variants can span a whole sentence, including a verb and its complements[7]. Next, we will study the mechanisms involved in obtaining syntactic and morpho-syntactic variants.

3.1 Syntactic Variants

In Spanish, syntactic variants of a multi-word term may involve variations in the inflection of its words, and syntactic alterations of the kind:

- *Coordination:* this consists of employing coordinating constructions (copulative or disjunctive) with the modifier or with the modified term. For example, *coches rojos* (red cars) and *motos rojas* (red bikes) combine into *coches y motos rojos* (red cars and bikes), which can be considered as a variant of any of the combined terms.
- *Substitution:* it consists of employing modifiers to make a term more specific. For example, *caída en las ventas* (sales drop) can be transformed into *caída anormal en las ventas* (unusual sales drop) by adding the adjective *anormal.*
- *Synapsy:* whereas the preceding constructions are binary, this is a unary construction which corresponds to a change of preposition or the addition or removal of a determiner. For example, we can obtain *abono para plantas* (fertilizer for plants) from *abono para las plantas* (fertilizer for the plants).
- *Permutation:* this refers to the permutation of words around a pivot element, for example *saco viejo* (old bag) and *viejo saco* (old bag).

3.2 Morpho-Syntactic Variants

According to the nature of the morphological transformations applied to the content words of the terms, we can classify morpho-syntactic variants into:

- *Iso-categorial:* the morphological derivation process does not change the category of the word, but only transforms one noun syntagma into another. There are two possibilities:
  1. *Noun-to-Noun:* they cover relations of the type process-result —*producción artesanal* (craft production) / *producto artesanal* (craft product)— and process-agent —*manipulación de las masas* (manipulation of the masses) / *manipulador de las masas* (manipulator of the masses)—.

[7] Let us consider *comida de perros* (dog food) and *los perros comen* (dogs feed on).

  2. *Adjective-to-Adjective:* covering relations of the type agent-result —*compuesto ionizador* (ionizer compound) / *compuesto ionizado* (ionized compound)—.

- *Hetero-categorial:* morphological derivation does result in a change of the category of the word. They are not restricted to the frontier of a noun syntagma.
  1. *Noun-to-Verb:* these variations involve semantic changes of the type process-result, e.g. *recortar gastos* (to cut back spending) / *recorte de gastos* (spending cutback).
  2. *Noun-to-Adjective:* in a noun syntagma the noun can be modified by adjectival constructions or equivalent prepositional ones, e.g. *cambio del clima* (change of the climate) / *cambio climático* (climatic change).

3.3 Term Extraction and Conflation

In information systems, many of the queries can be formulated as noun syntagmas of diverse complexity. Thus, we will take noun syntagmas as base terms from which we will obtain, through the corresponding mechanisms, their syntactic and morpho-syntactic variants, not necessarily noun syntagmas. All these multi-word terms, either the original noun syntagmas or their variants, can be used as index terms.

In Spanish, the basic structures for noun syntagmas are four: *Adj-Noun*, *Noun-Adj*, *Noun-Prep-Noun* and *Noun-Prep-Det-Noun*. So, we are interested in identifying such noun syntagmas and their variants for indexing.

To extract such index terms we will use syntactic matching patterns obtained from the syntactic structure of the noun syntagmas and their variants. For such a task we take as our basis an approximate grammar for Spanish:

$$\begin{array}{lll} S & \rightarrow NP\ \ V\ W^{?}\ (NP|PP)^{*} & (1) \\ NP & \rightarrow D^{?}\ AP^{*}\ N\ (AP|PP)^{*} & (2) \\ AP & \rightarrow W^{?}\ A & (3) \\ PP & \rightarrow P\ NP & (4) \end{array}$$

where the symbols D, A, N, W, V and P are the part of speech labels that denote determiners, adjectives, nouns, adverbs, verbs and prepositions, respectively[8]. The motivation of these rules is:

(1) shows a sentence structure of the kind *Subject-Verb-Complement.*
(2) defines a noun syntagma as a noun modified by adjectives and/or prepositional syntagmas.
(3) lets adjectives be modified by adverbs.
(4) shows a prepositional syntagma formed by a preposition and a noun syntagma.

[8] Coordinating conjunctions (C) and punctuation marks (Q) will be also used later to obtain variants.

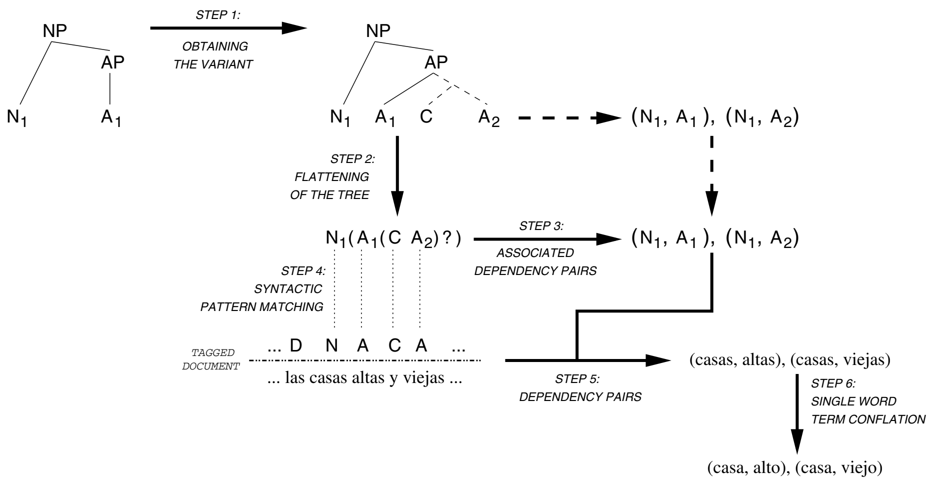


Fig. 1. Example of multi-word term conflation via dependency pairs

Other authors, such as [5], take a static approach based on the use of previous existing terminological databases, which are incorporated into a lexicalized parser. Since this kind of resources is very difficult to obtain for Spanish, we opt for a dynamic approach in which terms are dynamically identified during the indexing process without any deep syntactic processing of the document, only a surface process, this approach having no terminological reference at all. In this way, the increase of computational cost and the number of extra linguistic resources employed by the system are minimal, key questions for being employed in real-world applications.

The first task to be performed when indexing a text is to identify the index terms. Taking as our basis the syntactic trees corresponding to noun syntagmas and according to the approximate grammar we have previously shown, we manually apply the mechanisms described in Sects. 3.1 and 3.2. As a result, we obtain the syntactic trees corresponding to syntactic and morpho-syntactic variants of such noun syntagmas. This set of trees that we have obtained for multi-word terms (noun syntagmas and their variants) can be classified into four main groups: *noun modified by adjectives*, *noun modified by prepositional syntagmas*, *verb-complement* and *subject-verb*.

However, in our approach, these trees are not directly applicable to term extraction. First, they are flattened into regular expressions using the part of speech labels of the tokens involved. Let us take the example shown in Fig. 1:

1. We start with a noun syntagma whose syntactic structure is shown in the left tree, with the head noun N_1 modified by an adjectival syntagma.
2. We obtain one of its variants through the incorporation of a coordination into the adjectival syntagma (*step 1*).
3. The syntactic tree of the obtained variant is flattened to obtain the pattern which will be applied to the tagged text (*step 2*).

Table 1. Statistics of the composition of the test corpus

	source	*pln*	*lem*	*fam*	*FNL*	*FNF*
Total	9,780,513	4,526,058	4,625,579	4,625,579	2,666,190	2,666,190
Unique	154,419	154,071	111,982	105,187	1,210,182	1,036,005

Once index terms have been identified through syntactic matching patterns, they must be conflated. This process consists of two phases. Firstly, we identify syntactic dependencies between pairs of content words inside the syntactic tree of the multi-word term (*syntactic-dependency pairs*); such pairs are now associated with the matching pattern which corresponds with that tree. Secondly, single word term conflation mechanisms (lemmatization or morphological families) are applied to the words which form such pairs; the resultant pairs are the terms to be indexed.

The dependencies we can find in a multi-word term correspond to three main types:

1. *Modified-Modifier:* these kinds of relation are found in noun syntagmas. A dependency-pair is obtained for each combination of the head of the modifiers with the head of the modified terms. For example:
   chicos feos y altos → *(chico, feo),(chico, alto)*
   (ugly and tall boys → (ugly, boy), (tall, boy))
2. *Subject-Verb:* the main dependency is the one relating the head of the subject and the verb. For example:
   los perros comen carne → *(perro, comer)*
   (dogs feed on meat→ (dog, to feed on))
3. *Verb-Complement:* the main dependency is the one relating the verb and the head noun of the complement. For example:
   recortar gastos → *(recortar, gasto)*
   (to cut back spending → (to cut back, spending))

In Fig. 1, the dependency pairs associated with the variant are obtained in *step 3.*

In the case of syntactic variants, the dependencies of the original multi-word term always remain in the variant. Nevertheless, in the case of morpho-syntactic variants, this only happens when morphological families are applied to conflate the single word terms of the pair. For example, given the term *recorte de gastos* (spending cutback) and its morpho-syntactic variant *recortar gastos* (to cut back spending), using lemmatization we obtain the pairs *(recorte, gasto)* and *(recortar, gasto)*, respectively. Nevertheless, using morphological families we obtain the same dependency pair *(recorte, gastar)* for both the original term and its morpho-syntactic variant[9]. Therefore, the degree of conflation we obtain using morphological families is higher than using lemmatization.

[9] In this example we have supposed that *recorte* is the representative of the family of *recorte* and *recortar*, whereas *gastar* is the representative of the family of *gasto* and *gastar*

To end our explanation we can also see in Fig. 1 an example of the conflation process of the term *casas altas y viejas* (tall and old houses) using the structures previously obtained. In *step 4* tagged text is matched with the pattern, to obtain in *step 5* its associated dependency pairs. Finally, in *step 6* single terms forming each pair are conflated, obtaining the actual pairs to be indexed.

4 Evaluation of the System

The techniques proposed in this article are independent of the indexing engine we choose to use. This is because we first conflate each document to obtain its index terms; then, the engine receives the conflated version of the document as input. So, any standard text indexing engine may be employed, which is a great advantage. Nevertheless, each engine will behave according to its own characteristics [10].

For evaluating the system, five indexing methods have been tested:

pln: plain text eliminating stopwords.
lem: single word term conflation via lemmatization.
fam: single word term conflation via morphological families.
FNL: multi-word term conflation via syntactic dependency-pairs and lemmatization.
FNF: multi-word term conflation via syntactic dependency-pairs and morphological families.

The corpus used for evaluation is formed by 21,899 documents of a journalistic nature (national, international, economy, culture, ...) covering the year 2000. The average length of the documents is 447 words. We have considered a set of 14 natural language queries with an average length of 7.85 words per query, 4.36 of which were content words.

Table 1 shows the statistics of the terms that compose this corpus. The first and second row show the total number of terms and unique terms obtained for the indexed documents, respectively, either for the source text and for the different conflated texts. As we can observe in the upper row, single word term conflation techniques attain a reduction of more than 50% in the number of terms to index whereas multi-word term conflation techniques attain a reduction of nearly 75%. With respect to the number of different terms of the indexes, shown in the lower row, the reduction provided by the elimination of stopwords is negligible, whereas lemmatization and morphological families provide a reduction of 27% and 32%, respectively, with the consequent saving of space and reduction of accessing time to the indexes. Moreover, multi-word term conflation techniques significantly increase the number of index terms since they are complex terms which express syntactic relations. However, we must point out that the use of morphological families to construct such complex terms reduces the number of index terms with respect to the use of lemmatization by 14%, whereas their employment for single word term conflation only attained a relative extra reduction of 6%.

[10] Indexing model, ranking algorithm, etc.

	pln	*lem*	*fam*	*FNL*	*FNF*
Average precision	0.1714	0.2018	0.1982	0.3050	0.3215
Average recall	0.5515	0.6316	0.6028	0.4788	0.5615

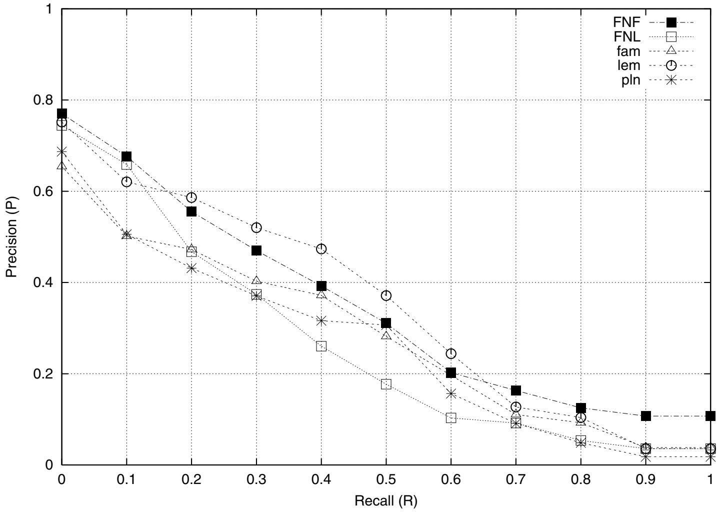


Fig. 2. Average precision and recall and precision vs. recall graph

The results we show in this section have been obtained for the vector-based search engine SMART[11]. In Fig. 2 you can find the results obtained for average recall and precision.[12] We can observe that the application of techniques for single word term conflation, *fam* and *lem*, has led to a remarkable increase in recall whereas the techniques for multi-word term conflation, *FNL* and *FNF*, has led to a remarkable increase in precision. It should be noticed that the isolated employment of morphological families (*fam*) does not always guarantees improvements with respect to lemmatization (*lem*). However, its employment together with multi-word terms (*FNF*) attains a noticeable increase in recall with respect to lemmatization (*FNL*).

With respect to the evolution of precision vs. recall, Fig. 2 confirms the technique *pln* as being the worst one, whereas the best behavior corresponds to *lem* and *FNF*. For low and high recall rates (≤ 0.2, ≥ 0.7) *FNF* is clearly the best one, whereas for the rest of the interval *lem* does better.

5 Conclusions

In this article we have shown how linguistically-motivated indexing can improve the performance of Information Retrieval (IR) systems working on languages

[11] `ftp://ftp.cs.cornell.edu/pub/smart/`

[12] Results for individual queries depend on the characteristics of each query [11].

with a rich lexis and morphology, such as Spanish. In particular, two new text operations to effectively reduce the linguistic variety of documents have been applied: productive derivational morphology for single word term conflation and syntactic dependency-pairs obtained from approximate grammars for multi-word term conflation.

Unlike other related approaches based on parsing and large terminological databases, which gives them a static nature, our approach is dynamic since index terms are identified in running time. It also requires a minimum of linguistic resources, which makes it appropriate for processing European minority languages. As it is a lexical approach, the increase of computational cost is also minimum due to the fact that it is based on finite state technology, allowing its practical application in real systems.

Experimental results allow us to conclude that the isolated employment of morphological families does not always guarantees improvements with respect to lemmatization, but their use together with multi-word terms substantially increases precision whilst maintaining a very acceptable level of recall.

References

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval.* Addison-Wesley, Harlow, England, 1999.
2. Elena Bajo Pérez. *La derivación nominal en español.* Cuadernos de lengua española. Arco Libros, Madrid, 1997.
3. M. Dillon and A.S. Gray. FASIT: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science*, 34(2):99–108, 1983.
4. J.L. Fagan. Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. In *Proceedings of ACM SIGIR'87*, pages 91–101, 1987.
5. Christian Jacquemin and Evelyne Tzoukerman. NLP for term variant extraction: A synergy of morphology, lexicon and syntax. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 25–74. Kluwer Academic, Boston, 1999.
6. J.S. Justeson and S.M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.
7. Gerald Kowalski. *Information retrieval systems : theory and implementation.* Kluwer international series on information retrieval. Kluwer Academic, Boston, 1997.
8. Mervyn F. Lang. *Spanish Word Formation: Productive Derivational Morphology in the Modern Lexis.* Croom Helm. Routledge, London and New York, 1990.
9. M. Lennon, D.S. Pierce, and P. Willett. An evaluation of some conflation algorithms. *Journal of Information Science*, 3:177–183, 1981.
10. Jesús Vilares, David Cabrero, and Miguel A. Alonso. Applying productive derivational morphology to term indexing of spanish texts. In *Computational Linguistics and Intelligent Text Processing*, LNCS 2004, pages 336–348. Springer-Verlag, 2001.
11. Jesús Vilares, Manuel Vilares, and Miguel A. Alonso. Towards the development of heuristics for automatic query expansion. In H. C. Mayr, J. Lazansky, G. Quirchmayr, and P. Vogel, editors, *Database and Expert Systems Applications*, LNCS 2113, pages 887–896. Springer-Verlag, Berlin-Heidelberg-New York, 2001.

Multi-document Summarization Using Informative Words and Its Evaluation with a QA System

June-Jei Kuo, Hung-Chia Wung, Chuan-Jie Lin, and Hsin-Hsi Chen

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
{jjkuo,hjwong,cjlin}@nlg2.csie.ntu.edu.tw,
hh_chen@csie.ntu.edu.tw

Abstract. To reduce both the text size and the information loss during summarization, a multi-document summarization system using informative words is proposed. The procedure to extract informative words from multiple documents and generate summaries is described in this paper. At first, a small-scale experiment with 12 events and 60 questions was made. The results are evaluated by human assessors and a question answering (QA) system respectively. This QA system will help to prevent from drawbacks of human assessors. They show good performance of informative words. That encourages large-scale evaluation. An experiment is further conducted, which contains in total 140 questions out of 17,877 documents. Amongst these documents, 3,146 events were identified. The experimental results have also shown that the models using informative words outperform pure heuristic voting-only strategy when the metric of relative precision rate is used.

