

Amy Neustein · Judith A. Markowitz
Editors

Where Humans Meet Machines

Innovative Solutions for Knotty Natural-
Language Problems

 Springer

Where Humans Meet Machines

Amy Neustein · Judith A. Markowitz
Editors

Where Humans Meet Machines

Innovative Solutions for Knotty
Natural-Language Problems

Editors

Amy Neustein
Linguistic Technology Systems
Fort Lee, New Jersey
USA

Judith A. Markowitz
J. Markowitz Consultants
Chicago, Illinois
USA

ISBN 978-1-4614-6933-9

ISBN 978-1-4614-6934-6 (eBook)

DOI 10.1007/978-1-4614-6934-6

Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013940919

© Springer Science+Business Media New York 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Where Humans Meet Machines: Innovative Solutions for Knotty Natural-Language Problems brings humans and machines closer together by showing how linguistic complexities that confound the speech systems of today can be handled effectively by sophisticated natural-language technology. Some of the most vexing natural-language problems that are addressed in this book entail recognizing and processing idiomatic expressions, understanding metaphors, matching an anaphor correctly with its antecedent, performing word-sense disambiguation, and handling out-of-vocabulary words and phrases. This 14-chapter anthology consists of contributions from industry scientists and from academicians working at major universities in North America and Europe. They include researchers who have played a central role in DARPA-funded programs and developers who craft real-world solutions for corporations. These contributing authors analyze the role of natural-language technology in the global marketplace; they explore the need for natural-language mapping tools that can cull important data from the vast array of social media postings; they describe innovative Natural Language Processing (NLP) methods and applications; and they offer NLP tools for physicians, educators, and translators. Their contributions constitute diverse and multifaceted solutions for the knotty natural-language problems that permeate everyday human communications.

This book is divided into three sections.

The first section takes a critical look at specific aspects of natural-language technology in the global marketplace.

The section opens with a thought-provoking discussion of present-day demands for high-throughput, low-latency NLP systems. The authors demonstrate that potential difficulties can be optimally addressed by “a unified, open-system architecture that allows for rapid development, deployment, and interoperation of NLP engines from different vendors.” Focusing on the free, open-source Unstructured Information Management Architecture (UIMA) used in conjunction with IBM’s Translingual Automatic Language Exploration System (TALES), the authors show how “English-speaking users can monitor English and foreign-language video and Web news media in near real-time as well as search over stored multimedia content.”

The following chapter examines ways to design NLP mapping tools that can exploit rich resources of data found in social-media postings. The authors analyze linguistic features of medical postings and juxtapose them with features of clinical narratives (e.g., discharge summaries, chart reviews, and operative reports) and biomedical literature, for which there exist tools for performing information extraction. They reveal how mapping tools that can extract critical data from clinical narratives or biomedical literature are wholly inadequate for data extraction from social-media. The authors then propose novel, yet infinitely practical, ways to improve mapping tools “so that the wealth of medical data that is embedded in medical social-media may be made available to health care providers and pharmaceutical companies, as well as to government-supported epidemiological agencies.”

In the subsequent chapter, two software engineers analyze the benefit of reducing human-translation costs for those who invest in commercial machine translation and for large, multilingual, and multicultural enterprises. Among their real-life examples is a company “with worldwide technical support in Germany that must support a new team in Brazil with real-time text chat” and a “hotelier with hotels in China who needs to integrate language into their world-renowned loyalty program.” In their comprehensive examination of this topic, the authors provide the reader with an incisive history of machine translation in the enterprise market showing “where it has succeeded and where it has failed.” They then extend the view forward to see “what the future looks like for an enterprise use of machine translation.”

The section concludes with a fascinating discussion of speech-enabled unified communications. The author highlights the European market, because of its varied mix of languages, pointing to the “growing need for innovative, multimodal natural language interfaces that can exploit the power of new devices and address the needs of an increasingly mobile world.” In his analysis of the creative and utilitarian solutions provided by speech-enabled unified communications, the author delves deeply into challenging real-world hurdles to implementation. He highlights ways to eliminate these obstacles by focusing on strategies for handling multilingualism and for overcoming unfavorable market practices, such as excessively high roaming fees.

The second section of the book explores creative ways of confronting thorny natural-language problems that have perplexed speech scientists and system developers for decades. In this section, the contributors present new research paradigms that employ advanced NLP methods and applications. They show how their approaches differ from state-of-the-art techniques and methods. Furthermore, regardless of the topic they address, they approach their work from a multilingual and multidisciplinary perspective, which ensures that their research findings can be readily applied across languages and discourse domains.

The section begins with a challenge to researchers working on word-sense disambiguation by calling for more qualitative analysis of disambiguation derived from an inter-disciplinary perspective. The author shows that “the difficulty of disambiguating a particular target word is a function of its information susceptibility, which depends on how the senses of the word were distinguished in the first place.”

This enables the author to demonstrate how word-sense disambiguation can be handled as “reverse engineering of lexicography, so that the use of knowledge sources as well as feature selection could be more informed with respect to individual words and their senses.” This results in an approach that supports lexically-sensitive combinations of algorithms and knowledge sources.

The chapter on word-sense disambiguation is followed by discussion of a distinctively-original approach to text summarization. The authors of this chapter present the compelling results of their study of summarization in a multilingual context using a discourse-centered approach. Diverging from state-of-the art summarization techniques, the authors treat a summary as “a-text-by-itself,” and opine that, “just like the original, it should preserve the qualities of being cohesive and coherent, even if it is made up of elementary pieces extracted from the original text and reassembled again.” They further show that a generic architecture can be used to define summarization systems for more than one language, which allows their system to operate in a language-independent fashion while, at the same time, permitting comparisons among and between languages. This novel methodology enabled the authors to “identify some very interesting correlations between quantitative data characterizing the training corpora and the evaluation results” obtained from their experiments.

Next, three Romanian scientists tackle the problem of out-of-vocabulary (OOV) words. OOV words have scuttled the performance of text-to-speech (TTS) synthesis systems since their inception. These authors approach this Sisyphean task with a dose of reality by acknowledging the many reasons for having to deal with OOV words, including use of technical terms, proper nouns, rare words that were not covered by the lexicon, and foreign words whose morphology has been partially adapted to the new language. In their chapter, the authors introduce “the basic architecture of a generic natural language processing module in TTS synthesis, proposing data-driven solutions to various tasks.” They further demonstrate the superiority of their architecture vis-à-vis state-of-the-art commercial TTS for handling OOV words and prosody modeling.

The ensuing chapter provides an insightful look at how to overcome the challenge posed by metaphoric expressions that occur “in a context unrestricted by domain, such as an ordinary conversation or a digression from a domain-specific context.” The author carefully describes how her Metaphor Analysis Program (MAP) processes novel and unconventional cross-modal metaphors using “an analysis of a metaphorically-used word in its literal sense and the role of this sense in structuring the topic of the metaphor.” The chapter is replete with vivid examples illustrating how the “problems of multiple metaphoric interpretations and the uncertainty of constraints on metaphoric coherence” may be addressed. Because it “relies on the kind of semantic analysis that models what humans are hypothesized to do when they extend a literal meaning to a metaphoric one” this chapter exemplifies the motif of the book. In short, this author delineates a possible path for bringing machines closer to humans!

The section is rounded out by a fascinating case study involving a tool for finding translations of idiomatic expressions across multiple languages. The authors

utilize TRANSSEARCH, a translation spotter and bilingual concordance available on the Web. They compare the translations identified by TRANSSEARCH to those returned by GOOGLE TRANSLATE showing that “by paying close attention to the queries made to the system, TRANSSEARCH can effectively identify a fair number of idiomatic expressions and their translations.”

The third section serves as the coda to this book. Here, the authors propose an array of innovative NLP tools for assisting physicians, educators, and translators.

The section begins with a study of argumentation-based dialog systems for medical training that “provide human-like mechanisms for computer–clinician interaction.” The authors employ an extended version of *arguEIRA*, a system based on the *ASPIC* argumentation engine, which “is able to detect anomalous patient responses using flexible reasoning processes and logical argumentation.” These components enable the system to actually argue with the clinician. Such solution-oriented debates are designed for expanding the clinician’s knowledge and, thereby, “changing the clinician’s behavior.” The authors reveal two additional objectives that maintain the utility of the system: “potentially enhancing the [clinician’s] acceptance of the system’s explanations” and providing “clinicians with simple mechanisms to discover through the training process if the knowledge base used by the explanation system should be updated or corrected.”

The chapter on argumentation-based dialog systems for medical training is followed by a comprehensive overview of dialog-based, intelligent-tutoring systems (ITSs) that simulate human–human tutoring. The authors describe them as “learning technologies that help learners develop mastery of difficult subject matter by holding conversations in natural language.” In their examination of basic issues in the design of dialog-based ITSs, the authors provide an analysis of human–human tutorial dialog and a discussion of the six major components of most dialog-based ITSs: input transformation, speech-act classification, learner modeling, dialog management, output rendering, and domain modeling. These discussions are followed by a fascinating look at how advanced, dialog-based ITSs are designed to address the “model learners’ emotional states in addition to their cognitive states” and, in some instances, to provoke changes in a user’s emotion state by strategically planting “confusion in the minds of learners to engender deeper modes of thinking.”

The NLP tool described in the subsequent chapter can be viewed as a boon to bilingual educators who work with hearing-impaired students. The author describes a software system called *TCAD* (Total Communication Animated Dictionary), which has been developed “to support the acquisition and retention of English vocabulary of pupils with hearing impairment in their bilingual education program.” *TCAD* employs visual aids and animated features (e.g., lip reading, finger spelling, sign, and animation) to optimize vocabulary acquisition and uses stories and games to enhance retention. Hearing-impaired students using *TCAD* have exhibited “an improved motivation and enthusiasm” for learning English, as evidenced by the fact that pupils were found to interact “with the system during break times and after school time.” The author astutely notes that the “*TCAD* tool has also provided a forum for social interaction” among pupils.

The next chapter continues the exploration of creative NLP tools for bilingual educators. It provides a fascinating study of a hybrid approach to automated rating of foreign-language proficiency using oral responses to test questions. This author describes English-language responses collected through Language Testing International's (LTI) Oral Proficiency Interviews using a Computer (OPIC). He explains "[i]n OPIC tests, a computer automatically asks questions from the candidate and the responses of the candidate are recorded and consequently rated." Lastly, although the test was done with adult learners of English, the results "may be readily extended to test other languages."

The last chapter on innovative NLP tools is also the final chapter of this anthology. It offers a comprehensive analysis of the effect of electronic tools on the human translator and explores how, in multilingual contexts, "machine translation can be integrated into translation-memory systems in the translation workflow." The chapter provides a rare glimpse of translation-memory; rare because these systems have traditionally stayed in the shadows. It considers how these "new technologies might affect the translation market and the activity of translation professionals." The chapter concludes with a discussion of tool usability and job satisfaction, recommending these topics for future research.

In assembling this anthology, the editors have witnessed the extraordinary commitment and zeal of natural-language researchers. Instead of being discouraged by the idiosyncrasies and quirks of human language, they have endeavored to find creative solutions that successfully address these challenges—solutions that untie the knots. In so doing, they bring us a giant step closer to where humans meet machines. This is no small undertaking, yet a most needed one if natural-language technology is to remain loyal to its primary goal of forging a positive connection between computers and human language.

Fort Lee, NJ
Chicago, IL

Amy Neustein
Judith A. Markowitz

Contents

Part I Natural-Language Technology and the Global Marketplace

| | | |
|----------|---|-----------|
| 1 | Making the Case for an Open, Unified System Architecture in Response to Rapid Developments in the Natural Language Industry: Translingual Automatic Language Exploration the System (TALES)..... | 3 |
| | Leiming Qian, Imed Zitouni, and John F. Pitrelli | |
| 2 | The Burgeoning of Medical Social-Media Postings and the Need for Improved Natural Language Mapping Tools | 27 |
| | Kerstin Denecke and Nazli Soltani | |
| 3 | Machine Translation: The Enterprise Point of View | 45 |
| | Brian Garr and Vadim Berman | |
| 4 | Speech-Enabled Unified Communications: Overcoming the Multilingual Challenges of the European Market | 59 |
| | Detlev Artelt and Hassan Sawaf | |

Part II Advanced NLP Methods and Applications

| | | |
|----------|---|------------|
| 5 | Exploiting Lexical Sensitivity in Performing Word Sense Disambiguation | 81 |
| | Oi Yee Kwong | |
| 6 | Summarizing Short Texts Through a Discourse-Centered Approach in a Multilingual Context..... | 109 |
| | Daniel Alexandru Anechitei, Dan Cristea, Ioannidis Dimosthenis, Eugen Ignat, Diman Karagiozov, Svetla Koeva, Mateusz Kopeć, and Cristina Vertan | |

| | | |
|--|--|-----|
| 7 | Handling Two Difficult Challenges for Text-to-Speech Synthesis Systems: Out-of-Vocabulary Words and Prosody: A Case Study in Romanian | 137 |
| | Tiberiu Boroş, Dan Ştefănescu, and Radu Ion | |
| 8 | MAP: An Abstraction-Based Metaphor Analysis Program for Overcoming Cross-Modal Challenges | 163 |
| | Sylvia Weber Russell | |
| 9 | Translation of Idiomatic Expressions Across Different Languages: A Study of the Effectiveness of TRANSSEARCH | 185 |
| | Stéphane Huet and Philippe Langlais | |
| Part III Innovative NLP Tools for Physicians, Educators and Translators | | |
| 10 | Argumentation-Based Dialogue Systems for Medical Training | 213 |
| | Adela Grando, Laura Moss, Gemma Bel-Enguix, M. Dolores Jiménez-López, and John Kinsella | |
| 11 | Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to-Human Tutoring | 233 |
| | Sidney D'Mello and Art Graesser | |
| 12 | TCAD: Vocabulary Acquisition Tool for Motivating Bilingual Pupils with Hearing Impairment in Learning English | 271 |
| | Santichai Wicha, Bernadette Sharp, Anthony S. Atkins, Pradorn Sureephong, and Nopasit Chakpitak | |
| 13 | A Hybrid Approach to Automated Rating of Foreign Language Proficiency Using Oral Test Responses | 285 |
| | Homayoon Beigi | |
| 14 | Multilingual Systems, Translation Technology and Their Impact on the Translator's Profession | 299 |
| | Carlos S.C. Teixeira | |
| | Editors Biographies | 315 |

Contributors

Daniel Alexandru Anechitei “Alexandru Ioan Cuza” University of Iași, Iași, Romania

Detlev Artelt aixvox GmbH, Aachen, Germany

Anthony S. Atkins Faculty of Computing, Engineering and Technology, Staffordshire University, Octagon Stafford, UK

Homayoon Beigi Recognition Technologies, Inc., Yorktown Heights, NY, USA

Gemma Bel-Enguix Laboratoire d’Informatique Fondamentale, CNRS-Aix-Marseille University, Marseille, France

Vadim Berman Linguasys, Boca Raton, FL, USA

Tiberiu Boroș Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

Nopasit Chakpitak College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand

Dan Cristea “Alexandru Ioan Cuza” University of Iași, Iași, Romania

Institute for Computer Science, Romanian Academy, Iași branch, Iași, Romania

Sidney D’Mello Department of Psychology and Computer Science, University of Notre Dame, Notre Dame, IN, USA

Kerstin Denecke Innovation Center Computer Assisted Surgery, University of Leipzig, Leipzig, Germany

Ioannidis Dimosthenis Atlantis Consulting SA, Thessaloniki, Greece

Brian Garr Linguasys, Boca Raton, FL, USA

Art Graesser Department of Psychology, Institute of Intelligent Systems, University of Memphis, Memphis, TN, USA

Department of Education, University of Oxford, Oxford, UK

Adela Grando Division of Biomedical Informatics, San Diego Super Computer Center 2.93E, University of California, San Diego, San Diego, CA, USA

Stephanie Huet Department of Computer Science, University of Avignon, Avignon, France

Eugen Ignat “Alexandru Ioan Cuza” University of Iași, Iași, Romania

Radu Ion Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

M. Dolores Jiménez-López Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, Spain

Diman Karagiozov Tetracom Interactive Solutions Ltd., Sofia, Bulgaria

John Kinsella Academic Unit of Anaesthesia, Pain, and Critical Care Medicine, School of Medicine, University of Glasgow, Glasgow, UK

Svetla Koeva Institute for Bulgarian Language, Bulgarian Academy of Sciences, Sofia, Bulgaria

Mateusz Kopeć Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Oi Yee Kwong Department of Chinese, Translation and Linguistics, City University of Hong Kong, Kowloon, Hong Kong

Philippe Langlais Department of Computer Science, University of Montreal, Montreal, QC, Canada

Laura Moss Department of Computing Science, University of Aberdeen, Aberdeen, UK

Academic Unit of Anaesthesia, Pain, and Critical Care Medicine, School of Medicine, University of Glasgow, Glasgow, UK

John F. Pitrelli IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

Leiming Qian IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

Sylvia Weber Russell Department of Computer Science, University of New Hampshire, Durham, NH, USA

Hassan Sawaf aixvox GmbH, Aachen, Germany

Bernadette Sharp Faculty of Computing, Engineering and Technology, Staffordshire University, Octagon Stafford, UK

Nazli Soltani University Medical Center Göttingen, Göttingen, Germany

Dan Ștefănescu Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

Pradorn Sureephong College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand

Carlos S.C. Teixeira Intercultural Studies Group, Universitat Rovira i Virgili, Tarragona, Spain

Cristina Vertan University of Hamburg, Hamburg, Germany

Santichai Wicha Faculty of Computing, Engineering and Technology, Staffordshire University, Octagon Stafford, UK

School of Information Technology, Mae Fah Luang University, Chiang Rai, Thailand

College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand

Imed Zitouni Microsoft Corporation, Redmond, WA, USA

Part I
Natural-Language Technology
and the Global Marketplace

Chapter 1

Making the Case for an Open, Unified System Architecture in Response to Rapid Developments in the Natural Language Industry: Translingual Automatic Language Exploration the System (TALES)

Leiming Qian, Imed Zitouni, and John F. Pitrelli

Abstract The demand for high-throughput low-latency natural language processing (NLP) systems calls for a unified, open system architecture that allows for rapid development, deployment, and interoperation of NLP engines from different vendors. The free, open-source unstructured information management architecture (UIMA) is such a platform. We demonstrate the effectiveness of UIMA with the IBM-developed Translingual Automatic Language Exploration System (TALES). TALES is a system for multimedia monitoring and search that allows English-speaking users to monitor English and foreign-language video and Web news media in near-real time and search over stored multimedia content.

Introduction

Modern communication technologies, such as television and the Internet, have made readily available massive amounts of information in many languages. More such data is being generated in real time, 24 hours a day and 7 days a week, aided by social networking sites such as Facebook and crowd-sourcing activities such as CNN iReport. This information explosion is in the form of multilingual audio, video, and Web content. It demands effective, scalable, multilingual media processing, monitoring, indexing, and search solutions.

Natural language processing (NLP) technologies have long been used to address this task, and several institutions have developed technical solutions for it.

L. Qian (✉) • J.F. Pitrelli

IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

e-mail: qianl@us.ibm.com

I. Zitouni

Microsoft Corporation, Redmond, WA, USA

e-mail: izitouni@microsoft.com

Such a system not only can handle multimodal data but also must cross the language barrier, and process data with high throughput and low latency. As an example, a user who only masters English should be able to monitor and search not only English content but also media originating in other languages, and have results presented in the user's language, English in this case. Nowadays, such systems typically rely on proprietary system infrastructure and feature a rigid, predefined set of analysis engines, quite often vendor-optimized.

Because of the rapid advancement in the NLP field, more vendors and different components optimized for different application settings have started to enter the picture. The situation clearly calls for an open, unified system architecture that allows interchangeable heterogeneous components, and can quickly integrate engines for various NLP subtasks, such as speech recognition, translation, and information extraction. It should also facilitate interoperating engines from different vendors and adapt to the ever-changing application requirements.

We present in this chapter one such architecture that moves in the direction of an open, unified system that allows interchangeable heterogeneous components. The architecture is validated with the development of the IBM Translingual Automatic Language Exploration System (TALES)¹, built on top of the open-source unstructured information management architecture (UIMA) (Ferrucci and Lally 2004) framework. We present in the following section a brief description of TALES' architecture. We then introduce the various NLP components of TALES. Section “[Introduction to UIMA](#)” introduces an important framework used by TALES, which is UIMA. In this section we show how a system similar to TALES may use heterogeneous components that can come from different vendors optimized for different application settings. UIMA has the advantage of allowing quick engine integration and adaptation to ever-changing application requirements.

TALES

TALES is a system which allows English-speaking users to monitor English and foreign-language video and Web news media in near-real time and search over stored content. TALES captures multilingual TV news broadcasts and crawls web sites in multiple languages daily. The collected data is passed through a series of NLP engines to extract metadata that can be displayed, indexed, and searched.

More specifically, for audio/video data, the following engines are used:

1. A Speech-To-Text (STT) engine to create an original-language transcription of the media, including the timestamp for every recognized word (Saon et al. 2005).

¹IBM Translingual Automatic Language Exploration System (TALES): This work was partially supported by the Defense Advanced Research Projects Agency under contract No. HR0011-06-2-0001, http://researcher.watson.ibm.com/researcher/view_project.php?id=2228.

2. A Statistical Machine Translation (SMT) engine to produce an English transcript of the media (if the media is non-English), including word token-level alignments between the source and target sentences (Tillmann and Ney 2003; Al-Onaizan and Papineni 2006).
3. Optionally, a Text-To-Speech (TTS) engine to produce an English sound track for the source media based on the English transcription generated in step 2 (Pitrelli et al. 2006).
4. An Information Extraction (IE) engine to identify mentions of entities such as person, location, organization, events, and relations between these entities. This analysis is done on both the source language and English (Florian et al. 2004).
5. A Speaker and Gender Segmentation (SGS) engine to track speakers based on acoustic signatures and to detect their gender (Huang et al. 2007a).
6. A Language and Dialect (LD) identification engine to label speech segments with language and dialect information (Navratil 2006).
7. For video data, keyframe images are extracted, which form the basis for visual presentation and the optional image search engine IMARS (IBM Multimedia Analysis and Retrieval System 2999) from IBM.
8. All the metadata generated in the above steps are stored and indexed, making them available for search using the open-source Apache Solr indexer².

When dealing with Web/text data, TALES first detects the page language using IBM Linguini technology (Prager 1999), and then processes it with engines in steps 2, 4, and 8, skipping the audio/video-specific processing steps.

TALES has been running 24×7 for many years now at IBM as well as at other commercial and government customer sites. IBM SMT technology and the TransBrowser/TransWidget toolset, as will be described in section “TransBrowser,” have been licensed by a major language technology provider. At the moment, TALES processes content from the following languages: English, Chinese (simplified and traditional), Modern Standard Arabic, Farsi, and Spanish. More languages are being added. We also find other news monitoring systems in the market. Examples include BBN Technology’s Broadcast Monitoring System (BMS) (BBN Technology Broadcast Monitoring System 2999), Autonomy’s Virage (Autonomy Virage 2999), and Volicon’s Observer (Volicon Observer 2999). One of the fundamental unique features of IBM’s TALES system is the underlying open-source UIMA infrastructure that we discuss later in this chapter.

NLP Engines in TALES

In this section we provide some technical details on all the engines used in the TALES system. It is meant to be a starting point for interested reader to investigate further.

²Apache Solr: An open-source enterprise search server based on the Lucene search library.

Speech Recognition

The Automatic Speech Recognition (ASR) engine operates as follows. First, the input audio is segmented into speech and nonspeech, and the speech segments are clustered together into clusters corresponding to single speakers. Next, the speech segments are decoded with speaker-independent models to provide a first-pass transcript. This transcript is used to perform several speaker adaptation steps: vocal tract-length normalization, and feature- and model-space maximum-likelihood linear regression. Lastly, the speech segments are re-decoded with a set of discriminatively trained speaker-adapted models to provide the final recognition output. Decoding is done using a time-synchronous Viterbi algorithm operating on weighted finite-state transducers. The engine is part of a more general ASR toolkit that is used to train and decode with Gaussian-mixture-based hidden Markov models. The toolkit was written in C++ and Python as has been described previously in more detail (Soltau et al. 2010).