1 Introduction

The research of text summarization begins in the early 60s (Edmundson, 1964, 1969) and is one of the traditional topics in natural language processing. Recently, it attracts new attention due to the applications on the Internet. At this information explosion age, how to filter useless information, and to adsorb and apply information effectively become important issues to users. Many papers about document summarization have been proposed (Hovy and Marcu, 1998). Most of the previous works were done on single document summarization. Recently, the focus shifted to multiple documents summarization (Chen and Huang, 1999; Lin and Hovy, 2001; Mani and Bloedorn, 1997; Radev and McKeown, 1998; Radev, Blair-Goldensohn and Zhang, 2001) and even multilingual summarization (Chen and Lin, 2000). Of these, Chen and Huang (1999) employed named entities and other signatures to cluster documents; while as punctuation marks, linking elements, and topic chains to identify the meaningful units (MUs); employed nouns and verbs to find the similarity of MUs; and finally used a heuristic voting-only strategy[1] to generate summaries.

Although experimental results of Chen and Huang (1999) seemed promising, some issues had to be addressed as follows.

[1] The MUs that were reported by more than reporters were selected.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 391–401, 2002.

(1) Goldstein, et al. (1999) mentioned that summary length depends on the document type, and fixed compression ratio is impractical. The summarization size of Chen and Huang's system is fixed and cannot be used to study the variance between the length and the precision rate on Chinese newswire documents.
(2) The presentation order of sentences in a summary was based on the relative positions in the original documents instead of their importance. Thus, users might stop reading or miss the deferred appearing information.
(3) The voting strategy gives a shorter summarization, which missed unique information reported only once.

This paper will follow the basic ideas of Chen and Huang (1999) on multi-document summarization and tackle the above problems. It is organized as follows: Section 2 presents a basic multi-document summarization system. Section 3 uses informative words to modify this system. The extraction of the related informative words and the sentence selection methodologies are described. Conventional evaluation model, i.e., human assessors, is adopted. Section 4 presents a QA system and introduces a new automatic evaluation model. Manual evaluation and automatic evaluation are compared. Section 5 shows a large-scale experiment. Two metrics, i.e., document reduction rate and QA precision rate, are considered. Finally, Section 6 is the conclusion.

2 A Basic Summarization System

Fig. 1 shows the architecture of a basic multi-document summarization system, which is used to summarize Chinese news from on-line newspapers. It is composed of two major components: a news clusterer and a news summarizer. The news clusterer receives a news stream from multiple on-line news sites, and directs them into several output news streams according to events. An event is denoted by five basic entities such as people, affairs, time, places and things. A news summarizer summarizes the news stories in each event cluster. All the tasks are listed below:

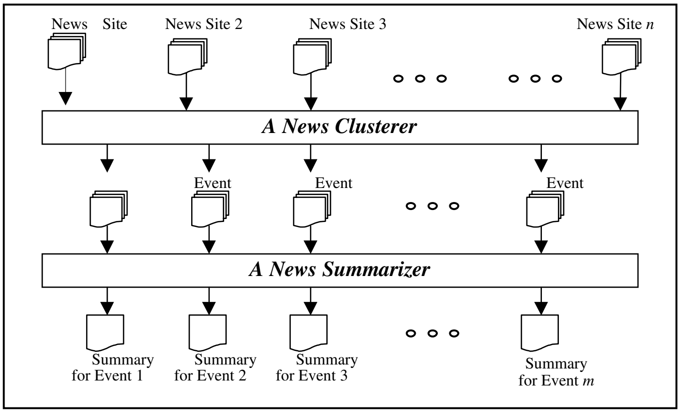


Fig. 1. System Architecture

(1) Employing a segmentation system to identify Chinese words.
(2) Extracting named entities like people, place, organization, time, date and monetary expressions.
(3) Applying a tagger to determine the part of speech for each word.
(4) Clustering the news stream based on the named entities and other signatures.
(5) Partitioning a Chinese text into several meaningful units (MUs)[2].
(6) Linking the meaningful units, denoting the same thing, from different news reports using the punctuation marks, linking elements, topic chains, etc.
(7) Generating the summarization results using the longest sentence preference and voting strategy, which selects sentences reported more than once.

3 Generating Summaries with Informative Words

The concepts of topic words and event words were applied to topic tracking successfully (Fukumoto and Suzuki, 2000). The basic hypothesis is that an event word associated with a story appears across paragraphs, but a topic word does not. In contrast to event word, the topic word frequently appears across all documents. Thus, the document frequency of each word becomes an important factor in searching for the appropriate sentences ready for making summaries. As to the event words, that have higher term frequency in a document, will be more distinctive for the document. Therefore, we defined the words that have both high document frequency and high term frequency as informative words, and used them to improve the performance of step (7) of the basic system, which is specified in Section 2.

3.1 Informative Words and Sentence Selection for Summarization

The score function (IW) of an informative word W_{id} is defined as (3). Ntf(W_{id}) is normalized term frequency of term W_{id}. tf(W_{id}) and mtf(d) are term frequency of W_{id}, and mean term frequency in document d, respectively. D(W_{id}) denotes document frequency of W_{id}, and N is total number of documents in an event. In formula (3), λ denotes a weighted number that can be learned from a corpus. λ was set to 1/2 and 1 in the later experiments.

$$\mathrm{Ntf}(W_{id}) = \frac{tf(W_{id})\text{-}mtf(d)}{tf(W_{id}) + mtf(d)} \tag{1}$$

$$\mathrm{DF}(W_{id}) = \mathrm{D}(W_{id}) / \mathrm{N} \tag{2}$$

$$\mathrm{IW}(W_{id}) = \lambda{*}(1) + (1\text{-}\lambda){*}(2) \tag{3}$$

In summarization, the more informative words a MU contains, the more possible the MU is used for generating summaries. In this paper, only the top 10 terms with

[2] Because Chinese writers often assign punctuation marks at random (Chen, 1994), the sentence boundary is not clear. Meaning units (MUs) are used for clustering instead of sentences. Here, a MU that is composed of several sentence segments denotes a complete meaning.

the higher IW scores will be chosen as informative words for a document. The score of each MU symbolizes the total number of informative words in it. The MUs with the highest score will be selected. Moreover, the selected MUs in a summary will be arranged in the descending order. In other words, the sentences which have more important MUs will appear before the less ones in a summary. In this case, even if the readers unfortunately stop reading the summaries half way, they would not miss out much important information.

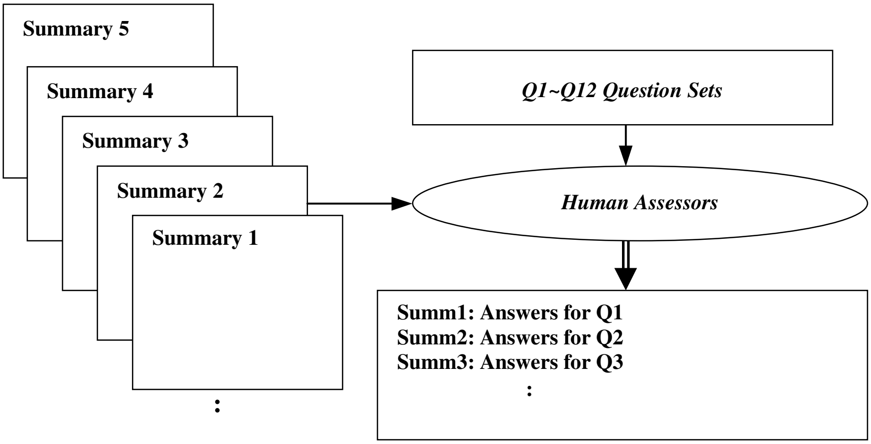


Fig. 2. Example of QA Task

3.2 Experiment Result

Fig. 2 shows our block diagram of the intrinsic evaluation task (Tsutomo, Sasaki and Isozaki, 2001) on text summarization by referring the SUMMAC Q&A evaluation (SUMMAC, 1998). For simplicity, we call it *QA task*. First, the question sets (query sets) are collected under the document collection. While as, the corresponding answer sets are made after reading all the documents. After various kinds of document summaries are completed, the assessors will be involved in the evaluation. Each assessor will be assigned for summary texts and their related question sets. During the evaluation, the reading and the answering time will be recorded. When assessors finish the question and answering task, we review their answers responding to its respective answer sets and compute the precision rate of each question. Besides, the average document reduction rate and the average Q&A precision of various types of summary text are computed, respectively.

In our experiment, the test data is collected from 6 news sites in Taiwan, they are: China Times, Commercial Times, China Times Express, United Daily News, Tomorrow Times, and China Daily News, through the Internet. There are in total 17,877 documents (near 13MB) from January 1, 2001 to January 5, 2001. The total number of MUs is 189,774. After clustering, there are 3,146 events. Because of assessor cost, only 12 events were selected randomly in the first stage. 60 questionnaires (5 questions of each event) are made manually with answers to their related documents. Moreover, 12 members of our laboratory who are all graduate students majoring in computer science are selected to conduct these following experiments: (1) full text (FULL), (2) Chen and Huang's system (1999) as the base line system (BASIC) (3) term frequency only with vote strategy (TFWV, i.e., λ=1), (4) informative words with vote strategy (PSWV, i.e., λ=1/2) (5) term frequency without vote strategy (TFNV, i.e., λ=1), and (6) informative words only without vote strategy (PSNV, i.e., λ=1/2). The above "proposed system" denotes our text summarization system using informative words. Each assessor evaluates a summarization method twice, using different question sets (i.e., answer only once per event) shown as Table 1. The characters A, B, C, …, L in the first column denote the assessors A, B, C, …, L. The names in the first row are the types of summary text. Symbol Q*n* in the cell denotes the question set for event *n*. To evaluate objectively, each assessor does not know the text types what he (she) assesses. The experimental results are shown in Table 2. R&A time means the summation of reading time and answering time. On the one hand, Reduction Rate-S and Reduction Rate-T mean the relative reduction rate of size and R&A time, respectively. The definition of Relative Reduction Rate of size is (Size of a specified system)/(Size of FULL). The average precision and its relative variance of each text type are also given to show the statistical information.

Table 1. Assessor Assignments

	FULL	BASIC	TFWV	PSWV	TFNV	PSNV
A	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12
B	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7
C	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8
D	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9
E	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10
F	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11
G	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12
H	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7
I	Q3, Q9	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8
J	Q4, Q10	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9
K	Q5, Q11	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10
L	Q6, Q12	Q1, Q7	Q2, Q8	Q3, Q9	Q4, Q10	Q5, Q11

3.3 Discussion

Several observations from Table 2 are shown below.

(1) The size of TFNV and PSNV is larger than that of BASIC (near 15%), but the precision rate of TFNV and PSNV is lower than that of BASIC.
(2) The size of TFWV and PSWV is smaller than that of BASIC, and their precision rate is still smaller than that of BASIC.
(3) The precision rates of both TFWV and PSWV are larger than those of TFNV and PSNV.

The above observations are out of our expectation. From observations (1) and (2), the informative words seem not to be useful in MU selection. From observation (3), the vote strategy seems to be useful in improving the precision. In other words, neglecting the news story reported by only one reporter seems to have no problems in Q&A. However, due to limitations and drawbacks of human assessment, evaluation shown below in the QA task may mislead.

(1) Due to different background among human assessors, the evaluation is unable to be objective. We have to conduct several evaluations in order to obtain correct and objective results. Nevertheless, this will be cost-effective.
(2) Fatigue and limited of time scale to work may effect the assessor to of the assessors to quit reading or read too fast so as to miss the information that will be useful to answer the questions. This will cause the low precision of summarizing the text.
(3) Due to the high cost of the assessors, the large-scale evaluation is nearly impossible.

Table 2. Results Using Question-Answering Task

	FULL	*BASIC*	*TFWV*	*PSWV*	*TFNV*	*PSNV*
Size (Byte)	59637	12974	12002	12348	15192	15267
Reduction Rate-S	**1**	**0.22**	**0.20**	**0.21**	**0.25**	**0.26**
Reading Time (sec)	2224	780	744	660	816	804
Answering Time (sec)	1752	1236	1200	1128	1356	1260
R&A Time (sec)	3976	2016	1944	1788	2172	2064
Reduction Rate-T	**1**	**0.51**	**0.49**	**0.45**	**0.55**	**0.52**
Precision	0.923	0.525	0.513	0.519	0.502	0.513
Variance	0.010	0.047	0.095	0.054	0.712	0.061

4 An Evaluation Model Using Q&A Systems

4.1 Model Using Q&A System

In order to improve the QA task and verify the experimental results, a QA system is used to substitute the human assessors in Fig. 2 and the flow of the revised evaluation model is shown in Fig. 3. Both full texts and summaries are read by QA systems, and QA systems find the answers from full texts and summaries. Although the efficiency of a QA system may affect the evaluation results, that is fair for all summarization models under the same evaluation environment.

The QA system we adopted was borrowed from Lin and Chen (1999), whose main strategies are keyword matching and question-focus identifying. This system has been used in open domain question and answering on heterogeneous data (Lin, *et al.*, 2001). It is composed of three major modules shown as follows:

(1) Preprocessing the Question Sentences
At first, the parts-of-speech are assigned to the words in question sentences. Then, the stop-words are removed. The remaining words are transformed into the canonical forms and considered as the keywords of question sentences. For each keyword, they find all synonyms from the related thesaurus, e.g. WordNet (Fellbaum, 1998). Those terms are the expansion set of the keywords. Moreover, no matter whether the keyword is a noun, a verb, an adjective or an adverb, all the possible morphological forms of the word are also added into this set.

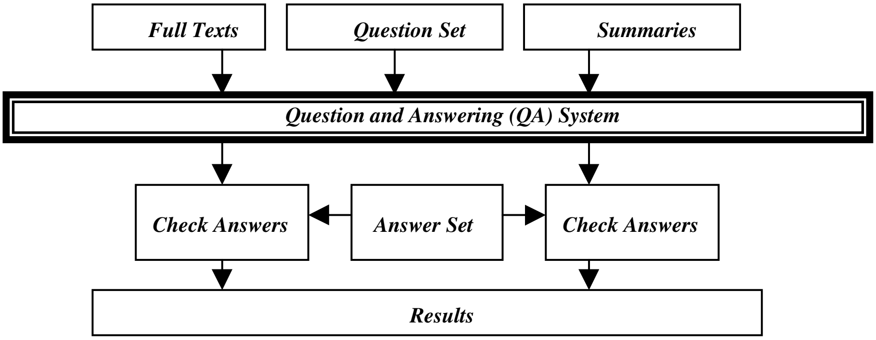


Fig. 3. Revised Evaluation Model

(2) Retrieving the Documents Containing Answers
A full text retrieval system is implemented to decrease the number of documents to be searched for the answering sentences. Each keyword of a expanded question sentence is assigned a weight. Especially, those words tagged with proper-noun markers have been assigned higher weights. This is because they may be presented in the answer. The score of a document D is computed as follows:

$$score(D) = \sum_{t\ in\ D} weight(t) \qquad (4)$$

where t is one of the keywords in expanded question sentence.
Those documents that score more than a threshold are selected as the answering documents. Threshold is set to the sum of weights of the words in the original question sentences. If documents do not have scores bigger than the threshold, we assume that there is no answer to the question.

(3) Retrieving the Sentences Containing Answers
Finally, each sentence in the retrieved documents is examined. Those sentences that contain most words in the expanded question sentence are retrieved. The top five sentences are regarded as the answers. The answers are sorted according to the number of matched words and the retrieving scores computed at step (2).

4.2 Evaluation

The experimental results using the same data in Section 3.2 are shown in Table 3. The precision from Table 1 is reproduced here for comparison. After the QA system reads all documents of 12 events, it will propose five plausible answers for each question. The metric is MRR (Mean Reciprocal Rank) (Voorhees, 2000):

$$MRR = \sum_{i=1}^{N} r_i \Big/ N \qquad (5)$$

where r_i = 1/rank$_i$ if rank$_i$ > 0, or 0 if rank$_i$ = 0. rank$_i$ is the rank of the first correct answer of the ith question, and N is total number of questions. That is, if the first correct answer is at rank 1, the score is 1/1=1; if it is at rank 2, the score is 1/2=0.5, and so on. If no answer is found, score is 0. In this way, the evaluation time can be reduced significantly. That makes large-scale evaluation feasible. Meanwhile, to compare with the precision of QA task in Table 2, we also use five strategies (e.g. Best-1, Best-2, and so on) to compute the precision of the QA system. With Best-1 strategy, the answer must exist in ranked one answer of QA system. With Best-2 strategy, the answer exists in either ranked 1 or 2, or both. Furthermore, to show the feasibility of the proposed evaluation method, we also perform a large-scale experiment that will be discussed in the next section, which human assessment is in question.