Machine Translation

IBM's machine-translation decoder presents IBM's extended implementation of some widely-used phrase-based decoders (Och and Ney 2004) for SMT. It tries to handle the search problem in a principled way: it links the word reordering problem in SMT to the traveling salesman problem (TSP) (Held and Karp 1962). A dynamic programming (DP)-based optimization algorithm to solve the TSP serves as a starting point to handle the word reordering problem efficiently. The DP-based search algorithm was extended to phrase-based SMT (Tillmann 2006). The IBM decoder uses only two stacks, which results in a run-time reduction of a factor of 2 as compared to the other phrase-based decoders (Koehn 2004).

IBM's TRL decoder is based on a simple block-based model for phrase-based SMT (Kumar and Byrne 2003). A block is a pair of phrases that are translations of each other. During decoding, we view translation as a block-segmentation process, where the input sentence is segmented from left to right and the target sentence is generated one phrase at a time. For some language pairs, close to state-of-the-art performance can be obtained by generating a largely monotone block sequence. Under this view, SMT becomes quite similar to sequential natural language annotation problems such as part-of-speech tagging, phrase chunking, and shallow parsing. The restriction to some local block-based word reordering allows for very fast DP-based decoding algorithms, e.g., several hundred words per second. A variant of the TRL decoder was also used for the first successful experiments with a phrase-based SMT system that uses only binary features (Tillmann and Zhang 2006).

The next generation of machine translation decoder, also known as the Direct Translation Model (DTM) (Ittycheriah and Roukos 2007), utilizes a Maximum Entropy model to guide the search over all possible target strings given a

source sentence. The model is trained over all Arabic-English corpora available through the LDC and DARPA's GALE (DARPA Global Autonomous Language Exploitation 2999) and BOLT (DARPA Broad Operation Language Translation 2999) programs. The decoder uses a multi-stack beam search to find the best target sentence where each extension is scored by a set of costs that are commonly found in phrase-based decoders. DTM incorporates blocks that capture (a) source deletions, and (b) variables on both the source and target side, with the constraint that the edges of such blocks have to be lexicalized. Results using this decoder have consistently been very competitive in various government evaluations conducted by NIST.

Information Extraction

TALES uses IBM's Statistical Information and Relation Extraction (SIRE) (IBM SIRE toolkit, http://researcher.watson.ibm.com/researcher/view_project.php?id=2223) toolkit to extract mentions, entities, and relations from text that may result from detagging Web news or from running speech recognition on audio content.

SIRE's mention detection engine uses statistical techniques to identify and classify textual references that correspond to real-life entities, such as person, organization, location, date, etc. It identifies contiguous nominal, named, and pronominal text regions (mentions) that refer to such entities and assigns them the corresponding semantic type; it groups them into chains of mentions that refer to the same entity (coreference chains), and also identifies relationships among them. As part of the processing, it performs a large set of NLP tasks, such as tokenization and word segmentation, sentence segmentation, parsing and semantic-role labeling, and case restoration. The statistical foundation of the mention detection engine consists of several exponential classifiers, including maximum entropy Markov models and conditional random fields. It achieves state-of-the-art results on many datasets, and has been used to create models on various datasets, such as MUC, ACE, and KDD (MUC-7 1998; The ACE Evaluation Plan and NIST 2007) and it can operate in multiple languages, including English, Chinese, Arabic, Spanish, Urdu, Italian, and French. SIRE's mention-detection capability also has provision for robustness to malformed input, such as unexpected markup (Florian et al. 2010).

SIRE's coreference engine is a statistical entity-mention-based system that uses a Bell tree (Luo et al. 2004; Luo and Zitouni 2005) to represent the search space and a maximum entropy model to score coreference hypotheses. The system is typically trained on a few hundred documents for which coreference chains have been manually annotated. At test time, the coreference system performs a beam search on the Bell tree, and the coreference result with the best score is returned. The trainable coreference system enjoys many benefits of the data-driven approach. First, it can benefit automatically from the availability of more training data; second, it is easily adaptable to multiple languages; third, linguistic constraints can be

expressed as features in the aforementioned maximum entropy model and their weights are determined automatically at training time.

The SIRE relation extraction engine consists of a collection of maximum entropy models applied in sequence (Kambhatla 2004; Bikel and Castelli 2012). The first model detects the existence of a relation between a pair of entity mentions that occur in the same sentence. The other models compute individual attributes of detected relations.

Speech Synthesis

IBM's TTS synthesis engine (Pitrelli et al. 2006) is a corpus-driven concatenative speech synthesizer. A corpus of recordings of texts chosen for richness in phonetic sequences and common words and phrases is cut into sub-phonetic segments. Then, at synthesis time, a front-end uses dictionaries and pronunciation rules to determine a phonetic sequence for input text; statistical models set pitch and duration targets. Targets may also be influenced by text markup; for example, if an airline-reservation-system user requested a flight from O'Hare and the system found a flight from Midway, the word "Midway" might be marked-up by the system for contrastive emphasis in the synthesis output. These targets as well as spectral-continuity criteria are used to select from the corpus a sequence of sub-phonetic segments to be joined to form the output speech. Signal smoothing is optionally applied to the splice points.

Language Identification

Automatic Language Identification is the process of automatically determining the language spoken in a speech recording. LID systems use several sources of phonotactic, acoustic, and high-level information to model the language at different levels. The lattice-based parallel phone recognition language modeling (PPRLM) system (Zissman 1996) is one of the most successful approaches to using phonotactic information in LID. This system generates phonetic lattices using multiple phone decoders with a null grammar. The phonetic outputs are compared against a set of models and the language of the model with the highest likelihood is selected. For the TALES system, an extension to this work using binary decision trees to represent phone N-gram probabilities was chosen. In this setup, a binary tree is established for each language that will be learned by the system. With this technique, the binary tree estimates for each language which phonetic sequences are more or less common. When a new speech segment is analyzed, a phone recognition system transcribes the speech and the resulting phone sequence is compared with each of the models. The corresponding language of the model with the highest likelihood is chosen. While language identification is an interesting technology, research (Omar and Pelecanos 2010) demonstrated that an automatic system can

determine not only the spoken language in a recording but also the native language of a speaker when the speaker is communicating in English.

Gender Identification

Speech can also be analyzed to infer gender. The technique used in TALES is to extract short-term spectral-based features (Mel-Frequency Cepstral Features) and compare them against male and female models. Other features, such as pitch, can also be helpful for this task. In the interest of efficiency, we used the same features as the speaker diarization system so that two classification problems could be quickly addressed using one set of features and a slightly modified model. The type of modeling used in this setup is similar to the modeling used by Campbell (Campbell 2008). In this scenario, a general model is trained on large quantities of speech from many channels and both genders. This general model is discriminatively trained to differentiate between male and female speakers. Training the model in this way provides large improvements over using two generic, maximum-likelihood-trained models.

Speaker Diarization

Automatic Speaker Diarization (Huang et al. 2007b) is the task of analyzing a recording and categorizing the speech based on speaker. The TALES system achieves this by chunking the audio into short homogeneous segments. Each segment is compared with all other segments in a recording. After each comparison round, the most similar segments are merged. The merging process continues until a clustering threshold is reached. The number of clusters represents the number of speakers. For each cluster of segments, a model is trained and the new models are used to label the speech according to speaker. Then, each speaker's labeled speech is used to retrain the model one more time. These new models are applied to determine the final speaker segments.

TALES Functionalities

Live Media Monitoring

Near-Real-Time Video Monitoring

One of the most prominent and useful features of TALES is near-real-time live media monitoring. TALES can capture any analog video signals from composite, S-Video, component or coaxial inputs for processing. Once processing starts, users

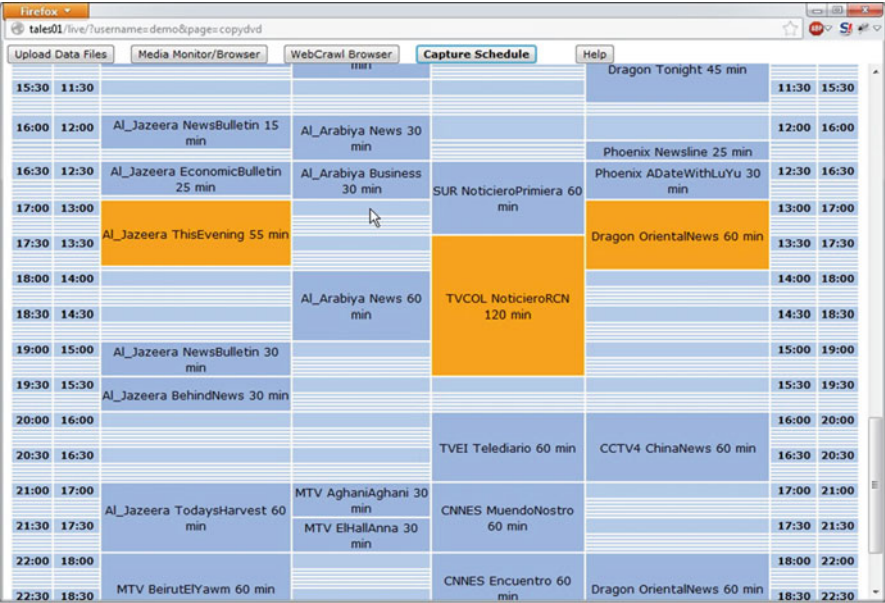


Fig. 1.1 Video capture schedule

can monitor live video by playing back the captured media and viewing machine-generated English caption with a latency of just 4.5 min.

At the moment, TALES captures Chinese-, Arabic-, Spanish-, and English-language news programs from DishNetwork based on a user-definable schedule, as shown in Fig. 1.1, with programs currently being recorded highlighted in orange. The scheduling feature is compatible with any video receiver that has an infrared remote control, so the content provider is not limited to DishNetwork. (An interesting side note is that some DishNetwork programs actually embed original source-language captions in the video signal. In that case STT is skipped and the embedded caption is extracted and used instead.)

Figure 1.2 shows the user interface for live video monitoring. Each row in the table corresponds to a news program. Programs are organized by provider, title, language, show start time and length, etc. Each show is segmented into a consecutive sequence of 2-min clips; clicking on link “6” of any particular show, for example, would start playing the program at 6 min into the show, and it would continue playing all the way until the end of the program.

The caption text displays additional color-coded metadata such as speaker, gender, language/dialect, and extracted named entities. In addition to *viewing* the caption, a user can also optionally *listen* to a TTS-generated sound track of the English caption.

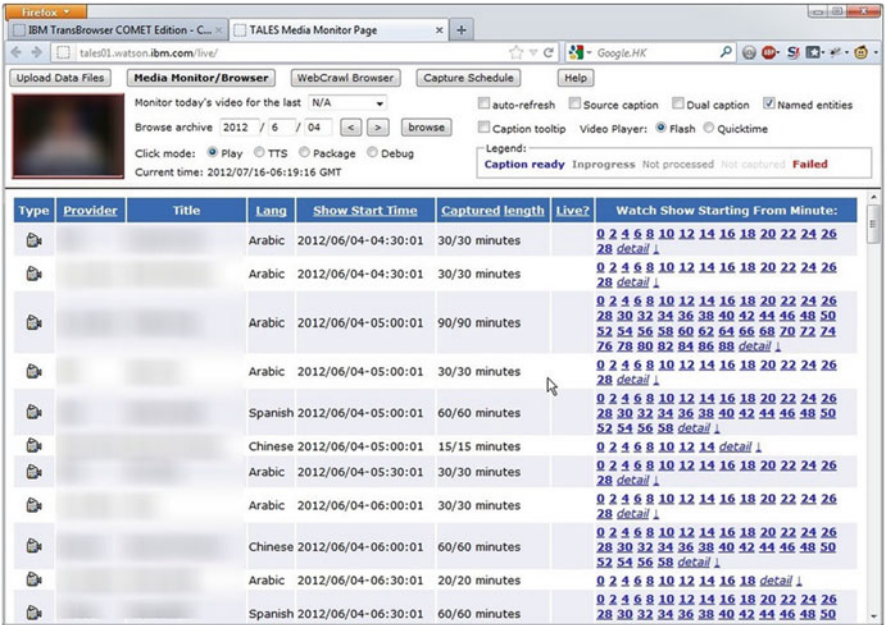


Fig. 1.2 Live media monitor

The Live Media Monitor page also allows the user to navigate the media repository by date, and export a particular clip or the entire show with associated caption, as self-contained zip packages that can be played back offline.