Table 3. Results with Small-Scale Data using a QA system

	FULL	BASIC	TFWV	PSWV	TFNV	PSNV
Precision of QA Task	**0.923**	**0.525**	**0.513**	**0.519**	**0.502**	**0.513**
Precision of Best-1	0.881	0.441	0.407	0.457	0.475	0.475
Precision of Best-2	0.915	0.475	0.475	0.508	0.576	0.559
Precision of Best-3	0.949	0.491	0.475	0.508	0.576	0.559
Precision of Best-4	0.966	0.508	0.491	0.525	0.576	0.559
Precision of Best-5	0.966	0.541	0.517	0.525	0.576	0.559
QA_MRR	**0.914**	**0.493**	**0.476**	**0.487**	**0.508**	**0.517**
Relative MRR	**1**	**0.576**	**0.521**	**0.533**	**0.556**	**0.566**

4.3 Discussion

Because the QA system avoids the above limitation and drawback of human assessments, the precisions of some types of summarization text are different from the results shown in Table 2. Observing Table 2 and Table 3, there are some differences shown below:

(1) QA_MRR values of TFNV and PSNV are larger than those of the corresponding TFWV and PSWV. Thus, we can conclude that the vote strategy will lose some useful information.

(2) QA_MRR values of PSWV and PSNV are larger than those of the corresponding TFWV and TFNV. We can draw to the conclusion that using both term frequency and document frequency of informative words will select more important MUs than only using term frequency of informative words.

(3) Comparing the precisions of QA task with the corresponding precisions of best-5 strategy, QA system is better than QA task. Thus, we can say that the QA system can find the answers more effective than human assessors.

In order to show the feasibility of large-scale evaluation using Q&A system, we continue to perform an even greater scale of experiment in the next section, which is impossible to be performed using QA task.

5 Experiments Using Large Documents and Results

5.1 Data Set

From the above analysis, we can conclude that a high performance QA system can be used to play the role of human assessors. Besides the evaluation time and scale, it can obtain more objective and precise results. In the next experiment, the complete data set as described in Section 3.2 was used. Under the data set, 140 new questionnaires are made and 93 questions have been answered. Thus, using these practical questions we can further observe the performance of QA system in text summarization evaluation. Some samples of questions are shown below.

Q68. 英特爾最新發表產品為何？
What is the newest product of Intel Company?
Q95. 歐拉朱萬何時受傷？
When was Mr. Olajuwon wounded?

5.2 Experimental Results and Discussion

Table 5 shows the experimental results using large documents. According the data obtained from the QA system using a large scale of documents, the results are summarized as follows:

(1) Due to the increase of document size, the QA_MRR of all models decreased.
(2) Due to increasing noise of FULL, the QA_MRR of FULL drops drastically. The relative MRRs of the other models increased when comparing with Table 3.
(3) The QA_MRR values of TFWV, PSWV, TFNV and PSNV are also larger than the value of BASIC. This is consistent with the above results in small-scale evaluation using QA system. Thus, informative words in MU's selection present good performance.
(4) The QA_MRR values of PSWV and PSNV are also larger than those of TFWV and TFNV, respectively. To achieve better result, it is recommended to use combination of term frequencey and document frequency in MU's selection.
(5) Since the performance of each model has the similar results to those shown in Table 4, it is feasible to use the QA system in evaluating the performance of large-scale multiple document summarization.

Table 4. Results with Large-Scale Data

	FULL	***BASIC***	***TFWV***	***PSWV***	***TFNV***	***PSNV***
Size (Kbyte)	13,137	1,786	1,771	1,773	2,226	2,218
QA_MRR	0.515	0.314	0.342	0.346	0.359	0.380
Relative MRR	**1**	**0.610**	**0.664**	**0.672**	**0.697**	**0.738**

6 Conclusion

This paper presents a multi-document summarization system using informative words and an automatic evaluation method for summaries using a QA system. Using the normalized term frequency and document frequency, the informative words can be extracted effectively. The informative words are shown to be more useful to select sentences for generating summaries than the heuristic rule. Moreover, the sentences in the summaries can be put in order according to the total number of informative words. In this way, the important sentences are generated in the early part. The summaries can be compressed easily by deleting sentences from the end without losing much important information, and the length of summary can be adjusted robustly. On the other hand, the evaluation processes show that QA system can play an important role in conducting large-scale evaluation of multi-document summarization and make the results more objective than the human assessors. There are still some issues that need further research:.

(1) Investigating to what extent the errors of QA system may affect the reliability of the evaluation results
(2) Using other QA systems to justify the feasibility of the above evaluation model.
(3) Introducing the machine learning method to obtain λ value and its possible size of summary for various kinds of documents.
(4) Using some statistical model and null hypothesis test to study the results' relationship between QA task and QA systems.
(5) Introducing the statistical methods, such as the dispersion values of words among document (Fukumoto and Suzuki, 2000) to find the informative words more effectively for the purpose of improving the performance of the summarization system.

Acknowledgements

The research in this paper was partially supported by National Science Council under the contracts NSC89-2213-E-002-064 and NSC90-2213-E-002-013.

References

1. Chen, H.H.: The Contextual Analysis of Chinese Sentences with Punctuation Marks. Literal and Linguistic Computing, Oxford University Press, 9(4) (1994) 281-289
2. Chen, H.H. and Huang, S.J.: A Summarization System for Chinese News from Multiple Sources. Proceeding of 4th International Workshop on Information Retrieval with Asia Language (1999) 1-7.
3. Chen, H.H. and Lin, C.J.: A Multilingual News Summarizer. Proceeding of 18th International Conference on Computational Linguistics, (2000) 159-165.
4. Edmundson, H.P.: Problems in Automatic Extracting. Communications of the ACM, 7, (1964) 259-263.
5. Edmundson, H.P.: New Methods in Automatic Extracting. Journal of the ACM, 16, (1969) 264-285.
6. Firmin Hand, T. and B. Sundheim (eds): TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop, Washington. (1998)
7. Fukumoto, F. and Suzuki, Y.: Event Tracking based on Domain Dependency. Proceedings of SIGIR 2000 (2000) 57-64.
8. Goldstein, J., Kantrowitz, M., Mittal, V. and Carbonell, J.: Summarizing Text Documents: Sentences Selection and Evaluation Metrics. Proceedings of SIGIR 1999 (1999) 121-128.
9. Hovy, E. and Marcu, D.: Automated Text Summarization. Tutorial in COLING/ACL98 (1998)
10. Lin, C.J. and Chen, H.H.: Description of Preliminary Results to TREC-8 QA Task. Proceedings of The Eighth Text Retrieval Conference (1999) 363-368.

11. Lin, C.J., Chen, H.H., Liu, C.J., Tsai, C.H. and Wung, H.C.: Open Domain Question Answering on Heterogeneous Data. Proceedings of ACL Workshop on Human Language Technology and Knowledge Management, July 6-7 2001, Toulouse France, (2001) 79-85.
12. Lin, C.Y. and Hovy E.: NEATS: A Multidocument Summarizer. Workshop of DUC 2001 (2001) [on-line] Available: http://www-nlpir.nist.gov/projects/duc/duc2001/agenda_duc2001.html
13. Mani, I. and Bloedorn, E.: Multi-document Summarization by Graph Search and Matching. Proceedings of the 10th National Conference on Artificial Intelligence, Providence, RI, (1997) 623-628.
14. Mani, I. et al.: The TIPSPER SUMMAC Text Summarization Evaluation: Final Report, Technique Report. Automatic Text Summarization Conference, (1998)
15. Radev, D.R. and McKeown, K.R.: Generating Natural Language Summaries from Multiple On-Line Sources. Computational Linguistics, Vol. 24 No. 3 (1998) 469-500.
16. Radev, D.R., Blair-Goldensohn and Zhang, Z.: Experiment in Single and Multi-Document Summarization Using MEAD. Workshop of DUC 2001 (2001) [on-line] Available: http://www-nlpir.nist.gov/projects/duc/duc2001/ agenda_duc2001.html
17. Regina Barzilay and Michael Elhada: Using Lexical Chains for Text Summarization. Proceedings of The Intelligent Scalable Text Summarization Workshop, ACL/EACL (1997) 10-17.
18. Tsutomo, H., Sasaki, T. and Isozaki H.: An Extrinsic Evaluation for Question-Biased Text Summarization on QA Tasks. Proceedings of workshop on Automatic Summarization (2001) 61-68.
19. Voorhees: QA Track Overview (TREC) 9, (2000) [on-line] Available: http://trec.nist.gov/presentations/ TREC9/qa/index.htm
20. Fellbaum, C.: WordNet. The MIT Press, Cambridge Masschusettes (1998)

Automated Selection of Interesting Medical Text Documents by the TEA Text Analyzer

Jan Žižka[1] and Aleš Bourek[2]

[1] Department of Information Technologies, Faculty of Informatics,
Masaryk University in Brno, Botanická 68a, 602 00 Brno, Czech Republic
zizka@informatics.muni.cz

[2] Department of Biophysics, Faculty of Medicine,
Masaryk University in Brno, Joštova 10, 662 43 Brno, Czech Republic
bourek@med.muni.cz

Abstract. This short paper briefly describes the experience in the automated selection of interesting medical text documents by the TEA text analyzer based on the naïve Bayes classifier. Even if the used type of the classifier provides generally good results, physicians needed certain supporting functions to obtain really interesting medical text documents, for example, from resources like the Internet. The influence of the functions is summarized and discussed. In addition, some remaining problems are mentioned.

The motivation of developing intelligent text-analyzing software tools is the fact that todays computer users can easily access very large volumes of various data, including text documents – often unformatted – from the Internet or databases. The typical search for text documents is mostly based on using a small set of key-words, which in too many cases results in obtaining very large amounts of data that contain only a small portion of relevant information for a user. Manual filtering of hundreds or thousands of text documents is a tedious and time-consuming work. Therefore, an automated intelligent selection support of interesting documents (see, e.g., [4]) helps users, for example physicians, with their primary tasks. An intelligent filter for pure-text documents naturally depends on the word contents. Thus, the basic task is to develop an efficient text document classifier that can separate documents into *two classes: interesting* and *uninteresting.*

The tool TEA (TExt Analyzer), described briefly in this paper, is based on the naïve Bayes classifier (see, e.g., [3], [5]), which learns to split a set of text documents into the two required classes. TEA's learning needs a good selection of interesting (*positive*) and uninteresting (*negative*) sets of training examples to assess values of *conditional probabilities of word occurrences* in text documents. During its learning for a certain text-document area, TEA creates a dictionary with individual words from documents, occurrence frequencies, and necessary parameters of these words. Words are strings of characters separated by white space (e.g., a space) or standard delimiters (e.g., commas, full stops, etc.). Later, these *a posteriori* probabilities with the *a priori* probability (which is given by

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 402–404, 2002.

the ratio of interesting and uninteresting documents in the whole document set, and which should be as close as to 50%) determine – during the classification process – the class of a new text document. Many extensive tests with real text medical documents (see [1], [2], and [5]) revealed that without physicians' active support the classification accuracy was not as high as necessary for new classified documents that were not used for the training process. Therefore, a set of supporting functions has been added, mainly for required modifications of word dictionaries in the areas of physicians' interests.

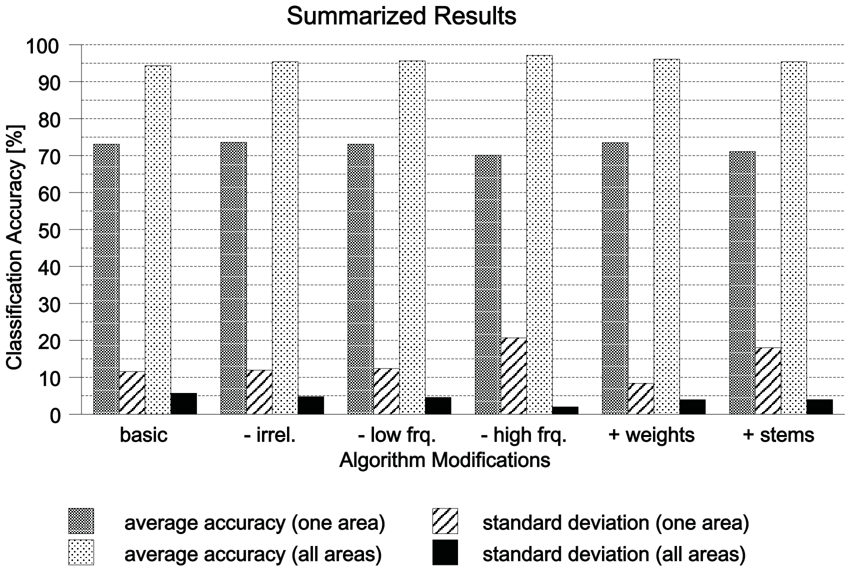


Fig. 1. The graph of various results influenced by TEA's gradually employed supporting functions for the tested large sets of medical text documents.

Fig. 1 illustrates the classification accuracy results (with the learning tested by cross-validation) obtained by gradually employing the various supporting functions for the large set of the real testing textual data obtained from the Internet (results of each function were added to results of previous functions, in the order shown in the graph). Two basic groups of experiments with medical text documents were performed: the *one area* had very similar documents from only one special medical area, *assisted reproduction*, which were choicely preselected before the final testing. In this case, the classification accuracy was between 70% to 75%. On the other hand, the *all areas* contained medical text documents from a rather larger area of the *gynecology*. Here the results were much better, approximately between 94% to 98%, because of a lower document similarity. The results for the basic, unmodified text documents' dictionaries are marked as *basic*. The classification accuracy itself actually did not increase very much, however, the results were more acceptable mainly because of *decreased devia-*

tions of filtering new text-document sets. The users very often need to *eliminate insignificant, irrelevant words*, marked as *−irrel.*, mainly the common words in a language (which was English) plus special words individually defined by a user. Moreover, *removing words having very high or very low frequencies* – marked as *−high frq.* and *−low frq.*, respectively – helps in many cases to obtain less amount of uninteresting documents, e.g., by decreasing the too high degree of document similarity. Especially, eliminating too frequent words usually improves the classification results. Also, in some cases, *increasing weights of selected words* helped for certain documents – marked as *+weights.* Finally, the last graph item marked *+stems* shows results obtained by the application of *word stems*, where different forms of the same word (e.g., *method* and *methods*) were transferred into one form (i.e., *method* in the example).

Generally, the naïve Bayes classification provided very good accuracy in the most cases. The best results can be expected when selecting documents from a particular area of interest among a large number of different-topic documents. The supporting functions for modification of dictionaries usually increase results for individual users who look for their special information. The splitting of very similar documents would need to employ a kind of background and/or domain knowledge because using only the word contents is not sufficient for obtaining high degrees of the classification accuracy – this approach would need more research. In addition, there are certain difficulties with collecting balanced classes of training examples. It is relatively easy to collect positive examples, however, very often too high numbers of downloaded negative examples prevent creating the balanced sets of training examples without the additional workload. And the last but not the least problem is also the necessity to use not only individual words as important elements for the classification – in many cases, word conjunctions define relevant documents whereas separated words eventuate in many irrelevant documents. For example, using words like *learning* and *machine* will provide a huge number of irrelevant documents if a user looks for documents from the *machine learning* area. In this case, TEA enables its users to add necessary word conjunctions and thus to improve the classification accuracy.

References

1. Bourek, A., Suchý, M., and Svoboda, P.: Standards of Efficient Medical Care (SEMC). In: Proceedings of the 7^{th} International Conference on System Science in Health Care, Lyon, ISSHC, (2000), 436–439
2. Bourek, A., Žižka, J., Ventruba, P., and Frey, L.: The Use of the Internet for Monitoring Trends in Assisted Reproduction and Reproductive Medicine. Gynekolog, 5, (2000), 220–223 (in Czech)
3. Michie, D., Spiegelhalter, D. J., and Taylor, C. C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood, New York, (1994)
4. Special Issue of Machine Learning Journal on Information Retrieval. Machine Learning Journal, Vol. 39, No. 2/3, May/June. Kluwer Academic Publishers (2000)
5. Žižka, J., Bourek, A., and Frey, L.: TEA: A Text Analysis Tool for the Intelligent Text Document Filtering. In: Text, Speech, and Dialogue. Springer Verlag, Berlin, Heidelberg, New York, (2000), LNCS 1902, 151–156

Chinese Documents Classification Based on N-Grams*

Shuigeng Zhou[1] and Jihong Guan[2]

[1] State Key Lab of Software Engineering, Wuhan University, Wuhan, 430072, China
zhousg@whu.edu.cn
[2] School of Computer Science, Wuhan University, Wuhan, 430072, China
jhguan@wtusm.edu.cn

Abstract. Traditional Chinese documents classifiers are based on keywords in the documents, which need dictionaries support and efficient segmentation procedures. This paper explores the techniques of utilizing N-gram information to categorize Chinese documents so that the classifier can shake off the burden of large dictionaries and complex segmentation processing, and subsequently be domain and time independent. A Chinese documents classification system following above described techniques is implemented with Naive Bayes, kNN and hierarchical classification methods. Experimental results show that our system can achieve satisfactory performance, which is comparable with other traditional classifiers.

Keywords: Chinese documents Classification, N-grams, Feature selection, Bayesian Classification, kNN Method, Hierarchical Classification.

1 Introduction

Documents classification is a supervised learning process, defined as assigning category labels (pre-defined) to new documents based on the likelihood suggested by a training set of labeled documents. With the rapid growth of online information, documents classification has become one of the key techniques for processing and organizing text data. And documents classification technique has been used to classify news stories [1], to find interesting information on the Web [2], and to guide a user search through hypertext [3]. Traditional documents classifiers are generally based on keywords in the documents, which means that the training and classifying processes need dictionaries support and efficient segmentation processing. As far as Chinese documents classification is concerned [4], segmentation is a complex task [5, 6]. Current Chinese segmentation systems are generally large, and of low accuracy and efficiency. Recalling that languages are domain-dependent and time-varying, the dictionaries and segmentation procedures used in the classifiers must be updated so that the classifiers are still effective and efficient in the changed language environment. With these points in mind, it is natural to pursue a documents classifier system that does not rely on dictionaries and segmentation processing. This paper explores the techniques of utilizing N-gram information to classify Chinese

* This work was supported by China Post-doctoral Science Foundation and the Natural Science Foundation of China (NSFC) under grant number 60173027.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 405–414, 2002.

documents so that the classifier can shake off the burden of large dictionaries and complex segmentation processing, and subsequently be domain and time independent. Such a Chinese documents classification system is developed with Naive Bayes, kNN and hierarchical classification methods. Experimental results show that classifying Chinese documents based N-grams can achieve satisfactory performance of being comparable with other traditional Chinese documents classifiers.

In Section 2, we present an algorithm for Chinese N-grams extraction. Classification features selection approaches are discussed in Section 3. A Chinese documents classifier system developed with the above-mentioned techniques is evaluated in Section 4. The paper is concluded in Section 5.

2 Chinese N-Grams Extraction

2.1 N-Grams and Chinese Documents Classification

As one of typical Oriental languages, Chinese is quite different from Western languages. Chinese has a character set of large size. A relatively comprehensive contemporary Chinese dictionary contains more than ten thousand unique Chinese characters or 汉字 (*han zi*). And a lot of Chinese characters also appear as a word in certain context. Furthermore, written Chinese text has no space to separate words or 词(*ci*) from each other, which makes word or phrase extracting from text automatically a very complicated task. An N-gram in Chinese is a sequence of N Chinese characters consecutively appearing in text, it may be a word or not. For a Chinese document d of length L, if all punctuation marks and other symbols but Chinese characters are ignored, *i.e.* the document is treated as a Chinese characters sequence of length L, then there are at most $L(L+1)/2$ N-gram items in d. In reality, a document cannot contain so many N-gram items. Usually, the documents are split by punctuation marks into series of sentences, and the N-grams are extracted from the sentences. Therefore, the longest N-grams are the longest sentences. Suppose the training documents collection D contains N_D documents, the average number of sentences in each document and the average length of sentences are N_s and L_s respectively, then there are at most $N_D N_s L_s(L_s+1)/2$ N-gram items in the training document collection D. Obviously, the number of N-gram items in the training collection will be substantially large, which reminds us of carefully selecting the N-grams while training classifiers. Furthermore, considering that documents classification is a kind of semantic oriented operation, the selected N-grams should be able to express documents implication accurately as far as possible. However, the contribution of each N-gram item to classification performance is quite different, so how to select the proper N-grams for classification poses a key technical problem here. The usefulness of a N-gram item for classification can be measured qualitatively by its occurrence *frequency*, *distribution* and *centralization*, which are defined as follows.

Definition 1 The *frequency* of N-gram item t occurring in document d is its occurrence count in d. We denote it *tf*.

Definition 2 The *distribution* of N-gram item t in document class c is the number of documents that contain t. We denote it *df*.

Definition 3 The *centralization* of N-gram item *t* in text collection *D* is defined to be the inversion of the number of classes that include *t*. We denote it *icf*.

Intuitively, a N-gram item with higher *tf*, *df* and *icf* is more useful to classification, *i.e.* it is more distinguishable. However, there is no simple mathematical approach to guide selection of the most distinguishable N-grams in terms of their *tf*, *df* and *icf*. In this paper, for the simplicity of processing and to reduce the chance of extracting N-grams with less distinguishing power, we specify three constraints as follows.

Constraint 1 Given a pre-specified minimum value of *tf*, being denoted as *min-tf*, a N-gram item *t* in document *d* is extracted only if its *tf* is no less than *min-tf*.

Constraint 2 Given a pre-specified minimum value of *df*, being denoted as *min-df*, a N-gram item *t* in class *c* is extracted only if its *df* is no less than *min-df*.

Constraint 3 Given a pre-specified minimum value of *icf*, being denoted as *min-icf*, a N-gram item *t* in collection *D* is extracted only if its *icf* is not less than *min-icf*.

Above, the thresholds *min-tf*, *min-df* and *min-icf* are selected by experiments over the training documents collection. Here, *tf*, *df* and *icf* are defined in a way of being independent of the training documents collection. Considering the fact that the length of different documents, the number of documents in different classes and the number of classes in different collections are quite different, it is more reasonable and practicable to define *tf*, *df* and *icf* with regard to the document length, the number of documents in the classes and the number of document classes in the training documents collection.

2.2 A Fast Algorithm for Chinese N-Grams Extraction

A naive algorithm for extracting Chinese N-grams from training documents collection is to scan the collection and obtain all Chinese N-grams conforming to *Constraint* 1, *Constraint* 2 and *Constraint* 3 in one pass. For small training collections, this way is effective and efficient. However, as the training collection becomes larger and larger, the number of N-gram items in the training collection will increase exponentially, and this naive algorithm cannot extract the required N-grams efficiently due to memory limit. Here we adopt a stepwise algorithm to extract the Chinese N-grams: Firstly, the 1-grams conforming to *Constraint* 1 and *Constraint* 2 are extracted by scanning the training documents; Then the candidates of 2-grams are created from the selected 1-grams, and the required 2-grams are obtained by filtering out the 2-grams not conforming to *Constraint* 1 and *Constraint* 2; In a similar fashion, the required 3-grams, 4-grams and so on are extracted. At last, the N-grams obtained from the above processing will be filtered again by applying *Constraint* 3 to get the final N-grams set. Before describing this algorithm in details, we present the following definition and lemma.

Definition 4 For *i*-gram item t_i and *j*-gram item t_j with $i \geq j$, if t_j is contained in t_i, than we say t_j is a sub-item of t_i, and denote $t_j \subseteq t_i$.

Lemma 1 If *i*-gram item t_i meets Constraint 1 and Constraint 2, then all the sub-items of t_i meet Constraint 1 and Constraint 2 too.

It is straightforward to prove *Lemma* 1 by applying Definition 1, 2 and 4. For the sake of space, the proof is omitted. Based on *Lemma* 1, we have a stepwise Chinese N-grams extraction algorithm as follows.

Algorithm 1: Chinese N-grams extraction
Input: document collection D, *min-tf*, *min-df*, *min-icf* and *MAX-N*.
Output: A set of N-grams S ($N \leq MAX\text{-}N$) that meet *Constraint* 1, 2 and 3.
Process (basic steps):

1. Finding the 1-grams set S_1: Scanning all documents in D one by one, and extracting all 1-grams that meet Constraint 1 and 2.
2. Finding the 2-grams set S_2: Carrying out Cartesian product $S_1 \times S_1$ to produce the candidate 2-grams set C_2 from which the items not conforming to Constraint 1 and 2 are removed, and the left items make up S_2.
3. For i=3 to *MAX-N* do:
   3.1 Constructing the candidate i-grams set C_i: $C_i = \Phi$, for two arbitrary (i-1)-gram items t_m and t_n in S_{i-1}, $t_m(k)$ and $t_n(k)$ (k=1~(i-1)) refer to the k-th character in t_m and t_n respectively. If $t_m(k+1)=t_n(k)$ for k=1~(i-2), then
   $C_i=C_i \cup t_m t_n(i-1)$.
   3.2 Removing the items in C_i that not conforming to Constraint 1 and 2, then the left items make up S_i.
4. $S'=S_1 \cup \ldots \cup S_{MAX\text{-}N}$.
5. Applying *Constraint* 3 over S', the results consititute the final N-grams set S.

2.3 Select the Maximum N

There is another problem to be solved: How much should the largest value of N be? Intuitively, the maximum N, simply *MAX-N*, should be such a value that the i-grams with $i \leq MAX\text{-}N$ can cover most of keywords in the training documents collection. According to the statistic analyses of Chinese documents [7], as far as occurrence frequency is concerned, in Chinese documents, 1-chrarcter words make up the dominating part, and the next is 2-character words, then 3-character and 4-character words. The number of words with more than 4 characters is quite small. So for any Chinese document, we can basically represent it by using words bag model with the words consisting of 1 to 4 characters. In other word, the largest value of N can be 4 because the N-grams (N=1~4) can cover all words consisting of 1 to 4 characters in Chinese documents.

3 Features Selection

We take the extracted N-grams above as documents classification features, which are also referred to as *terms*. However, the number of the extracted N-grams is still very large, which will affect the performance and efficiency of classification. So a feature selection process is necessary over the extracted N-grams to get a relatively more optimal and smaller subset of document features for classification. Three statistic approaches popularly used to accomplish the selection task in machine learning are used in the context of text classification. They are information gain (IG), mutual information (MI) and χ^2-statistic. By comparing their classification results, we can choose the best approaches for Chinese text classification based on N-grams.

There may be still some redundant features exist even after the above selection process. In Chinese documents, some fixed words such as specific names are composed of fixed Chinese characters that occur simultaneously in the documents. For example, "墨西哥" Mexico is a country name, if it is selected as a document feature, then "墨西", "西哥" will be selected too. However, these two terms are essentially redundant in the context under consideration, and should be removed from the selected features set. Considering such cases, we give a constraint for removing redundant features as follows. Certainly, this constraint cannot guarantee that all redundant features will be removed.

Constraint 4 Given two N-gram items t_i and t_j, if $t_i \supset t_j$ and score(t_i)=score(t_j), then one of them is redundant, and only t_i is kept as classification feature. Here, score (.) refers to any evaluation function of the three feature selection methods.

We take a features selection scheme that consists of four steps as follows.

1. Using *Algorithm* 1 described in Section 2 to extract the N-grams (1≤N≤4) that conform to *Constraint* 1, *Constraint* 2 and *Constraint* 3. We denote the selected features set *F*;
2. Scoring each feature in *F* with one of the feature selection approaches in this Section. For example, suppose *Information Gain* is chosen, then for each feature *f* in *F*, *IG*(*f*) is evaluated. When all features are scored, then sort the features according to their scores in decreasing order;
3. Removing the redundant features in *F* according to *Constraint* 4, the remaining features constitute a new features set F_1;
4. Suppose that N_s specifies the number of features used for classification, taking N_s N-gram items with the highest score from F_1, which make up the final features set F_s that is used for training classifier.

4 Performance Evaluations

4.1 Experimental Documents Collections

There are no commonly used Chinese documents collections for classification test available yet, such as the Reuters and OHSUMED collections for English classification test. Therefore, we have to collect training and test documents manually by ourselves. Two experimental documents collections were established, which we denote G1 and G2. G2 was built specifically for hierarchical classification experiment. Documents in G1 are news documents from the *People's Daily* and the *Xinhua News*. G1 contains 20 distinctive classes in which there are 2850 documents in total. The number of documents in each class of G1 is quite different. The largest class (Politics) has 617 documents, and the smallest (Electronics) owns only 55. Documents in collection G2 were downloaded from *Yahoo* China (http://cn.yahoo.com) and the BBS of Fudan University. Theses documents spread over the leaf nodes of a topic hierarchy. The topic hierarchy has three levels: the first level has 4 classes; the second level 12 classes, and the third 5 classes. The sum of class-labels in the topic hierarchy is 21, in which 15 are leaf classes. Totally, there are 1155 documents in collection G2, and all documents are in the leaf classes. The largest leaf class (Aerospace) has 141 documents, and the smallest (Football) has only 45.

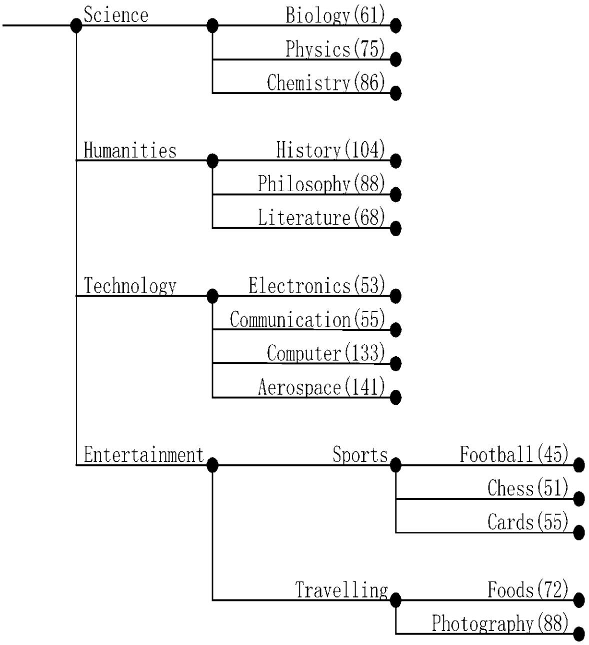


Fig. 1. Hierarchy of documents collection G2

Table 1. Documents collection G1

Class name	Documents number	Class name	Documents number
Politics	617	Mining	67
Sports	350	Military	150
Economy	226	Computer	109
Agriculture	86	Electronic	55
Environment	102	Communication	52
Astronomy	119	Energy	65
Arts	150	Philosophy	89
Education	120	History	103
Medicine	104	Law	103
Transport	116	Literature	67
Total documents number		2850	
Avg. documents per class		142.5	

4.2 Experimental Results

We developed a Chinese classification system based on the techniques described above with VC++ 6.0 on the Windows NT 4.0 platform. The system was trained and tested over collections G1 and G2. In the experiments, the documents collections are split into two parts in terms of a certain ratio over each class. One part is used for training, and the remaining for test. *Recall* and *Precision* are used to measure the system performance, which are abbreviated to *r* and *p* respectively. At first, *r* and *p* are computed separately for each class; then the final results are obtained by averaging the *r* and *p* values over all classes. The default experimental setting is as follows: the ratio of training documents to test documents is 7:3, *i.e.* 70% documents are used for training, and the remaining 30% for test; classification feature is 2-grams; feature selection approach is IG; classification method is kNN; experiments are carried out over collection G1. For simplicity, 1-gram represents using only 1-gram items for classification, 1/2-gram means using 1-gram items and 2-gram items simultaneously for classification, and so on.

Firstly, the effect of N-grams used for classification on classification performance is explored. Fig. 2 illustrates the classification results when taking different N-gram items for different feature sizes (different number of N-grams used for classification) on collection G1. We consider five different cases: 1-grams, 2-grams, 1+2-grams, 2+3+4-grams and 1+2+3+4-grams; the number of N-grams used for classification is from 300 to 2000. From Fig.2, we can draw the following conclusions:

1. Using only 1-grams can also achieve acceptable classification results. As the number of 1-grams grows, a performance peak can reach, following that, classification performance will degrade quickly. The reason lies in that the number of distinct 1-grams (*i.e.* Chinese characters) in the training documents is limited, and not all of them are relevant to document classes. As the number of classification features increase, some 1-grams irrelevant to document classes will be included in the classification features set, these 1-grams are equivalent to noise, which will influence the classification results negatively.
2. With the number of classification features less than 2000, 1+2-grams and 1+2+3+4-grams outperform the other three cases, and 1+2+3+4-grams outperforms 1+2-grams. Due to the effect of 1-grams, 1+2-grams and 1+2+3+4-grams also manifest similar performance trends, *i.e.* up first and then down.
3. When the number of classification features is less than 2000, the classification performance by using only 2-grams keeps improving as classification features increases. The reason is that the number of distinctive 2-grams (*i.e.* 2-character words) relevant to document classes is very large (at least larger than 2000 as far as our training collection is concerned); so increasing the number of classification features will improve the classification results.
4. While using equal size of classification features, 2-grams outperforms 2+3-grams due to the fact that 3-grams can be represented by 2-grams, which implies that a set of 2-grams can semantically cover a set of 2+3-grams of the same size when they are selected from the same collection with the same approach.

We then examine the effect of feature selection methods on classification performance. Fig. 3 gives the experimental results corresponding to three different feature selection approaches: information gain (IG), mutual information (MI) and χ^2-statistic, which shows that χ^2-statistic has the best effectiveness, and IG is the second. However, the performance difference between χ^2-statistic and IG is negligible. Following that, we compare the classification performance for different training documents sizes and illustrate the experimental results in Fig. 4. It’s understandable that the increasing of training documents number leads to improved classification performance. Finally, we investigate the performance of Bayes method, kNN method and hierarchical classification for different feature sizes on collections G1 and G2; the results are shown in Fig. 5 and Fig. 6 respectively. Fig.5 shows the comparison of Bayes and kNN; and Fig.6 illustrates the comparison of flat and hierarchical classification. Obviously, as far as flat classification is concerned, kNN outperforms Bayes, especially when the number of classification features is large. However, It's worthy of our attention that even with 200 or 300 classification features, Bayes method can achieve good performance, especially for the case of 1+2-grams. And the important result is that hierarchical classification can achieve much better performance than flat classification.