The Web Crawl Monitor page, as shown in Fig. 1.3, allows the user to see how many Web pages were crawled and processed.

Figure 1.4 shows the TALES video player in action. To the right of the main video area is the segment playlist panel, where the user can quickly scroll and jump to different segments. The caption displays a label “S3(ar_msa.f)” that contains detected speaker (labeled S3), language/dialect (Arabic MSA), and speaker gender (f stands for female). It also highlights several extracted named entities such as “Syria” and “Viewers.”

Furthermore, TALES supports a dual-caption mode, so a bilingual speaker can view both source language and English caption streams at the same time. In Fig. 1.5 both the English caption and the source Arabic captions are displayed. Note that both captions display the speaker, gender, language, and extracted named entity metadata.

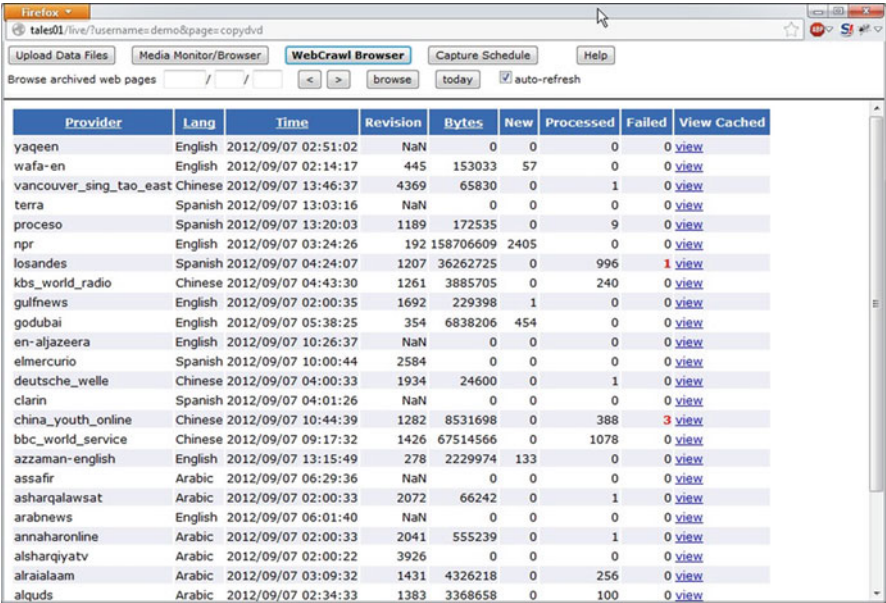


Fig. 1.3 Web crawl monitor

Speaker/Language/Dialect Display

The “view detail” link for each show on the Live Media Monitor page offers the user the ability to look at the speaker/language/dialect information for the entire captured program.

Figure 1.6 displays a waveform-like plot at the top; its horizontal axis corresponds to time, and the vertical axis corresponds to the number of words spoken during the current time window. Color highlighting is used to mark regions of speech from known speakers.

Under the plot, a horizontal colored bar is used to display regions of different detected languages and dialects.

Mousing over the speaker plot or the language bar displays the text spoken during the particular time window. Clicking on them plays the video starting from that point in time.

For example, in Fig. 1.6 the regions marked by the orange color are from a known Arabic news anchor, who speaks Modern Standard Arabic (MSA). Within the clip there is also a short interval, marked by the green color, where an off-screen speaker spoke with a Levantine Arabic dialect. This interface can help the user quickly locate within a show the speakers/dialects of interest.

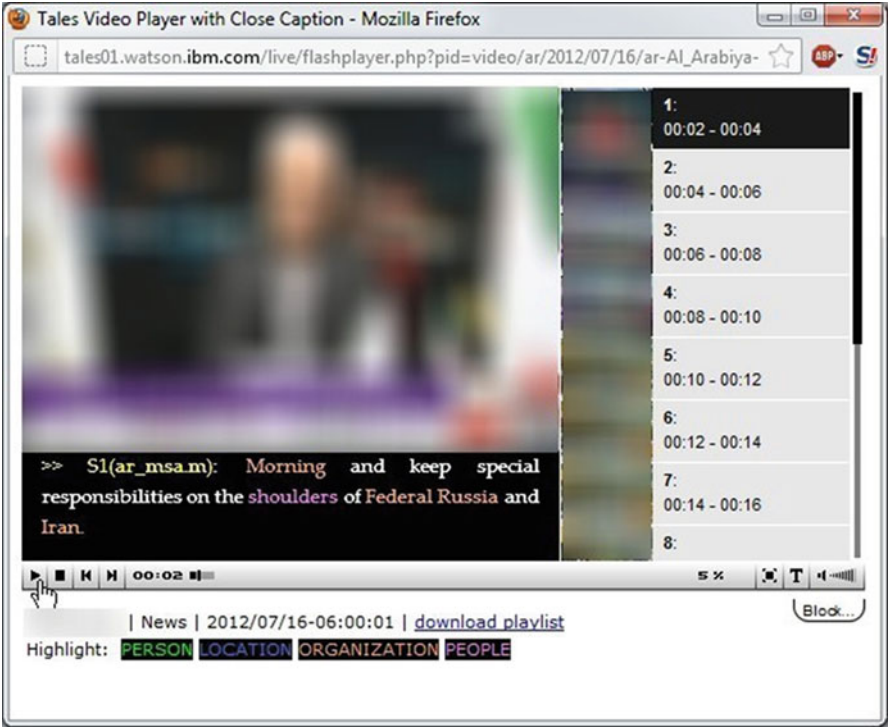


Fig. 1.4 Video player with caption and metadata display

Processing of User Data

TALES was designed from the very beginning so that user data can be easily processed. Users can upload their own media files in various formats (avi, wmv, flv, etc.) for processing, or request TALES to process files on a USB thumb drive, external hard drive, and CD/DVD media, as shown in Fig. 1.7. Users can even request that the TALES perform a one-time crawl on their Web sites of choice.

Searching Indexed Media

Keyword Search

In any system dealing with a large amount of data, a powerful, flexible search index system is essential. Multilingual media processed by the TALES system is immediately available for search by the end user using *either English keywords or keywords in the source language of the media*. TALES supports a rich set of search

Fig. 1.5 Dual-language caption



syntax including Boolean operators, sorting/filtering by language, date, and media modality. It can search for documents that contain a specific named-entity type. For example, it can search for documents that contain “Washington” the person instead of “Washington” the geographical location.

Figure 1.8 shows a search result summary page. It displays a thumbnail “slideshow” view of video results, a key image for the Web results (if available), and a snippet from the document surrounding the search keywords, with the keywords highlighted. It also includes an embedded video player with caption display. User can toggle the display of extracted named entity in the result summary.

Detailed View for Search Result

Once a search has been performed and results returned, a user can further look at the details of any returned document.

For video segments, TALES provides a “story board” view, where important key frames from the video are displayed together with the associated transcript, as shown in Fig. 1.9. Search keywords and named entities are highlighted accordingly. The user can navigate to the previous and next clip (TALES uses 2-min clips to segment long video programs). The story board view allows the user to scan quickly through



Fig. 1.6 Speaker/language/dialect display

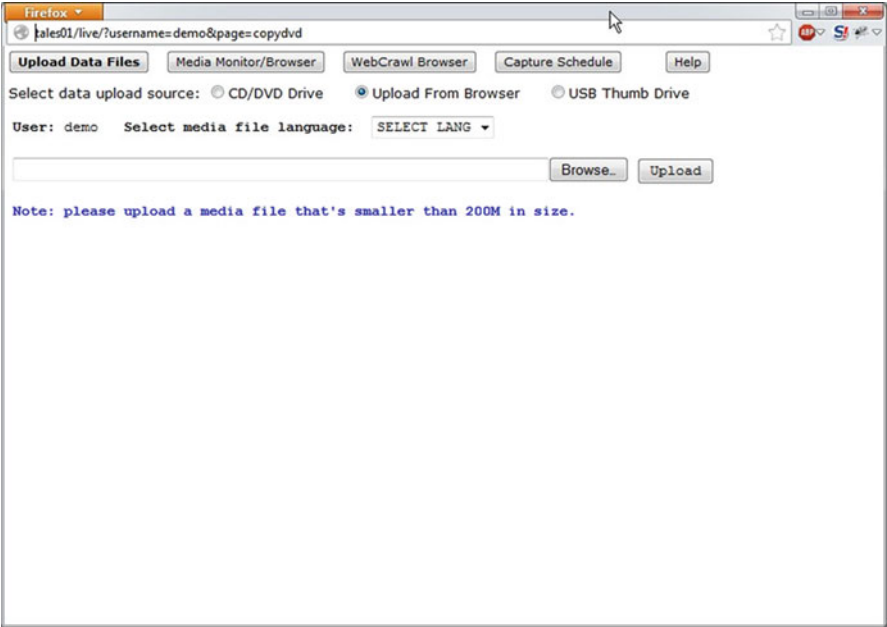


Fig. 1.7 Process uploaded file, CD/DVD and USB drives

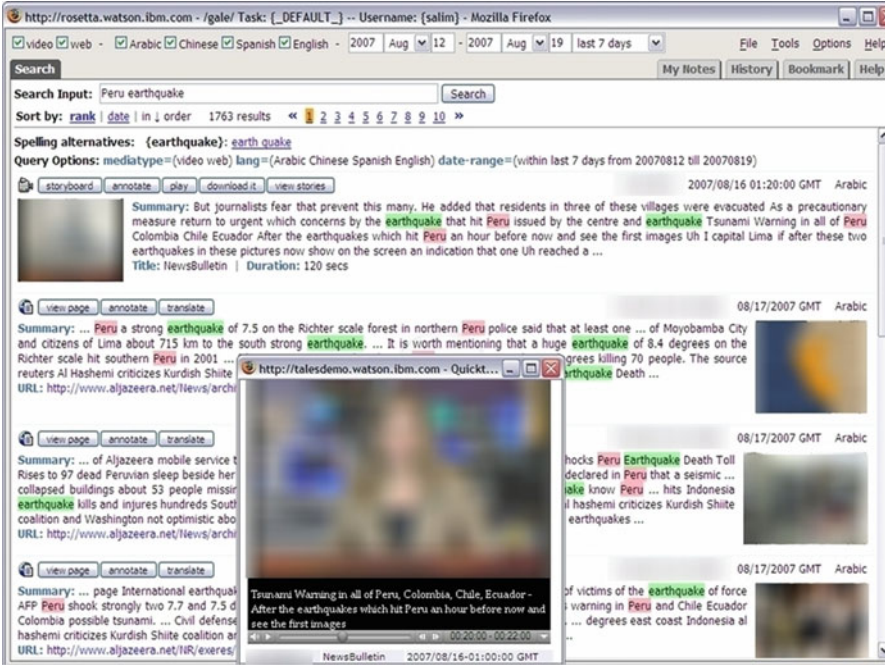


Fig. 1.8 Search results display

the document looking for interesting information, without having to listen through the entire media clip.

For Web documents, TALES offers a “cached page” view that allows the user to view both a plain-text and an HTML version of the original page, and optionally retranslate it on demand with the page layout perfectly preserved using the built-in TransBrowser tool. Similarly, named entities and search keywords are highlighted on the page, as shown in Fig. 1.10.

User Alerts

Depending on the daily volume of data processed by TALES, it could be overwhelming for any user to go through them manually. To address this issue, the TALES search user interface allows users to define customized search-based alerts, as shown in Fig. 1.11. Whenever TALES processes a document that matches the user’s predefined query, the user will be notified via e-mail. The alerts are defined using the search query language. This further enhances the “monitoring” capability of TALES.

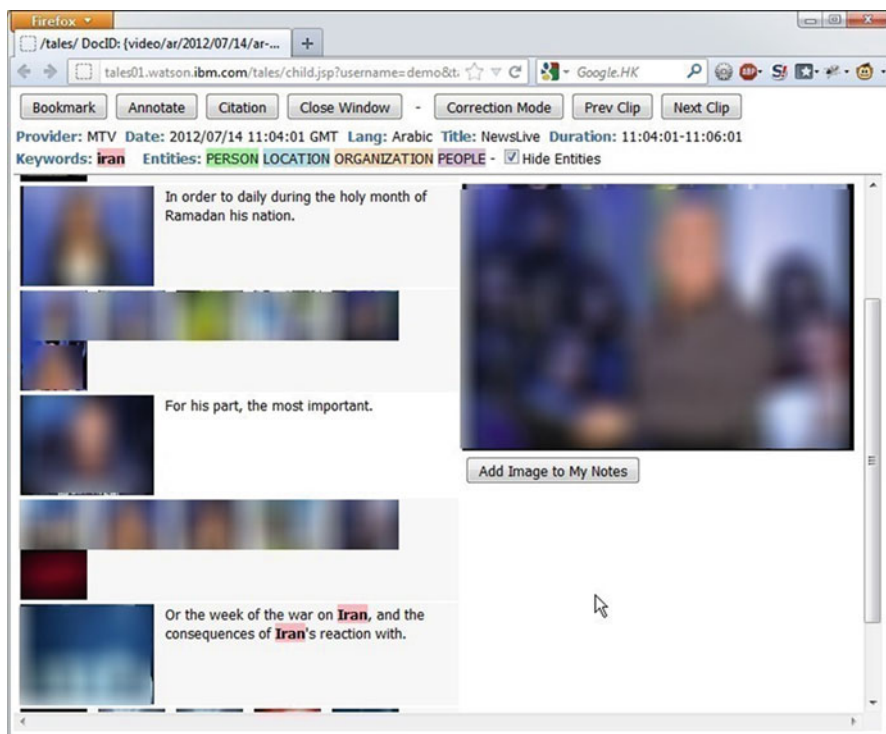


Fig. 1.9 Story board view for video result

Web Page Translation and Information Extraction

For Web pages, TALES provides a set of utilities that directly exposes the function of underlying NLP machine translation and information extraction.

TransBrowser

TransBrowser (patent pending) is an on-the-fly Web page translation tool that allows end users to seamlessly translate a Web page and all its linked pages with perfect layout preservation, effectively enabling them to navigate foreign news Web sites as if they were written in English. It has several advantages over other similar Web page translation tools in the following aspects:

1. TransBrowser translates the *page content* instead of the *page source code*; therefore it can handle dynamically generated pages and pages that rely heavily on Javascript manipulation.

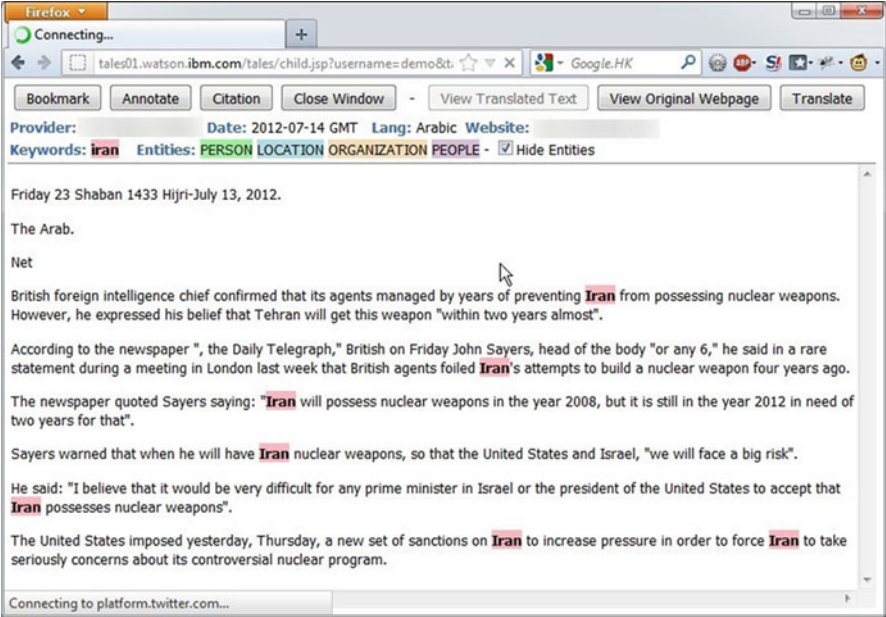


Fig. 1.10 Cached Web page result

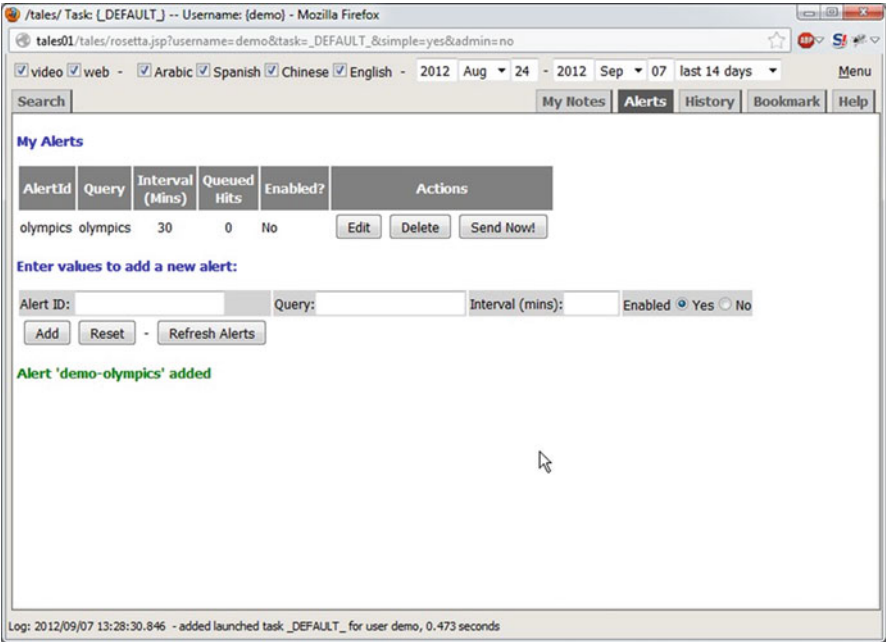


Fig. 1.11 User alert management user interface

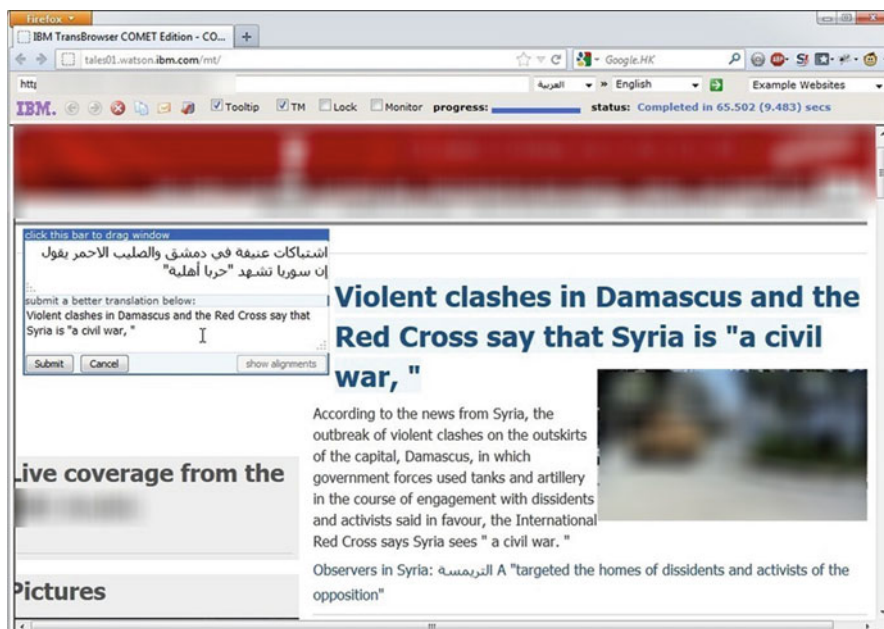


Fig. 1.12 TransBrowser with user correction

2. TransBrowser translates the text nodes in the Web page based on the Document Object Model (DOM) tree; therefore it can achieve perfect page layout preservation.
3. TransBrowser utilizes the server-push technology to send translations back to the browser client as soon as some partial translation output is available, instead of waiting for the entire page to be translated. As a result it can translate the page *progressively*, enabling the user to read the translated page while translation is still in progress, even giving the user the option to cancel the page translation.

As shown in Fig. 1.12, TransBrowser allows the end user to submit corrections to the machine translation engine. These human corrections will be used next time the same sentence is translated; they will also be used for improving the statistical machine translation model.

TransBrowser has three variants:

TransWidget: A single-line Javascript that can be embedded into a Web page by its author.

```
<script type='text/javascript'
  src='tw.jsp&from=en-us'></script>
```

This generates a pull-down menu widget on the page, as shown in Fig. 1.13, enabling visitors to the page to translate it into their native language.

Fig. 1.13 TransWidget

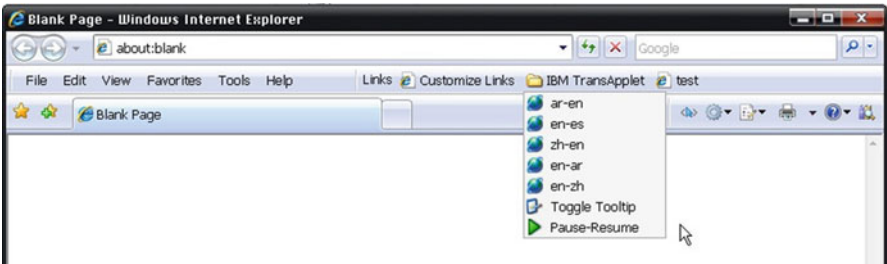


Fig. 1.14 TransApplet

TransApplet: A Javascript *bookmarklet* (Bookmarklet Wikipedia definition 2999), a small piece of javascript code that can be stored as a bookmark/favorite in the user’s browser, and used to translate any arbitrary Web page, as shown in Fig. 1.14. By design, Web-based translation proxies such as Google Translate and TransBrowser cannot translate any page that requires user authentication. Because TransApplet runs as a user-evoked bookmarklet inside the browser client, it is not bound by this restriction and can translate any page.

FoxTran: A Mozilla Firefox plug-in that provides several additional features beyond basic translation: automatic page source-language translation, do-not-translate blacklist, mixed-language content translation, etc.

InfoBrowser

InfoBrowser is similar to TransBrowser, except it substitutes page translation with page information extraction, highlighting all the named entities such as person, location, and organization found in the Web page. Furthermore, clicking on a person, for example, would link the user to the Wikipedia biography page of that person.

Introduction to UIMA

UIMA is a software specification that allows the analysis of large volumes of unstructured data in order to discover relevant information. It has been approved as an open standard by OASIS, a standards organization. It has an open-source framework implementation from Apache Foundation.

All NLP components can be considered as “annotators” that label the source data with metadata. For example, speech recognition generates transcription metadata for audio segments; machine translation generates annotation in a different language for the text in source language; word segmentation generates offset annotation; even TTS can be considered as audio annotation on text data. UIMA, as diagrammed in Fig. 1.15, takes this concept and provides the following functionality:

- It decomposes NLP applications into an aggregate of component annotation engines, defined by a set of configuration data in XML descriptor files.
- The framework manages the annotators and data flow between them.
- UIMA enables single-point invocation of an aggregate of engines on heterogeneous native computing platforms such as operating system and programming language.

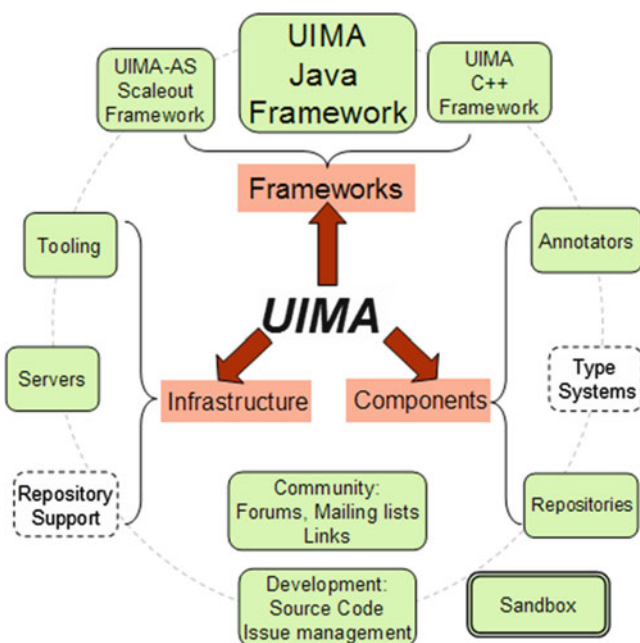


Fig. 1.15 UIMA framework

- Annotators can be wrapped as network services to allow asynchronous scaleout using a cluster of networked nodes.
- In addition, UIMA provides all the tooling, utilities, and debugging aid necessary to develop the NLP application.