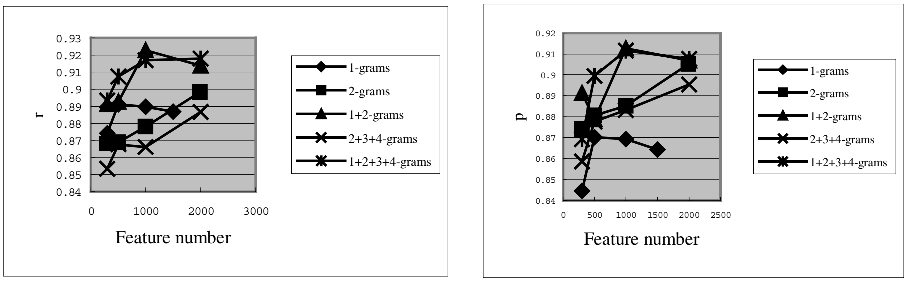


Fig. 2. Comparison of N-grams for different features sizes over collection G1

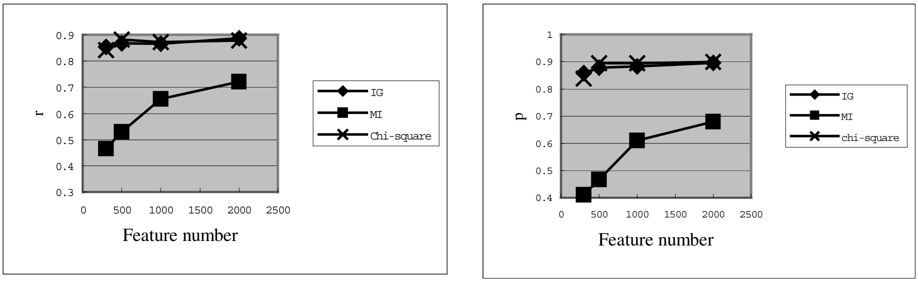


Fig. 3. Comparison of feature selection methods for different feature sizes over collection G1

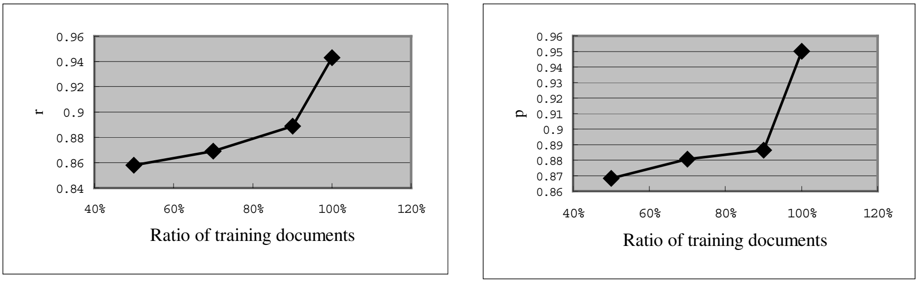


Fig. 4. Performance for different ratios of training documents of collection G1

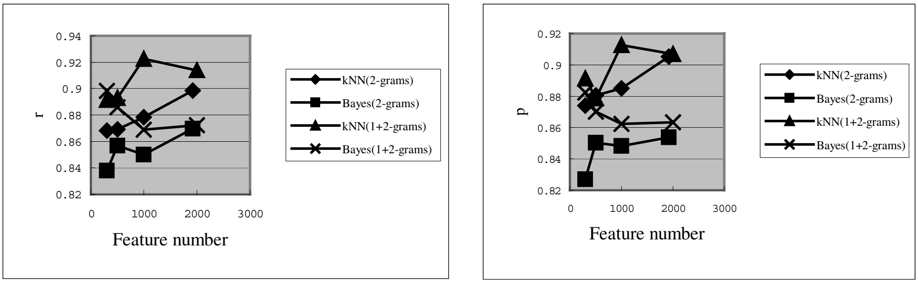


Fig. 5. Comparison between Bayesian and KNN methods for different feature sizes over collection G1

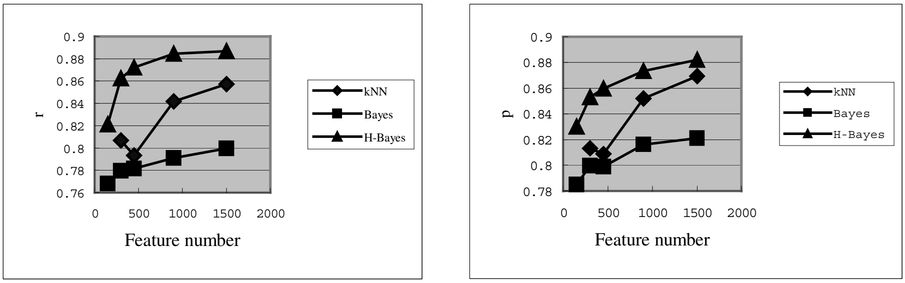


Fig. 6. Comparison between flat classification and hierarchical classification for different feature sizes over collection G2

5 Conclusions

In this paper, we have presented effective and efficient techniques to classify Chinese documents based on N-gram information. Due to using N-grams to represent documents, our classifier needs no dictionary support and segmentation processing, which makes it more competitive in flexibility and practicability than the conventional Chinese documents classifiers. Series of experiments demonstrate its satisfactory performance. More detailed information about the research can be found in [8].

References

1. B. Masand, *et al*. Classifying news stories using memory-based reasoning. In International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 59-65, 1992.
2. K. Lang. Newsweeder: learning to filter netnews. In International Conference on Machine Learning (ICML), 1995.
3. T. Joachims, *et al*. Webwatcher: A tour guide for the World Wide Web. In International Joint Conference on Artificial Intelligence (IJCAI), 1997.

4. T. Zou, *et al*. The Design and Implementation of an Automatic Chinese Documents Classification System, *Journal of Chinese Information Processing*, 13(3): 26-32, 1999. (In Chinese).
5. Y. Liu, Q. Tan, and X. Shen. *Modern Chinese Segmentation Specification and Automatic Segmentation Methods for Information Processing*, Tsinghua University Press. (In Chinese).
6. Z. Wu and G. Tseng. Chinese Text Segmentation for Text Retrieval: Achievements and Problems. Journal of th American Society for Information Science, 44:532-542, October 1993.
7. B. Zhao and L. Xu. Processing Chinese Information with Computer, Vol.2. Space Publisher House, 1988. (In Chinese).
8. S. Zhou. Key Techniques of Chinese Text Database. PhD thesis of Fudan University, China. 2000.

Cross-Lingual Document Similarity Calculation Using the Multilingual Thesaurus EUROVOC

Ralf Steinberger, Bruno Pouliquen, and Johan Hagman

European Commission, Joint Research Centre
Institute for the Protection and Security of the Citizen (IPSC)
Cybersecurity and New Technologies for Combating Fraud Unit (CSCF)
21020 Ispra (VA), Italy
{Ralf.Steinberger,Bruno.Pouliquen,Johan.Hagman}@jrc.it
http://www.jrc.it/langtech

Abstract. We are presenting an approach to calculating the semantic similarity of documents written in the same or in different languages. The similarity calculation is achieved by representing the document contents in a language-independent way, using the descriptor terms of the multilingual thesaurus EUROVOC, and by then calculating the distance between these representations. While EUROVOC is a carefully handcrafted knowledge structure, our procedure uses statistical techniques. The method was applied to a collection of 5990 English and Spanish parallel texts and evaluated by measuring the number of times the translation of a given document was identified as the most similar document. The good results showed the feasibility and usefulness of the approach.

1 Introduction

Following the introductory clarification of the question what semantic document similarity is (1.2), why (1.2) and how (1.3) to measure it, and how this work fits in with other activities at the JRC (1.4), section 2 summarises earlier work [11] on assigning controlled vocabulary thesaurus terms (henceforth called *descriptors*) to texts. Section 3 then describes how we use these lists of automatically assigned thesaurus descriptors as kind of a conceptual interlingua which allows to measure the semantic document similarity without using any dictionaries. Sections 4 and 5 discuss the limitations of the adopted method and give an outlook on future work.

1.1 What Is Document Similarity?

Although everybody has an intuition regarding the question whether two documents are similar, and to what extent, it is difficult to put one's finger on this intuition. Similarity measure can be based on the degree of lexical overlap between the texts that are to be compared, but it is also possible to use a more abstract measure by comparing the document contents. Latent semantic indexing approaches, for instance, go beyond counting the mere overlap of words used in texts and map words and documents to a more complex conceptual space [4]. Document similarity can also be based on stylistic information such as sentence length, the type-token ratio, word variation and other stylometric features. Finally, meta-information such as document type,

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 415–424, 2002.

author name, source of the text, time of writing and other information aspects could be used.

Our own approach is to map different texts onto an existing knowledge structure, i.e. the multilingual thesaurus EUROVOC (see 2.1 and [1]), which has the advantage that it exists in all eleven official European Union languages. Unlike other approaches to measure cross-language document similarity ([4], [10]), our own approach does not require language pair-specific linguistic data because EUROVOC acts as a conceptual interlingua. On the other hand, our own approach cannot be extended to languages other than the ones covered by the thesaurus used.

1.2 Motivation to Calculate the Semantic Similarity between Documents

Who is interested in the automatically calculated semantic similarity between documents written in different languages? What is it good for? Our own motivation for carrying out this work was to help users in the working environment of international organisations such as the European Commission to find their way through large multilingual document collections. One of the functionalities we considered to be useful is the capacity of showing users a ranked list of documents that are similar to one they are interested in, even if these other texts are written in different languages. Another functionality is the one to allow users to navigate through a multilingual document collection, using a document map [2, 12]. Document similarity calculation is also an essential tool for the automatic classification of texts into given classes.

In addition to our own motivation, efforts are made to compile automatically a collection of parallel texts in order to gain statistical knowledge on texts and their translations [5, 10]. Assuming that the translation of a text is the most similar text for a given one, our similarity calculation tool can be used for this application, too.

1.3 How to Evaluate Automatic Similarity Calculation

It is a non-trivial question how to judge, even intuitively, document content similarity. Are a three-page text and its 20-line abstract more similar than two 3-page documents talking about a similar subject? Should text length play a role at all in document similarity calculation? And should document language be a factor? It seems intuitively obvious that a text and its high-quality translation should be very similar. However, translators and other people speaking two languages very well know that different languages express concepts differently and have different ambiguities so that it is not reasonable to assume a 100% identity between a text and its translation.

Due to the complexity of the issue, automatic similarity calculation is rather difficult to evaluate. Lacking other alternatives, we decided to use the successful spotting of text translations as an evaluation criterion: we assume that, looking at a large text collection, if our system identifies the translation of a document as the most similar document, it performs well (more on this in sections 3.1 and 3.3).

1.4 The JRC's Text Management System

The document similarity calculation tool is part of a larger system put together by the European Commission's (EC) *Joint Research Centre* (JRC) that should help to manage the *information overflow* and to cross the *language barrier*. The JRC's system has three main components: one component whose task it is to find and retrieve documents in a variety of languages which are potentially relevant for the user's interests [8]; a second component that analyses the retrieved documents and extracts various information aspects from them; and a third component that visualises and presents the textual information and the extracted meta-information in a variety of ways [2, 12].

We consider the tool for calculating the similarity between texts to be part of the second (document analysis) component. The results produced by this tool are required for the information visualisation task, as carried out by the third component.

2 Assignment of EUROVOC Thesaurus Descriptors to Texts

We assign EUROVOC descriptor terms automatically, using a statistical approach that uses a training text collection to which descriptor terms had been assigned manually. For the application presented here, the training corpus consists of 6636 English and Spanish texts from the European Parliament (EP). After the off-line training phase (2.2), the descriptors can be assigned rather quickly online (2.3). Before applying any statistical techniques, we pre-process both the training material and the documents to be indexed by lemmatising all words, marking up the most frequent multi-word terms with underscore (e.g. *human_right*) and defining a large list of stop words. Details of this work have been published recently [11] so that we will only summarise this step here. However, the assignment algorithm has been improved since and has been applied to a text collection other than the set of EP training texts, so that we would like to describe the new algorithm and to present the latest assignment results.

2.1 The EUROVOC Thesaurus

EUROVOC was developed by the EP and the European Commission's Publications Office (OPOCE), together with national organisations of the EU member states for usage as a controlled vocabulary to index large multilingual document collections manually. EUROVOC exists in all eleven official European Union languages. Version 3 [1], which we use, consists of 5933 descriptor terms that are hierarchically organised into 21 fields and, at the second level, into 127 micro-thesauri. The maximum depth is eight levels. In addition to the 5877 pairs of *broader terms* (BT) and *narrower terms* (NT), there are 2730 pairs of *related terms* (RT) linking descriptors not related hierarchically. EUROVOC has a wide coverage and contains descriptors from the fields of politics, law, economics, finance, social questions (including culture and religion), education, science, employment, transport, environment, agriculture, forestry and fisheries, foodstuffs, technology and research, energy, geography, organisations and more.

Due to its wide coverage, EUROVOC is useful to describe texts from very different fields, but with less than 6000 descriptors it is not very detailed. Our main reasons to choose this thesaurus over others were that EUROVOC exists in exact translations in all eleven official European Union languages and that we were given access to both the thesaurus and to two manually indexed training collections (one from the EP and one from OPOCE). Furthermore, EUROVOC is used by many national and international organisations so that our work is sure to meet the interest of several user groups.

2.2 Training Phase

As EUROVOC descriptors are usually rather long and complex expressions which are unlikely to occur in their exact formulation in the running text of documents, we achieve the assignment of the relevant descriptors by producing automatically, for each descriptor in each language, large lists of semantically and statistically associated words (more precisely: *lemmas*) which, when found in a new text, trigger the assignment of the descriptor. We refer to these associated words as *associates*. For instance, the descriptor #12360607 (English text: PROTECTION OF MINORITIES) has associates such as *racism, xenophobia, minority, protection, human_right, indigenous_people, ethnic_minority,* etc.

We identify these associated lemmas in several steps, exploiting a training collection of texts for which professional indexers from the EP have identified the most appropriate EUROVOC descriptors manually. First, we compile, for each descriptor, a list of all texts of the training collection that were manually indexed with this descriptor. We refer to these text collections as *meta-texts*. We then compare the lemma frequency list of each meta-text with the lemma frequency list of the whole training collection, using the log-likelihood test [3]. The result of this comparison is, for each EUROVOC descriptor in each thesaurus language, a list of key lemmas that are particularly characteristic for this descriptor (associates). In addition to the lemma, a *keyness* value gives information on the degree of relevance of each lemma for this descriptor. This procedure is described in more detail in [11] and [12].

2.3 Assignment Phase

During the assignment phase, the lemmas of a new text that is to be indexed with EUROVOC descriptors are compared to the associate lists of all EUROVOC descriptors of the text language. Our assumption is that, the more similar an associate list is to the list of lemmas of the text, the more appropriate the corresponding descriptor is for this text. The descriptors can then be ranked according to their appropriateness, as expressed by an automatically calculated score.

After trying out a variety of different algorithms to compare the text with the associate lists (TFIDF, Cosine, Okapi, and others), we identified the *Cosine* formula [7] as producing the best EUROVOC descriptor assignment results, i.e. the overlap between manually and automatically assigned descriptors was biggest. Experiments showed that a mixed formula, using TFIDF, Okapi and Cosine with varying weights, produces precision results 3% to 6% higher than those shown in Figure 1. However, we did not use this optimised formula because its calculation for new documents is computationally heavier and its results are harder to use for the following document similarity calculation step. Interestingly, the document comparison procedure described in section 3 produces better results when using input (assigned EUROVOC descriptors) produced with the Okapi formula [6]. This shows that, for the purpose of similarity calculation, the *consistency* of the EUROVOC descriptor assignment is more important than its actual precision.

The *Okapi* formula (1) considers the number of times a lemma is used as an associate for a descriptor (DF_l), the number of associates in the associate list of the descriptor (|*d*|), the average number of associates in all associate lists (M), the total number of EUROVOC descriptors (N) and the occurrence frequency of the lemma in the text ($TF_{l,t}$), according to the following formula, with *d* being the descriptor, *t* being the text and *l* being a lemma (associate).

$$Okapi_{t,d} = \sum_{l \in t \cap d} \log(\frac{N - DF_l}{DF_l}) \frac{TF_{l,t}}{TF_{l,t} + \frac{|d|}{M}} \qquad (1)$$

The *Cosine* formula (3) computes the cosine of the angle of two multi-dimensional vectors [7]. If the vectors are about the same, the angle is about zero, so that the cosine is close to one. In our case, we calculate the cosine of a text's lemma frequency list with the lists of the various EUROVOC descriptor associates and their keyness. The Cosine formula uses the term weighting formula TFIDF (2), with the term frequency $TF_{l,d}$ being the number of times an associate lemma occurs in the meta-text and the document frequency DF_l being the number of descriptors for which the lemma *l* is an associate. So, when DF_l is one (lemma appearing only in this one descriptor), the TFIDF value will be high and when DF_l is about *N* (lemma appearing in all the descriptors), the TFIDF value will be low.

$$TFIDF_{l,d} = TF_{l,d} .((\log_2 \frac{N}{DF_l}) + 1) \qquad (2)$$

$$COSINE(d,t) = \frac{\sum_{l \in d \cap t} TFIDF_{l,d} .TFIDF_{l,t}}{\sqrt{\left(\sum_{l \in d} TFIDF_{l,d}{}^2\right)\left(\sum_{l \in t} TFIDF_{l,t}{}^2\right)}} \qquad (3)$$

Figure 1 shows the EUROVOC descriptor assignment results achieved for all the 2432 English texts of our OPOCE collection for which descriptors had been assigned manually. While both the EP and OPOCE use EUROVOC to index the documents in their archives, the two organisations deal with different kinds of texts so that the document collections used for training and for testing are different. In the OPOCE test collection, 5210 different descriptors had been assigned manually (EP training collection: 5142), with an average of 5.21 descriptors per text (EP training collection: 6.59). As our system produces a ranked list of descriptors of user-definable length, precision and recall can be calculated for any number of automatically assigned descriptors. Figure 1 shows that the highest-scoring descriptor (rank 1, x-axis) assigned by our system had also been assigned manually in 53% of all cases (performance on training set: 84%). Had the descriptors been assigned arbitrarily, the success rate for rank 1 would have been 0.088 % (5.21/5933).

As the EUROVOC thesaurus is not a flat list of terms, the relationship between descriptor terms had to be considered in the evaluation. Among the automatically assigned descriptors that had not been chosen manually, those which are an RT, BT or NT to a manually chosen one are *better* results than those which have no recognised relationship with the manually chosen descriptors at all. In addition to the percentage of correctly found manually assigned terms, Figure 1 therefore also shows performance information including RTs, BTs and NTs. While the human indexers were given instructions not to assign both the BT and an NT of a relevant concept, our system exclusively follows the similarity criterion. Figure 1 shows thus that, in 63% of all documents, the highest-ranking automatically assigned descriptor was either manually assigned or it was a BT, NT or RT of a manually assigned descriptor (performance on EP training set: 87%).

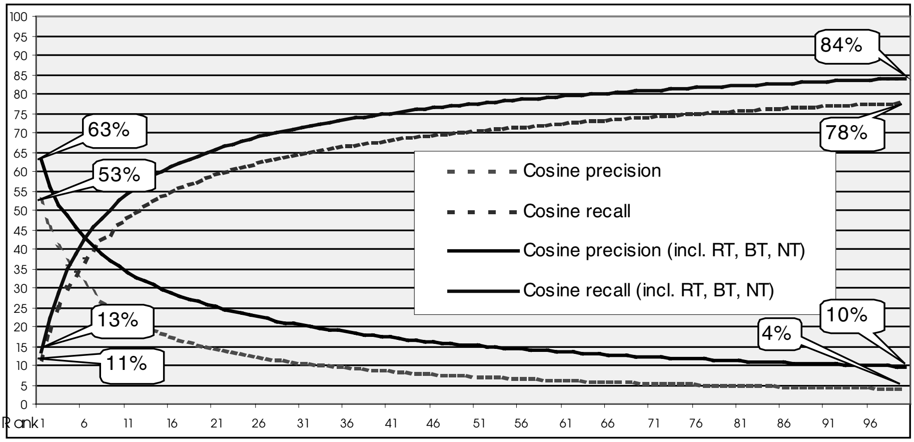


Fig. 1. EUROVOC descriptor assignment results for 2432 English OPOCE documents (Cosine formula), measuring the overlap between automatically and manually assigned descriptors for different ranks.

3 Document Similarity Calculation

The ranked lists of EUROVOC descriptors assigned to documents can be seen as an approximative representation of the document contents. Therefore, these descriptor lists can be used to calculate the similarity between documents. The more similar two descriptor lists are, the more similar we expect the two corresponding texts to be.

3.1 Translation Spotting vs. Similarity Calculation

As we mentioned in 1.3, the success rate with which translations of a text are identified as the most similar documents to a given one is the most obvious way of evaluating the similarity calculation performance automatically. The idea is that, within a document collection, the most similar document to a given one should be its translation. However, the task of identifying the translation of a given document is different from finding other similar documents. Firstly, translations are obviously written in a different language from the original text so that the search space is only half the search space of a bilingual document collection. Secondly, translations have a similar length and structure to the original document. These criteria can be used to optimise the performance of the translation spotting exercise. As our Spanish EP training texts used, on average, 13.5% more characters than their English equivalences (length factor LF = 1.135), our translation spotting formula assigned the highest similarity values to those texts that were 13.5% longer and punished texts with a different length in proportion to their deviation (see formula (5) in 3.2).

As the tasks of finding translations and finding other related documents, including those written in the same language and having different length, are different, we carried out two separate experiments: one which generally searches for similar documents (3.2), and one which searches specifically for translations (3.3). For both experiments, we used translation spotting as an evaluation criterion, i.e. the higher the score and ranking of the translation is in the list of automatically identified similar documents, the better are the results. As translation spotting is not our primary concern, but merely an evaluation criterion, we also calculated document similarity without considering the text length and without restricting the search space to the Spanish texts.

3.2 Calculating Document Similarity Based on Eurovoc Descriptor Lists

We calculated the similarity between documents by calculating the mutual distance between their automatically identified Eurovoc descriptor lists, using a cosine measure [7]. The documents which are the least distant are the ones which are the most similar. The first similarity formula (4) is a cosine on the vector space of the automatically assigned Eurovoc descriptors, with *d1* and *d2* being two documents, *e* being a Eurovoc descriptor, and $score_{e,d}$ being the Cosine or Okapi score of the Eurovoc descriptor for this document. The second formula (5) adds a length factor to the previous one, where *length* is the total number of characters in the document and LF is the language pair-specific length difference (1.135 for Spanish-English; see 3.1). Note that this Cosine formula uses automatically assigned Eurovoc descriptors as input and that these can be calculated with either the Cosine or the Okapi formula (see the discussion in 2.3). The results in Figure 2 are based on Eurovoc descriptors assigned by using the Okapi formula.

$$Sim(d_1,d_2) = \frac{\sum_{e \in d_1 \cap d_2} score_{e,d_1} . score_{e,d_2}}{\sqrt{\left(\sum_{e \in d_1} score_{e,d_1}{}^2\right)\left(\sum_{e \in d_2} score_{e,d_2}{}^2\right)}} \tag{4}$$

$$Simfl(d_1,d_2) = \frac{\min(length_{d1}, length_{d2})}{\max(length_{d1}, length_{d2})} Sim(d_1,d_2) \cdot \mathrm{LF} \tag{5}$$

We used a collection of 2995 English and 2995 Spanish Opoce texts (total of 5990 texts) that are translations of each other to test the performance of our system. The performance results are shown by the two lower lines in Figure 2. The x-axis shows the rank at which the translation was found, the ideal being that the translation was the highest-ranking document (rank 1; the most similar). We tested for 920 English documents whether their Spanish translation was found among all 5990 English and Spanish documents, and at what rank. Figure 2 shows that in 16% of all cases, the translation was found to be the most similar document (rank 1) to the original English document. Furthermore, it shows that, when using the length factor in the similarity calculation, the criterion of identifying the translation automatically is fulfilled much more successfully (55%). However, according to our own intuition, length should not play a role when solely looking for similar documents. When testing the system on the EP training collection, the results were 30% and 70%, respectively.

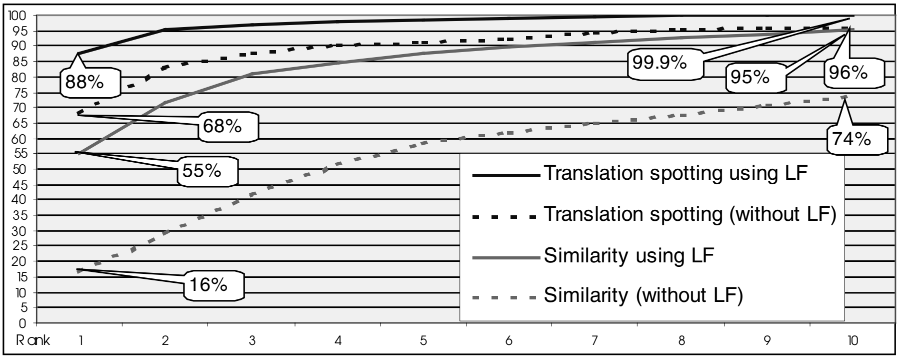


Fig. 2. Performance of the document similarity calculation (3.2) and translation spotting (3.3) tasks, using the automatically assigned descriptors produced with the Okapi formula as input.

The average similarity score of all translation document pairs is about 69% and is thus rather high (s.d. = 0.125; EP training collection: 77%, s.d. = 0.10). This value is slightly lower than the 78% (s.d.=0.09) produced by Landauer and Littman [4]. The fact that the translations *ranked* much lower (while still identified as being 69% similar to the original text) can be explained by the fact that the collection consists of many documents with very similar contents so that these similar documents outperformed the translations. For instance, many texts were resolutions taken by the EP on stopping nuclear tests, with small textual variations depending on the countries they discussed.

3.3 Spotting Spanish Translations of English Documents

The upper two lines of Figure 2 show the similarity calculation performance when the search space to find Spanish equivalents to English texts is restricted to the 2995 Spanish texts. Again, two different results are given: one for input produced considering the length factor and one for input not considering it. Both are produced using the Okapi Eurovoc descriptor assignment data as input. Applying the length filter again produces considerably better results than when not considering it (88% vs. 68% for rank 1). For the EP training data, the numbers were 93% and 91%.

The translation spotting precision (Spanish translations of English texts found) is much higher than the precision of the similarity calculation task. Presumably, this is not only due to the fact that the search space is halved (only the 2995 Spanish texts were considered as translation candidates). It is likely that our system is also slightly biased towards identifying similar documents in the same language as the original text because the likelihood of assignment of some descriptors may differ from one language to the other.

3.4 Implementation Details

The current system is implemented using mainly Perl, CGI and a relational database management system (RDBMS; either Oracle or MySQL) and runs on Unix or Windows/NT. The lemmatiser used is Lernout & Hauspie's *IntelliScope Search Enhan-*

cer. The tool to identify the associate lists for each descriptor is a customised version of the keyword identification functionality of Mike Scott's *WordSmith Tools* [9].

4 Limitations of This Method

As with any other automatically trained system, the performance depends heavily on the quantity and quality of the training data. For our current system, we have used training data received from the EP and applied it to texts of a different nature, which we received from OPOCE. As the EP texts do not cover all domains covered by EUROVOC, we do not have enough training data for all EUROVOC descriptors. Furthermore, the sublanguage used in EP texts is rather specific. We therefore expect better coverage (more associates for more descriptors) and better results when adding the OPOCE texts to our training data.

The EUROVOC thesaurus covers a wide range of domains (see 2.1), but it is not very detailed. Mapping document contents to such a relatively coarse knowledge structure means loosing some information when dealing with texts from very specific domains such as highly scientific texts. However, as it is our intention to apply this system to general Commission-related documents and to an automatically gathered collection of online newspaper articles, the detail of EUROVOC should be sufficient.

5 Planned Work

Our calculation of document similarity depends on the quality of the EUROVOC descriptor assignment results. We believe that we can achieve better results by improving text normalisation and data cleaning, by experimenting with various parameters, and by using additional training data from OPOCE. Once the process has been optimised for the languages English, Spanish and German, for which the system has currently been trained, we intend to apply it to the remaining EU languages.

The English language knows that "The proof of the pudding is in the eating" so that the ultimate criterion to measure the success of our system will be customer satisfaction. Therefore the application will have to be incorporated in a working system, together with other tools for document gathering, text analysis and information visualisation applications.

References

1. Eurovoc (1995). *Thesaurus Eurovoc - Volume 2: Subject-Oriented Version*. Ed. 3/English Language. Annex to the index of the Official Journal of the EC. Luxembourg, Office for Official Publications of the European Communities.
http://europa.eu.int/celex/eurovoc
2. Hagman Johan, Domenico Perrotta, Ralf Steinberger & Aristide Varfis (2000). *Document Classification and Visualisation to Support the Investigation of Suspected Fraud*. Workshop on Machine Learning and Textual Information Access (MLTIA). Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2000), 12 pages. Lyon, September 2000.

3. Kilgariff, Adam (1996). *Which words are particularly characteristic of a text? A survey of statistical approaches*. Proceedings of the AISB Workshop on Language Engineering for Document Analysis and Recognition, Sussex, April 1996, pp. 33-40.
4. Landauer Thomas & Michael Littman (1991). *A statistical method for language-independent representation of the topical content of text segments*. In Proceedings of the Eleventh International Conference: Expert Systems and Their Applications, volume 8, pp. 77-85, Avignon, France, May 1991.
5. Resnik Philip (1999). *Mining the Web for Bilingual Text*. 37th Annual Meeting of the Association for Computational Linguistics (ACL'99), Maryland, June 1999.
6. Robertson, S. E., S. Walker, M. Hancock-Beaulieu & M. Gatford (1994). *Okapi in TREC-3*, Text Retrieval Conference TREC-3, U.S. National Institute of Standards and Technology, Gaithersburg, USA. NIST Special Publication 500-225, pp. 109-126.
7. Salton G. (1989). *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer*. Reading, Mass., Addison-Wesley
8. Scheer Stefan, Ralf Steinberger, Giovanni Valerio & Paul Henshaw (2000). *A Methodology to Retrieve, to Manage, to Classify and to Query Open Source Information - Results of the OSILIA Project*. JRC Technical Note No. I.01.016, 35 pages.
9. Scott, Michael (1999). *WordSmith Tools v.3.0*. Oxford University Press, Oxford, UK. www.liv.ac.uk/~ms2928/wordsmith
10. Smith Noah (2001). *Detection of Translational Equivalence*. Unpublished Undergraduate Honours Thesis. University of Maryland, College Park, Maryland, USA.
11. Steinberger Ralf (2001). *Cross-lingual Keyword Assignment*. Proceedings of the XVII Conference of the Spanish Society for Natural Language Processing (SEPLN'2001), Procesamiento del Lenguaje Natural, Revista No. 27, pp. 273-280. Jaén, Spain.
12. Steinberger Ralf, Johan Hagman & Stefan Scheer (2000). *Using Thesauri for Information Extraction and for the Visualisation of Multilingual Document Collections*. Proceedings of the Workshop on Ontologies and Lexical Knowledge Bases (OntoLex'2000), 12 pages. Sozopol, Bulgaria, September 2000.

Empirical Formula for Testing Word Similarity and Its Application for Constructing a Word Frequency List

Pavel Makagonov[1] and Mikhail Alexandrov[2]

[1] Moscow Mayor's Directorate, Moscow City Government,
Novi Arbat 36, Moscow, 121205, Russia
makagonov@um.mos.ru
[2] Center for Computing Research, National Polytechnic Institute (IPN),
Av. Juan de Dios Batiz, C.P. 07738, DF, Mexico
dyner@cic.ipn.mx

Abstract. In many tasks of document categorization and clustering it is necessary to automatically learn a word frequency list from a corpus. However, morphological variations of words disturb the statistics when the program considers the words as mere letter strings. Thus it is important to identify the strings resulting from morphological variation of the same base meaning. Since using large morphological dictionaries has its well-known technical disadvantages, we propose a heuristic approximate method for such identification based on an empirical formula for testing the similarity of two words. We give a simple method for the determination of the formula parameters. The formula is based on the number of the coincident letters in the initial parts of the two words and the number of non-coincident letters in the final parts of these two words. An iterative algorithm constructs the word frequency list using common parts of all similar words. We give English and Spanish examples. The described technology is implemented in our system *Dictionary Designer*.

1 Introduction

In many tasks of document categorization and clustering it is necessary to automatically learn domain-oriented keyword dictionaries from a text corpus. The first step in this process is compilation of a word frequency list. However, morphological variations of words (e.g., *ask*, *asks*, *asked*, *asking*) disturb the statistics of word frequencies when the program considers the words as mere letter strings. To improve the statistics, it is important to identify the strings resulting from morphological variation of the same base meaning (e.g., *to ask*).

One of the possible approaches to this problem is the grammar approach relying on a morphological grammar (tables) and a large morphological dictionary. With this, on the basis of dictionaries of suffixes and inflectional endings of a given language the words are reduced to the standard form, i.e. nominative case singular for noun, indefinite infinitive for the verbs, etc. To compile a word frequency list, all words with the same standard form are collected together and their frequencies are summed

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 425–432, 2002.

up. There is a number of implementations of a morphological analizer; see, for example [Gelbukh, 1992].

Another morphology-based method is stemming: all words are reduced to some truncated form reflecting their invariant sense. After stemming, the frequencies of all words with the same stem are summed up, and we also obtain a word frequency list. There is a number of effective stemming algorithms; see, for example, the widely used Porter algorithm [Porter, 1980].

Obviously, in case of multilingual texts one should use the dictionaries of suffixes and inflectional endings for all languages used in these texts.

Such knowledge-rich approaches have their technological disadvantages, such as the need in expensive dictionaries, the difficulties in their maintenance, poor performance, and low robustness with respect to the changes in the language (new words). In addition, for a given language or subject domain there might not exist any available dictionaries of the necessary coverage.

We suggest a knowledge-poor heuristic algorithm based on empirical formula for testing similarity of word pairs. This algorithm reduces similar words to their common initial part (which can contain the root, suffixes, and even a part of inflective ending). Since our algorithm sometimes does not handle correctly some infrequent words, the resulting list of reduced words should be proofread by an expert. If the expert finds many mistakes in the algorithm's output, a slight change in the parameters of the empirical formula can fix the problem. Also, for the program to handle multilingual texts, the only things to be adjusted in the algorithm are the parameters of the formula. Such simplicity and flexibility are the main advantages of our algorithm.

2 Testing Similarity of Word Pairs

2.1 Simplest Formula for Word Similarity. Selection of Its Parameters

The key element for our consideration is the number of final letters differing for the two words. This is the number of letters that remains after removing from the words their maximal common initial substring. For example, for the words *asked* and *asking*, the maximal common initial substring is *ask-*, thus the differing final parts are *-ed* and *-ing* so that there is a total of $n = 5$ differing letters, $n_1 = 2$ in the first one and $n_2 = 3$ in the second one.