UIMA applications require a “type system,” a definition of data types to be shared by the components in order to enable interoperating them. An example type system is the GALE Type System (GTS) (Pitrelli et al. 2008a) which defines types such as AudioToken to represent a word of speech recognition output by associating a character string with a span of time, and WordToken which represents a word in a text string as a span of characters. Aggregating existing engines into a UIMA aggregate typically entails “wrapping” them to convert their inputs from the type system’s format to the engine’s native format, run the engine, and then convert the outputs to the type system’s format.

When components are colocated and share a computing platform, a UIMA aggregate may be operated as a single process; TALES is normally operated in this mode. In the case of heterogeneous, distributed components, however, UIMA’s Asynchronous Scaleout option provides for deployment of individual components or aggregates as services, which are addressed remotely using, e.g., TCP by a client application representing the entire aggregate. The client need not even be aware of which operating system or programming language is employed on the server side. An example of such a heterogeneous, distributed NLP application is the GALE Interoperability Demo (IOD) system (Pitrelli et al. 2008b), which aggregates deployed versions of many of the TALES engines, including speech recognition, information extraction, and machine translation, but additionally interoperates parallel engines with the same functions from sites other than IBM, enabling system-combination strategies, as well as other engine functions from other sites, such as topic clustering and multi-document summarization, to form an application which allows English text browsing of summaries of topic clusters of multilingual media.

UIMA aggregates can be configured with parallel and serial elements, in order to optimize for latency, using XML constructs such as

```
<flow>
<engine>Language_ID_A</engine>
<parallelEngines>STT_A,STT_B,STT.C</parallelEngines>
<parallelEngines>IE_A,IE_D,IE.E</parallelEngines>
<parallelEngines>SMT_B,SMT_E,SMT.F</parallelEngines>
<engine>SMT_System_Combination.G</engine>
<engine>TTS_B</engine>
</flow>
```

where suffixes A through G represent different engine providers.

In addition to remote operation and heterogeneous platforms, UIMA’s Asynchronous Scaleout capability also enables load balancing within an application and sharing of engine services across applications. For example, in an aggregate

like this one, STT is often the most time-consuming component, often suffering three to five times as high latency as any of the rest. Rather than leaving the other components idle 65–80 % of the time, multiple instances of the high-latency engines may be deployed to serve the same request queue. Deployment of several copies of each STT engine in IOD reduced latency and increased utilization of engines' computational resources dramatically. In addition, scaleout enables multiple applications to share server resources, by simply referencing the same engine service URLs. Thus, a speech-to-speech-translation application could be configured like

```
<flow>
  <engine> STT_B</engine>
  <engine> SMT_B</engine>
  <engine> TTS_B</engine>
</flow>
```

and more instances of STT_B might be deployed for responsive support of the two simultaneously operating applications. In this way, UIMA provides for multiple TALES-like applications, operating locally or remotely, using heterogeneous computing environments, with scaleout to manage load balance within and across such NLP applications.

Summary

TALES is a multilingual, multimodal analytic system that lets English speakers collect, index, and access information contained in English and foreign-language news broadcasts and Web sites. TALES technology is built on top of the UIMA platform and uses multiple IBM natural language technology components. TALES enables users to search English and foreign-language news, play back streaming video with English captioning, monitor live video with low latency time, browse and translate foreign Web sites, etc. TALES has been deployed in multiple customer sites.

References

- Al-Onaizan Y, Papineni K (2006) Distortion models for statistical machine translation. In: Proceedings of 21st international conference on computational linguistics and 44th annual meeting of the ACL, Sydney, pp 529–536
- Autonomy Virage, <http://www.virage.com/>
- BBN Technology Broadcast Monitoring System, http://bbn.com/products_and_services/bbn_broadcast_monitoring_system/
- Bikel D, Castelli V (2012) Relations and events. In: Bikel D, Zitouni I (eds) Multilingual natural language processing and applications, chapter 9. IBM Press, Upper Saddle River, NJ

- Bookmarklet Wikipedia definition, <http://en.wikipedia.org/wiki/Bookmarklet>
- Campbell W (2008) A covariance kernel for SVM language recognition. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Las Vegas, Nevada, USA
- DARPA Broad Operation Language Translation (BOLT) program, <http://www.darpa.mil/Our-Work/I2O/Programs/Broad-Operational-Language-Translation.%28BOLT%29.aspx>
- DARPA Global Autonomous Language Exploitation (GALE) program, <http://projects ldc.upenn.edu/gale/>
- Ferrucci D, Lally A (2004) UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat Lang Eng* 10(3–4):327–348
- Florian R, Hassan H, Ittycheriah A, Jing H, Kambhatla N, Luo X, Nicolov N, Roukos S (2004) A statistical model for multilingual entity detection and tracking. In: Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics: HLT-NAACL 2004, Boston, MA, pp 1–8
- Florian R, Pitrelli JF, Roukos S, Zitouni I (2010) Improving mention detection robustness to noisy input. In: Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), Cambridge, MA, USA, October 2010
- Held M, Karp RM (1962) A dynamic programming approach to sequencing problems. *SIAM* 10(1):196–210
- Huang J, Marcheret E, Visweswariah K, Potamianos G (2007a) The IBM RT07 evaluation systems for speaker diarization on lecture meetings. In: Stiefelhagen R, Garofolo J (eds) *Multimodal technologies for perception of humans*. Springer, Baltimore, MD
- Huang J, Marcheret E, Visweswariah K (2007) Improving speaker diarization for CHIL lecture meetings. In: Proceedings of interspeech, Antwerp, Belgium, pp 1865–1868
- IBM Multimedia Analysis and Retrieval System (IMARS) http://researcher.watson.ibm.com/researcher/view_project.php?id=877
- IBM Statistical Information and Relation Extraction (SIRE) toolkit, http://researcher.watson.ibm.com/researcher/view_project.php?id=2223
- Ittycheriah A, Roukos S (2007) Direct translation model 2. In Proceedings of HLT 2007, Rochester, New York, pp 57–64
- Kambhatla N (2004) Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on interactive poster and demonstration sessions, Barcelona, Spain, p 22
- Koehn P (2004) Pharaoh: a beam search decoder for phrase-based SMT models. In: Proceedings of AMTA'04, Washington, DC
- Kumar S, Byrne W (2003) A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In: Main proceedings of HLT-NAACL 2003, Edmonton, Alberta, Canada, pp 142–149
- Luo X, Zitouni I (2005) Multi-lingual coreference resolution with syntactic features. In: Proceedings of HLT-EMNLP, Vancouver, British Columbia, Canada, pp 660–667
- Luo X et al (2004) A mention-synchronous coreference resolution algorithm based on the bell tree. In: Proceedings of the 42nd ACL, 2004, Barcelona, Spain, pp 135–142
- MUC-7 (1998) Proceedings of the Seventh Message Understanding Conference. Morgan Kaufmann, San Francisco, CA
- Navratil J (2006) Automatic language identification. In: Schultz T, Kirchhoff K (eds) *Multilingual speech processing*. Academic, Orlando, FL, pp 233–272. ISBN 13.978-0-12-088501-5
- Och F-J, Ney H (2004) The alignment template approach to statistical machine translation. *Comput Ling* 30(4):417–450
- Omar M, Pelecanos J (2010) A novel approach to detecting non-native speakers and their native language. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Dallas, Texas, USA
- Pitrelli JF, Bakis R, Eide EM, Fernandez R, Hamza W, Picheny MA (2006) The IBM expressive text-to-speech synthesis system for American English. *IEEE Trans Audio Speech Lang Process* 14(16):1099–1108

- Pitrelli JF, Lewis BL, Epstein EA, Quinn JL, Ramaswamy G (2008) A data format enabling interoperation of speech recognition, translation and information extraction engines: the GALE type system. In: Proceedings of Interspeech 2008, Brisbane, Australia, 23–26 September 2008, pp 1654–1657
- Pitrelli JF, Lewis BL, Epstein EA, Franz M, Kiecza D, Quinn JL, Ramaswamy G, Srivastava A, Virga P (2008) Aggregating distributed STT, MT, and information extraction engines: the GALE interoperability-demo system. In: Proceedings of Interspeech 2008, Brisbane, Australia, 23–26 September 2008, pp 2743–2746
- Prager JM (1999) Linguini: language identification for multilingual documents. In: Proceedings of HICSS-32. The 32nd annual Hawaii international conference on system sciences, vol 2. Maui, HI, USA
- Saon G, Povey D, Zweig G (2005) Anatomy of an extremely fast LVCSR decoder. In: Proceeding of Eurospeech 2005, Lisbon, Portugal
- Soltau H, Saon G, Kingsbury B (2010) The IBM Attila speech recognition toolkit. In: Proceedings workshop on speech and language technologies, Berkeley, CA, pp 97–102
- The ACE Evaluation Plan, NIST, 2007, www.nist.gov/speech/tests/ace/index.htm
- Tillmann C (2006) Efficient dynamic programming search algorithms for phrase-based SMT. In: Proceedings of the workshop CHPSLP at HLT 2006, June 2006, New York, pp 9–16
- Tillmann C, Ney H (2003) Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput Ling* 29(1):97–133
- Tillmann C, Zhang T (2006) A discriminative global training algorithm for statistical MT. In: Proceedings of ACL-COLING 2005, Sydney, Australia, pp 721–728
- Volicon Observer, <http://www.volicon.com/index.php/products/observer/>
- Zissman M (1996) Comparison of four approaches to automatic language identification of telephone speech. *IEEE Trans Speech Audio Process* 4(1):31–44

Chapter 2

The Burgeoning of Medical Social-Media Postings and the Need for Improved Natural Language Mapping Tools

Kerstin Denecke and Nazli Soltani

Abstract Medical social-media data provides a wealth of data generated by both healthcare professionals and patients alike. In fact, there are many medical social-media sites such as forums, where patients freely dialog with a healthcare professional or with other patients, often posing questions and responding to advice, or Weblogs, where groups of people describe their experiences with medical conditions and the various treatment plans to treat those conditions. All in all, one can no longer ignore the fact that social media has dramatically changed the structure of healthcare delivery in many ways. Simply from a medical data standpoint alone, social-media platforms have altered the way medical information is disseminated. That is, important medical information is no longer found exclusively in patients' clinical narratives, commonly shared by physicians and other healthcare workers at regular professional meetings and conferences. Instead, user-generated content on the Web has become a new source of useful information to be added to the conventional methods of collecting clinical data. The challenge we face, however, is to design information extraction tools that can make the rich resources of medical data found in social-media postings exploitable. In this chapter we analyze the linguistic features of medical social-media postings juxtaposed to the linguistic features of both clinical narratives (e.g., discharge summaries, chart reviews, and operative reports) and biomedical literature, for which there already exists tools for performing information extraction. We show the shortcomings of these mapping tools when applied to medical social-media postings, and propose ways to improve

K. Denecke (✉)

Innovation Center Computer Assisted Surgery, University of Leipzig, Leipzig, Germany

e-mail: kdenecke@web.de

N. Soltani

University Medical Center, Göttingen, Germany

e-mail: nazli.soltani@med.uni-goettingen.de

such tools so that the wealth of medical data located in medical social-media can be made available to healthcare providers, pharmaceutical companies, and government-supported epidemiological agencies.

Social Media and Its Use in Healthcare

As advances in the Internet and mobile technologies have improved the way how people access, use, and share information in the last few years, new ways of communicating about health have evolved, enabling a 24/7 and location-independent medical-information exchange. These new media comprise instant messaging (e.g., Twitter), blogs, online forums, social networking (e.g., Facebook), or video sharing (e.g., YouTube).¹ Mayo Clinic researchers have opined that social media has begun a process of “revolutionizing healthcare” by improving healthcare and quality of life (Aase et al. 2012). In fact, patients increasingly rely on the Internet when looking for medical information and advice. The Internet also facilitates patients’ ability to share their personal experiences and opinions with others who have the same health concerns.