Our method is based on the following simple hypothesis about similarity of two words: Two words are similar if the relative number of differing final letters is less than some threshold, which linearly depends on the number of coincident letters in the two words. Specifically, we consider the two words similar if

$$n / s < a - b \times y \qquad (1)$$

where $n = n_1 + n_2$ is the total number of differing letters in two words, s is the total number of letters in two words, and y is the number of coincident letters in the initial part of the words. Thus, the smaller the part of non-coincident final letters, the more probable that these words are similar; the more the number of coincident initial letters the stronger this condition is. This hypothesis was successfully tested on Russian, English, and Spanish texts.

The parameters *a* and *b* depend on a specific language. They can be found using the following method. For a given language, several longest varying final parts of words are considered (such a part can contain only ending or a suffix and ending together, e.g., *-linessless* as in *timelinessless*). For each such a part, some short and some long word with this final part are considered. The more the number of considered examples the more accurate formula can be obtained.

This data set allows constructing a lineal function separating the common part of two similar words from their final parts. The initial approximation of the parameters of this function can be found with the least square method. Our experiments show that the coefficients *a* and *b* can be manually adjusted (starting from these initial approximations) for the grouping algorithm to produce better results.

Example 1 (English)

(1) We take the final part *–bility,* a pair of short words 'sense' and a 'sensi*bility'*, and a pair of long words 'distinguish' and 'distinguisha*bility*'. For these cases (1) gives

$$8/16 = a - 4b \quad \text{and} \quad 7/28 = a - 10b.$$

(2) We take the final part *–fully,* a pair of short words 'care' and 'carefully', and a pair of long words 'distrust' and 'distrust*fully*'. Here (1) gives:

$$5/13 = a - 4b \quad \text{and} \quad 5/21 = a - 8b$$

The method of a least square gives a system of 4 equations and 2 variables, with the solution a = 0.55, b=0.032. Thus the similarity test for *English* words consists in checking the inequality:

$$n / s < 0.55 - 0.032\, y$$

This empirical formula proved to be very close to the formula for testing *Russian* words obtained in a similar way using 10 pairs of Russian words:

$$n / s < 0.57 - 0.033\, y$$

Such a closeness of the empirical formulas for *English* and *Russian* means that mixed English-Russian texts containing large parts on every of these languages can be analyzed without separation, i.e. simultaneously. But such cases of seeming independence of a language need the methodology to be complicated and improved.

Example 2 (Spanish)

(1) We take the final part *–mente,* a pair of short words 'debo' and 'debida*mente'*, a pair of long words 'radical' and 'radical*mente*'. Formula (1) gives:

$$9/15 = a - 3b \quad \text{and} \quad 5/19 = a - 7b.$$

(2) We take the final part *–miento,* a pair of short words 'nació' and 'nacimiento', a pair of long words 'restablecer' and 'restableci*miento*'. Formula (1) gives:

$$7/15 = a - 4b \quad \text{and} \quad 9/27 = a - 8b$$

The method of a least square gives a system of 4 equations and 2 variables, with the solution: a = 0.68, b=0.046. After testing these values with the grouping algorithm and analyzing the results, we slightly corrected them: *a* = 0.69 and *b* = 0.044, which gives better results of the work of the algorithm. So the similarity test for *Spanish* words consists in checking the inequality:

$$n / s < 0.69 - 0.044\, y$$

In a similar way, empirical formulas for other European languages can be constructed. The general property of these languages is that the word's base (the morphologically invariant part) is located at the beginning (and not, say, at the end) of the word.

2.2 General Approach to Formula Construction

Formula (1) used in our algorithm resulted from a general hypothesis about similarity of two words: Two words are similar if the relative number of their differing final letters is less than some threshold depending on the number of initial coincident letters of the two words:

$$n / s < F(y) \tag{2}$$

where *n, s, y* are as defined in Section 2.1. We will try to find the model function *F* in a polynomial form:

$$F(y) = a + b_1 y + b_2 y^2 + ... + b_n y^n \tag{3}$$

To choose the optimal model complexity (the degree of the polynomial) and the optimum model parameters (the values of the coefficients) we used the method of grouped accounting of arguments (MGAA) [Ivahnenko, 1980]. In our experiments we used Russian texts. Following MGAA, we separated all words of the texts into a training set and a test set. Then we tested the formula (3) for the cases $n = 1, ..., 4$ using external criteria of regularity and unbiasedness. The criterion of regularity requires a minimum error on the test set. The criterion of unbiasedness requires closeness of models constructed using the training set and the test set. For a given *n*, we used the least square method to determine the best values of the coefficients *a* and b_i as described above.

We found that the case $n = 1$ proved to be the best. The model quality for the case $n = 2$ was almost the same, while the cases of $n = 3, 4$ proved to be significantly worse. Therefore, the lineal model

$$F(y) = a - by \quad (a, b > 0)$$

was adopted and used in the main algorithm.

We did not check whether the results obtained for Russian texts apply also to English and Spanish ones. Also, we did not try any functions other than (2). This will be the topic of our future work. However, one can see that even the results obtained so far are quite promising (see Section 3.2).

3 Main Algorithm

3.1 Constructing a Word Frequency List

The algorithm consists in the following steps. Initially, the text or a group of texts is transformed into a sequence of words. With every word in this sequence, a counter is associated and set to the initial value 1. Then the algorithm works as follows:

Step 1. All words are ordered alphabetically, literally equal words are joined together, and their counts are summed up (e.g., 3 occurrences of the string *ask* with counters 2, 3, and 1 are replaced with one occurrence with the counter 6).

Step 2. The similarity for each pair of adjacent words is tested according the criterion described above (namely, the 1st word is compared with the 2nd one, 3rd with the 4th, etc.). If a pair of words is similar then these two words are replaced with one new "word"—their common initial part, with the counter set to the sum of the counters of the two original words. If the list has an odd number of words, then the last word is compared with the immediately preceding one (or with the result of substitution of the last word pair).

Step 3. If no changes were made at the step 2, then the algorithm stops. Otherwise it is repeated from the step 1.

Example 1 (English). Suppose we have the following list of English words ordered alphabetically: *transform* (7), *transformed* (5), *transformation* (7), *translating* (6), *translator* (7), *transport* (11), *transported* (2). The digits in brackets are the counters. The following table illustrates the work of the algorithm.

Initial list	First pass	Second pass
transform (7)	transform (12)	transform (19)
transformed(5)	transformation (7)	translat (13)
transformation(7)	translating (6)	transport (13)
translating (6)	translator (7)	
translator (7)	transport (13)	
transport(11)		
transported (2)		

Example 2 (Spanish). Suppose we have the following list of Spanish words ordered alphabetically: *transformación* (7), *transformado* (5), *transformamos* (7), *traducción* (6), *traductor* (7), *transporte* (11), *transportado* (2). The digits in brackets mean the number of word's repetitions. The following table shows the work of the algorithm.

Initial list	First pass	Second pass
transformación (7)	transforma (12)	transforma (19)
transformado (5)	transformamos (7)	traduc (13)
transformamos (7)	traducción (6)	transport (13)
traducción (6)	traductor (7)	
traductor (7)	transport (13)	
transportado (2)		
transporte (11)		

3.2 Experimental Results

Of course, our algorithm gives false results making errors of two kinds: sometimes it does not join the words that in fact are morphological variants of the same base meaning, and sometimes it joins the words that are not. In our experiments, we compared the results of our algorithm with the opinion of the expert, see Table 1.

Table 1. Results of experiments with some domain-oriented texts

Language	Size of text corpus	Character of texts	False negative	False positive
English	215 Kb	Conf Proceedings	7%	5%
Spanish	133 Kb	Master thesis on Computer Science	6%	3%

It may be supposed that the number of such errors depends on the style of the document, its subject domain, etc. Slightly changing the parameters *a* and *b*, we can reduce the number of errors to a minimal value. So far we did not try this, so our results are very preliminary. However, 9%-12% error rate achieved in our experiments is quite acceptable because the program is intended to help the human user to compile a dictionary rather than to work in a totally unsupervised manner. Thus, it is better not to join some similar words (an error of the first kind) than to join non-similar ones (an error of the second kind) because an expert can join such words manually at the stage of proofreading. As usually, decreasing the error rate for one of these two kinds increases it for the other one. Thus, the expert should find an optimal level of error rate by fine-tuning the parameters *a* and *b* of the formulas. It should say that we do not think that the words with small differing final parts are always to be considered as the words with the same base meaning. It is just the expert who should make such a conclusion.

The word frequency list can be constructed using a stemming procedure based on simplified morphological analysis. For a brief overview of stemming, see a fundamental work [Manning, 1999]. As we mentioned above, Porter's stemmer is one of the most popular ones. In the future, we will compare our algorithm with Porter's one.

4 Implementation

The empirical algorithm presented above has been implemented in our system *Dictionary Designer* that belongs to the software family *Document Investigator*. This software family is currently used by the Moscow city administration, Russia, and is under experimental exploitation by the administration of Mexico City.

The system uses at its input either one document or a set of domain-oriented documents, producing as its output the word frequency list. This list is automatically transformed into the list of domain keywords on the basis of criteria of word selection (see, Appendix). Both lists consist of the words in a shortened form (similar to stems, though they might not be exactly what is called a stem in linguistics) as described above. The system allows the expert to change the parameters of empirical formula and the criteria of word selection. Also, he or she can correct the lists of words manually adding or removing words.

5 Conclusions

We have suggested a formula for deciding whether the two given words are similar, i.e., whether they are probable morphological variations of each other. This formula does not require any morphological dictionaries of the given language and can be constructed manually on several selected examples.

We have described an algorithm for creating word frequency lists based on the suggested formula. The behavior of the algorithm can be easily fine-tuned by adjusting its parameters that provides a satisfactory accuracy of results. A set of simple criteria allows transforming the frequency list of words into a list of domain-oriented keywords. The suggested technology has been implemented in user-oriented software.

References

1. Gelbukh, A. (1992): *Effective implementation of morphology model for an inflectional natural language.* "Automatic documentation and Mathematical Linguistics", Allerton Press, 26, N 1, pp. 22-31.
2. Ivahnenko, A. (1980): *Manual on typical algorithms of modelling.* “Technika” Publ., Kiev (in Russian).
3. Makagonov, P., Alexandrov, M., Sboychakov, K. (2000): *A toolkit for development of the domain-oriented dictionaries for structuring document flows.* In: H.A. Kiers et al (Eds.), *Data Analysis, Classification, and Related Methods*, Springer, 2000 (Studies in classification, data analysis, and knowledge organization), pp. 83-88.
4. Manning, D. C., Schutze, H. (1999): *Foundations of statistical natural language processing*. MIT Press.
5. Porter, M. (1980): *An algorithm for suffix stripping.* Program, 14, pp. 130-137.

Appendix

Selection of Keywords from a Word Frequency List

A word frequency list can be used for constructing domain-oriented list of keywords; we call such a list of selected words domain-oriented dictionary (DOD). For this, first of all it is necessary to have a domain-oriented set of texts and to construct the corresponding word frequency list. Then, it is necessary to use some criteria of keyword selection. Here we follow the methodology described in [Makagonov, 2000].

Criterion 1. Only those words W are included in the DOD for which $F_{Dom}(W) >> F_{Com}(W)$, namely, $F_{Dom}(W)>k\times F_{Com}(W)$. Here $F_{Dom}(W)$ and $F_{Com}(W)$ are the frequencies of the word W in the domain texts and in the general mixture texts, respectively. The coefficient k is determined after additional investigation. Its value is related with the statistical estimation of the mean error in the measuring of the frequencies due to a limited size of the sample texts. A good default value is 2.

To formulate the second criterion, we will need the notion of the Gini index $G_{\mathbf{T}}(W)$ for a given word W relative to a set of texts $\mathbf{T} = \{T_i\}$. Let n_i be the number of occurrences of the word W in the text T_i. Let us assume that n_i are arranged in ascending order; otherwise the texts are to be re-numbered correspondingly. For each i, let N_i be the accumulated number of occurrences: $N_i = \sum_{j\leq i} n_j$. Obviously, for a uniform distribution of the numbers of occurrences n_i = const, N_i represent a straight line: N_i = const × i. In all other cases, $N_i \leq$ const × i. The Gini index $G(W) = G_{\mathbf{T}}(W)$ for a given word W relative to a set of texts **T** is then determined as the relative difference between the area under the chart for N_i and under the uniform chart (straight line) const × i. Namely, $G(W) = 1 - \frac{1}{S}\sum_i N_i$, where S is the area under the uniform chart const × i, $S = \frac{1}{2}\left(|\mathbf{T}| + 1\right)\sum_i n_i$, where $|\mathbf{T}|$ is the total number of the texts and $\sum_i n_i$ is the total number of occurrences of the word W in all texts of **T**.

Criterion 2. Only those words W are included in the DOD for which the Gini index is between two fixed thresholds, low G_L and high G_H: $G_L < G(W) < G_H$. Their values are fixed empirically by an expert, good default values being 0.8 and 1, respectively.

The application of this criterion requires the number of values $n_i \neq 0$ (i.e., the number of the texts in **T** that contain the word W) not to be too small. On the other hand, if the DOD being constructed is to be later subdivided into sub-DODs, then the word W should not occur in all (or too many) of the texts of **T**. Namely, to obtain not less than 5 to 10 sub-DODs, each word should occur in not more than approximately 10% to 20% of the texts. Thus, for interesting DODs, the following criterion holds:

Criterion 3. Only those words W are included in the DOD for which the number N of texts in which they occur, $N = |\{i: n_i \neq 0\}|$, is between two fixed thresholds: $N_L < N < N_H$.

All these three criteria are used in the framework of a dialog with an expert. If the expert estimates the number of words in the resulting DOD as too large, the parameters of the three criteria should be adjusted to be more restrictive and the whole procedure repeated.

AutoMarkup: A Tool for Automatically Marking up Text Documents

Shazia Akhtar, Ronan G. Reilly, and John Dunnion

Smart Media Institute, Department of Computer Science, University College Dublin,
Belfield, Dublin 4, Ireland
{Shazia.Akhtar,Ronan.Reilly,John.Dunnion}@ucd.ie

Abstract. In this paper we present a novel system that can automatically mark up text documents into XML. The system uses the Self-Organizing Map (SOM) algorithm to organize marked documents on a map so that similar documents are placed on nearby locations. Then by using the inductive learning algorithm C5, it automatically generates and applies the markup rules from the nearest SOM neighbours of an unmarked document. The system is adaptive in nature and learns from errors in the automatically marked-up document to improve accuracy. The automatically marked-up documents are again arranged on the SOM.

1 Introduction

The dramatic growth of the World Wide Web with the availability of large collections of textual resources in electronic form has created a need for intelligent text processing. Extensible Markup Language XML was developed to address the need of electronic publishing and intelligent document management. Its power goes beyond the current Hypertext Markup Language and intelligent document management. Its power goes beyond the functionality of Hypertext Markup Language (HTML) for example it makes explicit the content and structure of the documents to make them easier to identify and retrieve. XML provides key features such as extensibility, validation and structure and is considered a complete solution for content management and electronic publishing. Despite the widespread adoption and popularity of XML, it is still a significant challenge to automatically markup documents in XML. Automatic XML markup is therefore currently a major research issue and many projects are involved in such research, as manual XML markup of documents is tedious and expensive. However most systems that have been developed are limited to certain domains and require a considerable amount of human intervention. There is as yet no tool available to solve the hard problem. In addressing this need we present a novel system that automatically marks up text documents into XML by using the Self-Organizing Map (SOM) algorithm (Kohonen, 1997) and an inductive learning algorithm C5 (Quinlan, 1993, 2000).

2 System Overview

Our system has two phases. The first phase of the system deals with the formation of a map of valid XML documents by using the SOM algorithm. In the second phase the

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 433–435, 2002.

system automatically learns and applies rules from the nearest SOM neighbours of a new unmarked document. The system learns from markup errors of the automatically marked up document and improves the markup. These two phases of the system are independently implemented and our intention is to combine the two phases to form a hybrid system.

Phase 2 of the system dealing with the automatic markup of documents is shown in Figure 1. It comprises two main modules, a Rule Learner and a Markup module. The first module learns classifiers by using the machine-learning algorithm C5. This module processes a set of pre-tagged valid XML documents.