In analyzing the popularity of social media one cannot help realize that one distinct advantage is that the communication barriers are considerably lower than face-to-face communication, allowing patients to write in social-media platforms a bit more freely about their illnesses and their experiences with drugs and medical treatments than they would normally do in other settings. One of the most active groups of online health-information seekers are those who suffer from chronic conditions or from rare diseases. By surfing for medical information in cyberspace this group of users can learn what others have to say about quality of care, or about important issues regarding treatment and diagnosis. In addition to gaining knowledge, this group of online medical-information seekers is able to communicate with other persons who are suffering from the same disease, thus serving as a hedge against feelings of isolation and loneliness. All in all, social-media platforms offer a range of possibilities to facilitate the sharing of useful medical information and personal experiences related to healthcare, particularly when the information seeker is unable to obtain adequate support and feedback from fellow sufferers in their home community.

Another important observation worthy of mention here is that one cannot ignore the fact that social media has dramatically changed the structure of healthcare delivery in the modern world. From a medical data standpoint, important medical information is no longer exclusively found in patient clinical narratives, usually shared by physicians and other healthcare workers at regular professional meetings and conferences. Instead, user-generated content on the Web has become a new source of useful information to be added to the conventional methods of collecting clinical data.

¹While younger populations were fast in adopting these new technologies, the number of older adults using social media is also growing fast.

Given the wealth of user-generated content, automatic methods are essential for extracting relevant information, for organizing and digesting the data for various user groups as well as for preparing it for statistical analysis. Medical social media has also created a burgeoning new class of empowered patients “armed with sophisticated technological tools” (Altarum Institute (2012)), yet we can’t afford to lose this fountain of medical data generated daily by healthcare consumers and providers who avail themselves of social-media platforms for sharing their clinical experiences.

Against this ambitious task of extracting, organizing, and distilling medical social-media data automatically are a number of concerns that cannot be overlooked. For example, the existing approaches to information extraction primarily focus on clinical narratives or biomedical literature. It is still unclear whether these approaches and tools are suited for processing and analyzing medical social-media data. Furthermore, one must consider whether existing systems that understand the technical language of biomedical literature or the language of healthcare professionals appearing in clinical narratives for that matter can be adapted to the way ordinary people speak about their medical conditions. In this chapter, we explore the unique linguistic characteristics of medical social media and analyze two existing information-extraction tools for clinical narratives and biomedical literature to see if they can similarly be used to identify important medical concepts found in consumer blog postings. We also suggest important ways of modifying such tools so that they can perform better in analyzing medical social-media content.

Examples of Medical Social-Media Communication Platforms

Medical social media comprises Weblogs, forums, or social network platforms that deal with health-related issues. Forums are basically Internet message boards, where patients or friends and relatives of patients discuss their own experiences and personal thoughts or in the alternative ask questions and seek advice. A blog differs from a discussion forum in that all of the Web site entries are displayed in reverse chronological order; a blog often has only one author whereas many persons contribute to a forum. A blog is defined specifically as a medical blog, when its main topic is related to health or medicine (Boulos et al. 2006). The exact number of forums, blogs, and blog postings dealing with health issues is unknown. For blogs in general, Weblog hosting services have made some numbers available. WordPress,² one of the popular Weblog hosting services, reports on its Web site that WordPress.com users produce about 29.2 million new posts and 40.5 million new comments each month.

To bring together health bloggers, entire communities have been set up. *Blognation* and *HealthBlogger Network* are examples of such communities. *Blognation*³ is

²<http://en.wordpress.com/stats/>.

³<http://www.medical-blogs.org>.

a network of blog directories that lists blogs for different categories including health, but also dealing with other topics, such as books, art, music, and lifestyle. The *HealthBlogger Network*⁴ engages over 3,500 bloggers which can be broken down into two categories: healthcare professionals and patients. In general, about 50–60 % of the healthcare bloggers are healthcare practitioners and medical researchers, often from the leading American medical schools such as Harvard or Yale, while the remaining 40–50 % of bloggers comprise patients suffering from chronic or acute illnesses (Miller and Pole 2010).

Besides blogger networks, patients or healthcare professionals are forming communities where they share their knowledge, discuss, or learn from each other. A well-known example of a medical social-media communication platform is *PatientsLikeMe*.⁵ This is a social network for patients that allows them to share health-related experiences and to compare various treatment plans. Such online conversations which can also be in the form of a medical diary often contain vast amounts of experiential knowledge. Other platforms, in contrast, try to make this kind of patients' first-hand experiences useable in some fashion. For example, patients' experiences and opinions extracted from social-media postings can be exploited for planning marketing strategies in the pharmaceutical industry or in the healthcare insurance industry. As a result, healthcare products may be improved based on patients' reported personal experiences found in medical social media.

Medical social media also represents a rich resource for learning about patient-compliance behavior, as well as their feelings, attitudes, and experiences with medical and surgical treatment. *Treato.com*, for example, is a social health site that analyzes online patient discussions, collects automatically information about what patients have to say about their medications and conditions found on the blogs and other social-media fora, and subsequently extracts and summarizes the relevant information from these postings. This is done in order to provide valuable insights into patients' opinions, attitudes, and experiences that would help pharmaceutical companies better improve their products.

Platforms such as *Webicina*⁶ provide access to curated, medical social media which is defined as media that is filtered, selected, and reviewed. This helps to make more efficient use of the massive amount of social-media data that is out there. Many of those who rely on Webicina are physicians, pharmaceutical companies, and other healthcare professionals. Webicina is certainly not in a category all by itself as there are other enterprises that likewise try to make use of medical social-media data by effectively curating the information found in cyberspace (Hillan 2003; Himmel et al. 2008). Beyond getting information on health-related issues, sudden changes in the public health status can be identified in medical social-media data. This allows for a prompt reaction, early on, from health organizations to such critical events like a swine flu outbreak (Denecke et al. 2012).

⁴<http://www.wellsphere.com/health-blogger>.

⁵<http://www.patientslikeme.com/>.

⁶<http://www.webicina.com>.

Linguistic Characteristics of Medical Social Media

Medical social-media data is written for different purposes than clinical texts and biomedical literature, even though authors can be healthcare professionals as well as patients. Thus, the literary style of medical social media is markedly different than that of clinical texts and biomedical literature. Whereas the linguistic characteristics of clinical and biomedical texts have been analyzed in painstaking detail by other researchers (Kovic et al. 2008; Meystre et al. 2008; Friedman et al. 2002), the literary composition of medical social media has unfortunately not yet been analyzed with the same degree of precision.

Table 2.1 summarizes the linguistic characteristics of these three text types, showing how the first two differ from medical social media:

1. *Clinical texts* comprise documents produced by physicians (e.g., discharge summaries, chart reviews, and operative reports), which are mainly produced to fulfill the physician's reporting duties and to document their diagnosis and treatment of patients.
2. *Biomedical texts* refer to biomedical literature, where researchers are presenting the results of clinical studies.
3. *Medical social media* consists of textual content that is made available by either patients or healthcare professionals on the Web using social-media tools such as blogs or forums.

The content and language of medical social media depends on whether the author is a healthcare professional or a layperson and on their relationship to the disease, such as a treating physician, a patient undergoing care, or a relative or a friend of a patient. For example, one may find on the Web an assortment of clinical cases⁷ written by healthcare professionals which are similar in content and language to clinical text (that is, short, abbreviated sentences that can be devoid of nouns and verbs). However, the majority of social-media postings are of a rather personal, as opposed to clinical, nature. As a result, they are often written in a narrative fashion. Since a Weblog consists of a kind of personal diary, it is not surprising that in this kind of media outlet one often finds that personal opinions are freely expressed. In such postings one can find a large number of personal pronouns (e.g., “I saw ...”, “I experienced ...”, or “My daughter is sick”). Furthermore, rather long sentences may also be prevalent, containing a wealth of adjectives to describe situations, experiences, impressions, etc. Postings are written in common everyday language—even using language that is modern and hip—so as to appeal to the average reader. An example from the multiple sclerosis blog *Stellarlife*⁸ illustrates this point: “Yesterday I finally got to see an orthopedic, okay hold the buggy,

⁷e.g., <http://clinicalcases.org>.

⁸<http://dj-astellarlife.blogspot.de/>.

Table 2.1 Linguistic characteristics of clinical texts, biomedical texts, and medical social media

| Text type | Clinical text | Biomedical text | Medical social media |
|--------------------|--|--|---|
| Sentence structure | Ungrammatical sentences | Often literature style (long sentences) | Rather long sentences |
| | Short, telegraphic phrases (<i>Aspirin or Fever</i>) | Academic | |
| Word usage | Often without verbs or other relational operators | Grammatically correct formulated sentences | |
| | Word compounds (<i>high blood pressure</i>), formed ad hoc | Frequent use of passivization | Adjectives |
| | Modifiers are related to temporal information (e.g., <i>sudden</i>), evidential information (e.g., <i>rule out</i> , <i>no evidence</i>), severity information (<i>mild</i> , <i>extensive</i>), body location | Word compounds (e.g., <i>high blood pressure</i>), formed ad hoc | Descriptive and narrative words |
| Spelling | Misspellings Abbreviations, acronyms | Nominalization and noun compounding Correctly written words | Abbreviations Misspellings |
| Language | Mix of Latin and Greek roots with corresponding host language (German, English, ...) | Scientific language and writing | Common language, rather than domain-specific language or clinical terminology |
| | Domain-specific language | Mix of Latin and Greek roots with corresponding host language (German, English, ...) Domain-specific language | Host language |

I just googled the guy and he is a PA-C/MPAS!! Shut the front door! That means: Physician's Assistant-Certified/Masters Physician's Assistant Studies—huh.”

While the intent is to use everyday language, the insertion of medical terminology in the posting actually depends entirely on the content of each posting. So, for example, when a person is writing about their experiences with a disease, the corresponding terminology may be used, sometimes even with explanations of the clinical terms. Here is a snippet from Diabetesmine⁹ to illustrate this pattern: *No one is sure what causes this dead-in-bed syndrome, but the theory is that a nighttime low blood sugar—called a nocturnal low—episode triggers some kind of fatal cardiac arrhythmia.*

Another distinction between social-media postings and clinical texts is that in the former, abbreviations are often explained in the text of the posting itself. In addition, sometimes words or word phrases are typographically highlighted. This can be done via the use of full caps, quotation marks around a word or a word phrase, or other stylistic means of allowing part of the text to stand out more saliently. We draw again from Diabetesmine to make this point: *While many parents are likely relieved to “get a break” and have their kids back in class, this can be a very stressful experience for the parents of Children With Diabetes (CWDs) who have a LOT more to worry about than just textbooks and extracurricular activities.* In this sentence, the word “lot” was highlighted by using capital letters. For automatic language processing programs, the use of full caps can become a problem, since words that appear in upper case are often considered abbreviations by such programs.

Automatic language processing program can likewise become confused when bloggers insert in their postings a verbatim quote from another person, often a high-profile scientist or public official who has something to say about a major health issue. In *Diabetesmine*, we found that the blogger inserted a quote from a high-profile scientist to support her own contention about the danger of sugar levels dropping precipitously at night: *“We’ve now known for decades that (overnight) is the most common time for severe hypoglycemia,” says Dr. Irl Hirsch, assistant professor and endocrinologist at the University of Washington, and a type 1 PWD himself.* In short, the problem with automatic processing of texts that contain verbatim quotes is that in order to have a correct reference to the quote a processing algorithm must correctly link the cited person, and not the blog author, to the quote itself. Co-reference resolution algorithms are required for this task.

Not only do blogs contain certain kinds of conversational features, as illustrated above, but they are also very much prone to short block paragraphs and bullet point itemization. Bloggers have been found likewise to give headings to their postings as a way of categorizing the content of their blogs. In health forums, categories are often formulated by the forum host in advance. In such cases, forum contributors just scroll down, adding their posting to one of these predefined categories. These categories may cover a wide spectrum of healthcare topics such as how patients (and their relatives, caretakers, and friends) cope with debilitating

⁹<http://www.diabetesmine.com>.

diseases, treatment plans, drugs and medical devices, health insurance coverage issues, or even suggestions of useful resource material on combatting illnesses and diseases, such as new books, informative radio and television programs, as well as television documentaries on the topic at hand.