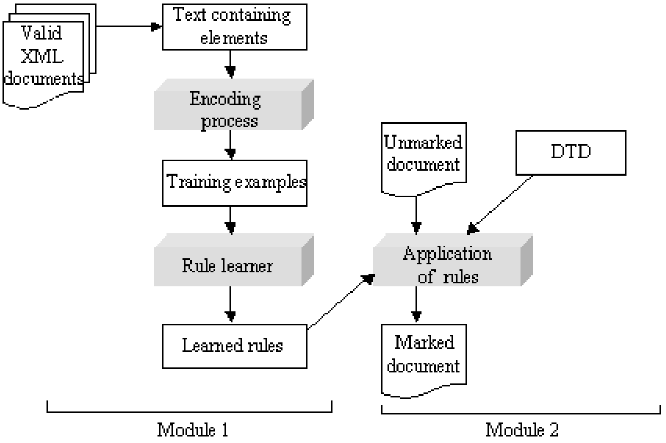
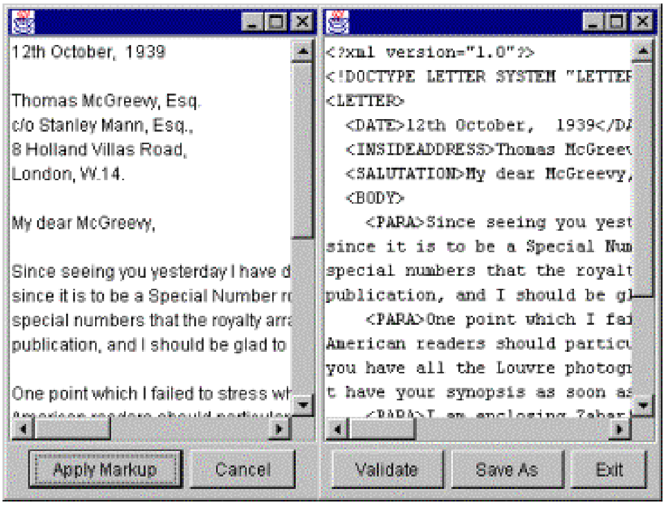


Fig. 1. Process of automatic markup. (a) Rule Learner (b) Markup

All documents in the set should be from a specific domain and conform to a single *Document Type Definition* (DTD). The system automatically gathers the *training examples* from the set of documents. Each *instance* corresponds to a text-containing *element* of the marked collection of documents. Instances are encoded using a fixed width feature vector. We have used twenty-two features such as word count and character count, in our experiments. All the encoded instances form a training set. Rules are learned when the training set is input to the C5 classifier. The second module deals with the markup of a new un-marked document from the same domain. The markup is obtained automatically by applying the generated rules and the rules of the DTD to an unmarked document. The DTD provides us with a set of rules using a number of operators for sequence elements (‘,’), repeated elements (‘+’), optional elements (‘?), and alternatives for recognizing the logical structure of a document. In this process, the unmarked document is chunked into pieces of text by using delimiters such as blank lines. By applying the rules of the DTD and the learned rules automatically generated by the system. The automatically marked up document is also valid XML.

We have used documents from a few different and simple domains using simple DTDs as an initial test bed for our experiments. One example is the automatic markup of letters from the MacGreevy archive (Schreibman, 1998). We have used valid marked-up letters from this archive to learn classifiers. By using the learned classifiers, unmarked letters from the same domain are marked up (see Figure 2). In an earlier version of this system we worked with well-formed documents comprising letters from the MacGreevy archive. We tested it on the elements of about 20 letters and achieved 94% accuracy. The accuracy rate is calculated by considering the correctly marked up elements as a percentage of the total number of elements of the tested letters. We are currently working with valid documents and hope to achieve higher accuracy.



12th October, 1939

Thomas McGreevy, Esq.
c/o Stanley Mann, Esq.,
8 Holland Villas Road,
London, W.14.

My dear McGreevy,

Since seeing you yesterday I have d
since it is to be a Special Number r
special numbers that the royalty arr
publication, and I should be glad to

One point which I failed to stress wh

```
<?xml version="1.0"?>
<!DOCTYPE LETTER SYSTEM "LETTEF
<LETTER>
  <DATE>12th October,  1939</D
  <INSIDEADDRESS>Thomas McGree
  <SALUTATION>My dear McGreevy,
  <BODY>
    <PARA>Since seeing you yest
since it is to be a Special Num
special numbers that the royalt
publication, and I should be gl
    <PARA>One point which I fai
American readers should partic
you have all the Louvre photog
t have your synopsis as soon a
```

Fig. 2. Unmarked letter and valid XML markup of the same letter produced by the system

References

1. Kohonen, T. (1997). *Self-Organizing Maps.* Springer Series in Information Science, Berlin, Heidelberg, New York.
2. Quinlan, J. R. (1993). C4.5: *Programs For Machine Learning.* Morgan Kauffman Publishers, San Mateo, Calif.
3. Quinlan, J. R. (2000). *Data Mining Tools See5 and C5.0.* [http://www.rulequest.com/see5-info.html]
4. Schreibman, S. (1998). *The MacGreevy Archive.* [http://www.ucd.ie/~cosei/archive.htm]

Identification of Recurrent Patterns to Extract Definitory Contexts

Gerardo Sierra and Rodrigo Alarcón

Instituto de Ingeniería, UNAM. México 04510, D. F.
{gsm,ralm}@pumas.iingen.unam.mx

Abstract. In order to develop an engine to extract likely definitory contexts in specialised corpus, we describe the acquisition and classification of an inventory of recurrent patterns on Disasters texts.

1 Introduction

Among different automatic works on terminology, one concerns concept extraction, i.e., term and definition extraction from specialised corpora. Our purpose in this paper relies on the need to get the corresponding lexical set of a domain, as a part of a specialised dictionary to get terms from concept descriptions.

From a computational linguistics point of view, specifically related to information extraction, terminology uses statistical methods and rule-based methods [1] in order to extract terms from specialised texts. Furthermore, terminology needs to identify the corresponding definitions of a specific term. Often, when an author introduces a new term, which is not well known to the readers, he/she uses a set of syntactic and typographic patterns to give the definition. We name here "definitory context" to the structure consisting of the term and the definition given in a specialised text.

In order to develop a tool capable of extracting definitory contexts automatically, an inventory of recurrent patterns used by authors to introduce concepts is necessary, as well as a computational linguistic technique capable to identify concepts in specialised texts from such inventory. Some efforts to the automatic identification of definitions from specialised corpus have been documented [2]. However, there is a lack of further surveys to identify term definitions.

2 Pattern Acquisition and Classification

There is a variety of ways in which each author introduces new concepts in a text. However, there are common sequences of syntactic and typographic patterns. For the acquisition of an inventory that can show a variety of those patterns, we chose texts on Disasters, provided by the *Instituto de Ingeniería*, UNAM.

This selection relies in the fact that those texts presented a conceptual frame for Disasters research. Therefore, the correspondent terminology is introduced. This fact facilitated the visual identification of the definitory contexts.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 436–438, 2002.

In order to facilitate patterns identification, in a first state we delimitated a definitory context for those structures integrated by a term (T) and its definition (D), where both parts belong to the same paragraph. Then we will attempt to analyse more complex forms, like definitory contexts where a term is not next to its definition.

We classified patterns in 4 groups. This groups go from simplex to complex forms: typographic, syntactic, mixed, and compound. This classification aims to facilitate our patterns study, as well as the analysis systematisation. In all cases of Disasters texts, terms are formed by a nominal phrase, a prepositional phrase, a noun, or both nominal and prepositional phrase. Verbal phrases were not considered, even though there is the possibility of finding terms formed by this kind of structures, like in some terminologies [3]. The complete relation of Disaster Patterns can be found at http://iling.iingen.unam.mx/patrones.

Typographic patterns contain some text format factors to emphasise either the term, the definitions, or both, without any verbal predication. The most common typographic forms are words in capital letters, bold, italic, underlined, or those introduced by a bullet or with punctuation signs. As punctuation signs substitute verbal predications, the term appears in first place, then a punctuation sign or a line break mark, and finally the definition. We found 4 different typographic patterns, in a total of 13 cases.

Syntactic patterns present a syntactic predication in their form, without any kind of typographic characters. The structure of these patterns is classified in three different ways. Each one consists of a term (T), a definition (D) and a pragmatic (P_1) or verbal predication (P_2). Therefore, the formula for syntactic patterns is:

$$P_1 + T + D; \quad T + P_2 + D; \quad P_1 + T + P_2 + D$$

We found 21 cases of syntactic patterns.

Mixed patterns combine both previous characteristics. So, their structure is represented by any typographic characteristic as well as by some of the predicative forms. We found a total of 32 different mixed patterns, due to the different kind of predications in a set of 45 cases.

Finally, there are two different types of compound patterns. In the first one, the same definitory context involves two or more different terms. In the second one, the definition of a term includes another definitory context for the introduction of a new concept. We found 11 cases of mixed patterns.

3 Conclusions

We integrated an inventory of 67 different patterns on Disasters texts in a total set of 90 occurrences: 4 typographic, 20 syntactic, 32 mixed and 11 compound patterns. Future work will consist in an exhaustive comparison of these patterns with others found in different texts of diverse areas: Linguistics, Engineering and Physics. For this purpose, a database for each subject is carried on, trying to concentrate a more extended inventory of recurrent patterns in definitory contexts.

By using these databases, the inventory could be synthesized trying to obtain a huge number of patterns and all probable variations. Therefore, the final inventory will consist of a huge numbers of structural possibilities that will be used in other texts to obtain an engine capable to extract definitory contexts in specialized corpus.

This engine requires the inventory as a knowledge base, presented here, as well as an information extraction technique, still in progress.

References

1. Cabré, T., Estopà, R., Vivaldi, J.: Automatic term detection. A review of current systems. In *Natural Language Processing*. Amsterdam, John Benjamin's (2001).
2. Pearson, J.: Terms in context. Amsterdam, John Benjamin's (1998).
3. Cardero, A.: *El procesamiento de una terminología. Referencia especial a la terminología de control de satélites en el área de las telecomunicaciones en México*. PhD Thesis, UNAM, México (2001).

Appendix: Examples of Patterns

The following table shows different kinds of patterns and a definitory context for each one.

	TYPOGRAPHIC PATTERNS
T (bullet) + "," + D	+Lesiones, perturbación causada en los órganos del cuerpo, como contusión [...]
T (italics capital) + D	*IMPACTOS AGREGADOS PRODUCTIVOS* Los que impactan a los sistemas de [...]
	SYNTACTIC PATTERNS
P_1 + T + D	Se entiende por evacuación el desalojo rápido o paulatino de [...]
T + P_2 + D	La evaluación de la vulnerabilidad se refiere a la estimación de la susceptibilidad al daño de [...]
P_1 + T + P_2 + D	En este sentido, el estado de un sistema se define como una característica global [...]
	MIXED PATTERNS
T (bullet) + "," es + D	+ Daño nulo, es cuando el elemento no quedó afectado por los impactos.
T (italics bullet) + se identifican como + D	+*Las tareas* se identifican como las partes de un subprograma, conforme con [...]
de acuerdo con + (...) + T (italics) + se concibe como + D	De acuerdo con el enfoque integral expuesto, el *sistema de gestión* se concibe como una organización, cuyo [...]
	COMPOUND PATTERNS
T1 + y + T2 + La primera + (...) + trata de + D1 + "." + La segunda se caracteriza por + D2	A su vez, en el proceso de gestión se distinguen dos modalidades polares y complementarias: la gestión correctiva y la planificada. La primera modalidad trata de mantener al objeto conducido en [...] La segunda, se caracteriza por preestablecer un estado futuro [...]
Se considera + T1 + D1 (T2 + D2)	Se considera calamidad todo acontecimiento que pueda impactar el sistema afectable, en este caso la central y sus alrededores, [...] y transformar su estado normal o deficiente en [...]

Specification Marks Method: Design and Implementation

Sonia Vázquez, Mª Carmen Calle, Susana Soler, and Andrés Montoyo

Research Group of Language Processing and Information Systems
Department of Software and Computing Systems
University of Alicante, Alicante, Spain
montoyo@dlsi.ua.es

Abstract. This paper presents the design and implementation of an interface web to resolve lexical ambiguity of nouns in English texts, using hierarchy organization of WordNet. This interface web is based on the Specification Marks Method [1]. It is an unsupervised knowledge based method and consists basically of the automatic sense-disambiguating of nouns that appear within the context of a sentence and whose different possible senses are quite related.

1 Introduction

Word sense disambiguation (WSD) is an open research field in Natural Language Processing (NLP). The task WSD resolves the lexical ambiguity, therefore, WSD endows a given word with a specific meaning that distinguishes it from all of the other possible meanings that the word might have in other contexts.

The method described here requires the knowledge of how many of the words in the context are grouped around a Specification Mark, which is similar to a semantic class in the WordNet taxonomy. The word-sense in the sub-hierarchy that contains the greatest number of words for the corresponding Specification Mark will be chosen for the sense-disambiguating of a noun in a given context. For a better understanding of the method before described, Montoyo in [1, 2] describes with detail the method and a series of improving incorporated to the disambiguation process. I want to clarify that we don't deep into any details of the Specification Marks Method and theoretical developments employed.

We present at this paper the functionality and the user interface that shows how this method resolves lexical ambiguity for English nouns using the notion of Specification Marks and employing the noun taxonomy of the WordNet lexical knowledge base.

2 Web Interface

In order to use the WSD module in Internet it is necessary the design and implementation of an interface web. The architecture employed in developing this interface web is shown in Fig. 1.

A. Gelbukh (Ed.): CICLing 2002, LNCS 2276, pp. 439–441, 2002.

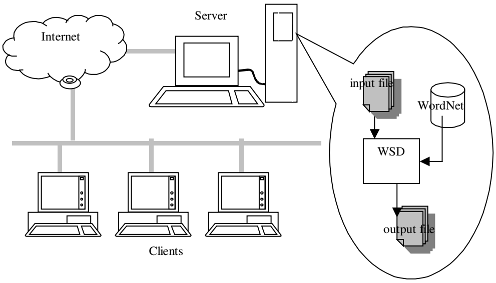


Fig. 1. Architecture of interface web with WSD module

First, users introduce a group of words in the interface web. This group of words is sent to the server web for starting the WSD module. After, a process checks the information introduced by the user in the server web and endows it the appropriate structure for their handling for the WSD module. Once obtained the file with the input data properly formatted, it will be at the same time the input data to the WSD process that carries out the disambiguation of the text based in Specification Marks method and using the lexical database WordNet. Finally, when the WSD process concludes another process formats the information disambiguated and the server web sends this information to the interface web in order to show it to the user.

This interface web is accessible from Internet at the URL[1] using any navigator. It is illustrate in the figure 2. The user interface offers the operations followed:

Run WSD Process. The command button WSD allows one to run the lexical ambiguity algorithm based in [2]. The input to the algorithm is English nouns that appear in the left text window of the interface, named *Nouns to Disambiguate*. The result of the disambiguation is shown in the right text window, named *Senses of WordNet*, in four columns. The first column is a set of synsets provided by WordNet. The second column is the noun to disambiguate. The third one is the number sense selected among all the possible senses offered by WordNet. And finally, the fourth column is the gloss associated to the selected sense of WordNet.

Clear Process. The user clicks on this command button to delete the information that appears in both text windows.

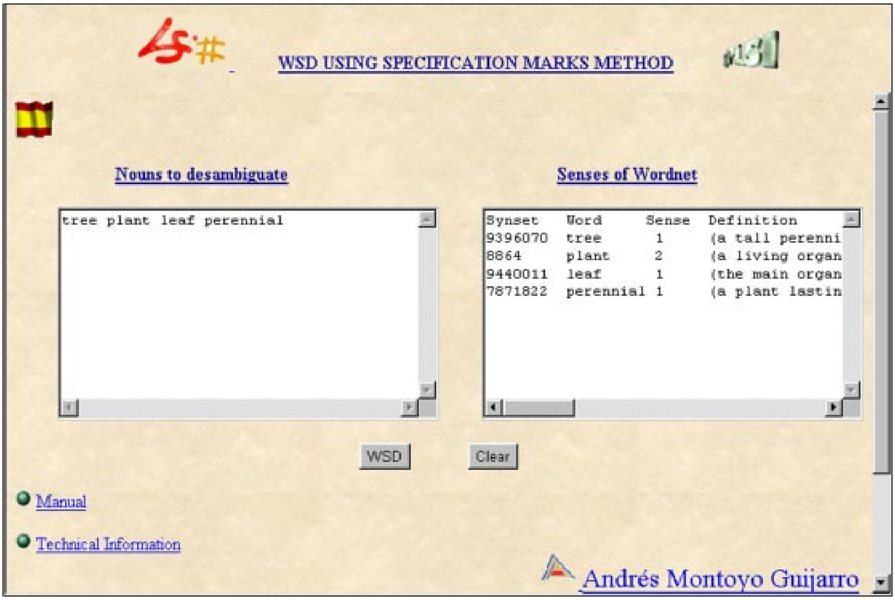
Sometimes one or more words cannot be disambiguating. You can see this kind of words in the right text window preceded by the symbol asterisk (*). In this case it is shown all the possible senses of the word. You can also see a manual of this interface in its URL as well as some references to another papers, which discuss about disambiguation.

3 Conclusion

This paper presents the design and implementation of an user interface to resolve lexical ambiguity for English nouns, which is based on Specification Marks between

[1] http://gplsi.dlsi.ua.es/wsd

words, using word sense tags from WordNet. The types of resources accessible via the Internet are growing at an astounding rate, therefore we have chosen Internet technologies to build this interface web. It provides to users, who are interested in word sense disambiguation, a tool for resolving the lexical ambiguity of nouns in their English texts.



```
tree plant leaf perennial
```

```
Synset   Word      Sense  Definition
9396070  tree        1    (a tall perenni
8864     plant       2    (a living organ
9440011  leaf        1    (the main organ
7871822  perennial 1      (a plant lastin
```

Fig. 2. User Interface.

References

1. Andrés Montoyo and Manuel Palomar. Word Sense Disambiguation with Specification Marks and in Unrestricted Texts. *In Proceedings of 11th International Workshop on Database and Expert Systems Applications*, pages 103-107, Greenwich, London, UK, 2000.
2. Andrés Montoyo and Manuel Palomar. WSD Algorithm applied to a NLP System. In Proceedings of 5th International Conference on Applications of Natural Language to Information Systems (NLDB-2000). Versailles, France, 2000.

Author Index