In short, tools for the automatic processing of medical social media need to consider the range of stylistic preferences in the presentation of blog and forum content, the syntactic features characterizing blog communications—which can be notoriously long-winded, ungrammatical, and idiosyncratic as displayed by the use of full caps and italics for emphasis—along with other features emerging indigenously in this new form of communication. We briefly showed how medical social-media postings differ from clinical texts and biomedical text. Unfortunately, at present, the existing text-mining tools are best suited for processing clinical and biomedical texts, where the language usage and content differ significantly from medical social media.

Extracting Information from Text

Structured or coded data is required for extracting relevant information from medical social-media postings and to make it accessible to humans in a useful and understandable way. Consequently, well-designed natural language processing methods are sorely needed for the extraction of information. But first, let's understand what information extraction entails.

Information extraction identifies facts or information in texts (Grishman 1998). Named-entity recognition (NER) is a subfield of information extraction which aims at identifying within a collection of text all of the instances of a name for a specific type of thing (Cohen and Hersh 2005). Examples include names of diseases and illnesses, drugs, persons, or locations. A potential use case of NER is to acquire metadata from texts for user modeling (Barla and Bielikova 2010) or to support text classification, filtering, or information retrieval (Denecke 2012).

Entities can be recognized in natural language text in two ways:

1. A simple lexicon lookup
2. Extraction patterns that are either manually created or learned from training corpora using supervised machine learning techniques

Lexicon lookup approaches search for matches with words of a lexicon of named entities in a given text. Difficulties are found to arise, namely, because there is no complete dictionary for most types of medical or biomedical entities. Therefore, the simple text-matching algorithms that are commonly used in other domains are not sufficient here. In extraction pattern-based approaches, patterns such as “[Title] [Person]” for the extraction of a person name (e.g., “Mr. Warren”) are generated either by hand or by supervised machine learning techniques. Manual rule-based approaches can be very efficient, but unfortunately such systems require manual efforts to produce the rules that govern them. Machine learning techniques on the

other hand that don't require costly human annotators do however require large training corpora to train their underlying models.

Despite these difficulties, there are tools available for extracting named entities from medical social-media postings which have been originally developed for clinical and biomedical texts. Such tools are premised on different methods and lexical resources. Below, we provide a brief overview of the lexical resources available for processing medical texts and present two NER tools which consider the linguistic peculiarities of such texts.

Knowledge Resources for Processing Medical Texts

Unified Medical Language System (UMLS¹⁰) is composed of three main knowledge components: Metathesaurus, Semantic Network, and SPECIALIST Lexicon.

- The UMLS Metathesaurus integrates vocabularies from the biomedical domain (e.g., Medical Subject Headings (MeSH), Systematized Nomenclature of Medicine (SNOMED CT)) and provides a mapping structure between them. Each concept has specific attributes that define the meaning of the concept. In the release version 2012AA, the Metathesaurus comprises 2,669,792 concepts in 21 languages. The vocabularies that are integrated into the UMLS contribute thesaural relationships between concepts (e.g., “child” or “parent” relationships). Each concept is assigned to at least one semantic type of the Medical Semantic Network (MSN). The UMLS MSN (McCray 2003) is a network of general semantic categories or types where semantic types are linked by relationships. It provides 134 semantic types that have been aggregated into a set of 15 semantic groups to reduce complexity (McCray et al. 2001) (e.g., the concept *atrial fibrillation* belongs to the semantic types *Finding* and *Pathologic Function* that in turn belong to the semantic group *Disorders*).
- Semantic network relationships connect UMLS semantic types to each other. For example, the semantic type *Body Part, Organ, or Organ Component* is associated with the semantic type *Body Location or Region* by the relations labeled *location of*, *has location*, and *adjacent to* with.
- The SPECIALIST Lexicon provides linguistic *knowledge*. For example, syntactical information on (medical) terms, and natural language processing tools, such as a tokenizer which splits a sentence into tokens, are part of the SPECIALIST Lexicon.

As mentioned before, the UMLS integrates various vocabularies including MeSH and SNOMED CT.

¹⁰<http://umlsinfo.nlm.nih.gov/>.

*MeSH*¹¹ is the controlled vocabulary thesaurus of the National Library of Medicine which is used for indexing articles for their digital library known as PubMed. MeSH consists of sets of terms, naming descriptors in a hierarchical structure. At the most general levels of the hierarchical structure are the very broad headings, such as “*Anatomy*” or “*Mental Disorders*.” More narrowly defined headings for these general terms are found at the more restricted levels of the hierarchy. For example, terms such as “*Ankle*” and “*Conduct Disorder*” reflect refinements of the categories of anatomy and mental disorders, respectively.

*SNOMED CT*¹² is a multilingual collection of medical terms. It contains more than 311,000 active concepts and around one million relationships. SNOMED CT provides general terminology for the electronic health record, consisting of concepts, descriptions, and relationships:

- Concepts represent clinical ideas, such as “*neoplasm*” or “*abscess*.”
- Descriptions link appropriate human-readable terms to concepts.
- Relationships link each concept to other concepts that have a related meaning. As such, relationships provide formal definitions as well as other characteristics of concepts.

Tools for Extracting Medical Information

The vocabularies and terminologies introduced in Section “[Knowledge Resources for Processing Medical Texts](#)” can become relevant for identifying instances referring to medical entities by NER tools. Two tools are described below that are based on the UMLS.

1. The *MetaMap System* (Aronson 2001) is provided by the National Library of Medicine. The tool maps natural language text to concepts of the UMLS Metathesaurus. MetaMap follows a lexical approach and works in several steps. First, it parses a text into paragraphs, sentences, phrases, lexical elements, and tokens. From the resulting phrases, a set of lexical variants is generated. Candidate concepts for the phrases are retrieved by lexicon lookup from the UMLS Metathesaurus (version: UMLS 2012) and evaluated. The best candidates are organized into a final mapping in such a way as to best cover the text. Precision of MetaMap, which is the fraction of retrieved concepts that are relevant, was assessed for different text types already, namely, for respiratory findings (Chapman et al. 2004), mailing lists (Stewart et al. 2012), and figure captions in radiology reports (Kahn and Rubin 2009). The precision for these text types ranges between 56 and 89.7 %. Figure 2.1 shows an example mapping of MetaMap for a given sentence.

¹¹ <http://www.nlm.nih.gov/mesh/meshhome.html>.

¹² <http://www.ihtsdo.org/snomed-ct/>.

Example of MetaMap mapping result:

Input sentence: *Laboratory results in a patient with pneumonia, septic shock, and acute renal failure.*

Concept mapping:

laboratory (Laboratory domain) [Functional Concept]

Results (Result) [Functional Concept]

patient (Patients) [Patient or Disabled Group]

Pneumonia [Disease or Syndrome]

Septic Shock [Pathologic Function]

Acute Renal Failure (Kidney Failure, Acute) [Disease or Syndrome]

Fig. 2.1 Mapping example of MetaMap for sentence “Laboratory results in a patient with pneumonia, septic shock, and acute renal failure.” Semantic types are shown in parentheses, while semantic groups are shown in *square brackets*

2. The *Dragon Toolkit* (Zhou et al. 2007) is a Java-based development package for information retrieval and text mining. It provides a linguistic parser, text clustering and classification algorithms, and an ontology-based biomedical text annotator. The biomedical text annotator MaxMatcher, which is part of the Dragon Toolkit, uses a generic extraction approach (referred to as “approximate dictionary lookup”) to cope with term variations. The basic idea of this approach is to capture the significant words only, rather than all the words associated with a particular concept. MaxMatcher has already been evaluated on biomedical abstracts collected from MEDLINE. A precision rate of 71.6 % and a recall of 75 % were achieved (Zhou et al. 2007). The underlying terminology of MaxMatcher can be varied; and either the UMLS or the MeSH thesaurus (version UMLS 2004AA Version) can be used.

Tools in Practice

Ironically, while for online news a comparison of several NER tools (e.g., Alchemy API, DBpedia Spotlight, OpenCalais) has already been performed (Rizzo and Troncy (Rizzo and Troncy 2012)), there are yet no such similar comparisons made of NER tools for medical social media. As such, evaluation results of NER tools in the medical domain are only available for extraction from clinical or biomedical texts while not for medical social media. In this section, we present results of a qualitative comparison of the two described UMLS mapping tools, MetaMap and Dragon Toolkit, which are applied to medical social media.

Method

We applied the two tools to (1) twenty texts drawn from the forum “This MS¹³,” where patients with multiple sclerosis are discussing their problems, and (2) blog postings from “WebMD¹⁴,” where physicians are writing about topics related to health and medicine. The results were checked manually, sentence by sentence. The assessment of the output of the tools comprised:

- Judging presence of the detected named entity (*present* in the text or not)
- Judging relevance of the detected named entity (*relevant* or *irrelevant*)
- Judging the type of the detected named entity (*correct* or *incorrect*)

We identified words that are crucial for understanding the text or the sentence which could not be identified by either one of the tools used. The objective of the assessment was to give insights into the possibilities and limitations of these tools when they are applied to medical social-media data. MetaMap was run with UMLS 2012AA, while the available Dragon Toolkit was based upon UMLS 2004AA.

Observation on Quality of Information-Extraction Tools

The main observation for both tools is that the produced mappings do not contain concepts for all content-bearing terms that are used in medical social media. While medical terms are mostly reflected in the mappings, descriptive or concept-relating words are missing. Detailed observations are provided below in the discussion of both tools.

Mapping Observations of MetaMap

Terms from common language or consumer health vocabulary (CHV) referring to medical concepts are often mapped incorrectly by MetaMap or are even missing in the mapping altogether. Wrong mappings occur in particular for personal pronouns: “I” is mapped to “Iodides [Inorganic Chemical]”; “my” is mapped to “Malaysia [Geographic Area]”; and “she” is mapped to “SHE gene [Gene or Genome].” Verbs are often not mapped at all or are wrongly mapped. For example, the verb “found” in the sentence “Keratin is found in your hair” is mapped to (clinical) “Finding”; or the verb “go” is mapped to the concept “GORAB gene [Gene or Genome].” Keeping the meaning of verbs after mapping is extremely important for interpreting a text automatically (and also manually).

¹³<http://www.thisisms.com/forum/daily-life-f35/topic20839.html> (Section “Daily Life”).

¹⁴http://rssfeeds.webmd.com/rss/rss.aspx?RSSSource=RSS_PUBLIC.

Adjectives can also be incorrectly mapped or, like verbs, adjectives may not be mapped at all as in “*nasty*” or “*embarrassing*” which can drop off the mapping altogether.

Another class of wrongly mapped lexical items is that of words or word phrases used in free text (nonclinical texts) which are errantly mapped to clinical phrases. Some of the most common errors are words or word phrases such as “*of course*” which is mapped to “Course [Temporal Concept],” “*Hi*” which is mapped to “*ABCC8 gene [Gene or Genome]*,” or “*Thanks*” which is mapped to “*TNFSF13B wt Allele [Gene or Genome]*.” In addition, numeric expressions can stump mapping programs as they require separate processing. MetaMap, for example, destroys the expression “*about two to 2 1/2 months*” which is mapped to two concepts: “*Two [Quantitative Concept]*” and “*month [Temporal Concept]*.”

And yet another problem is that MetaMap often provides multiple mappings which may differ significantly in regard to the underlying concepts of which they are comprised. The reason for this is that because words can have different meanings this often results in various possible mappings to concepts with different semantic types. No doubt, having multiple mappings available becomes a problem. MetaMap provides confidence values for these different mappings. Nevertheless, when there are several mappings with the same confidence value, it remains a question of how to select the “correct” mapping automatically.

Mapping Observations of Dragon Toolkit

For the mapping of the Dragon Toolkit or the underlying mapping algorithm MaxMatcher we have made the following observations: In general, Dragon Toolkit maps less terms to UMLS concepts than MetaMap. However, even though it maps fewer terms, the medical terms that it maps to concepts are for the most part correctly identified by Dragon Toolkit. For instance, “*cat scan*” is mapped to “*cat scan [Diagnostic Procedure]*,” and “*MS*” (multiple sclerosis) is mapped to “*ms [Disease or Syndrome]*.” Nevertheless, there are some abbreviations that are wrongly mapped as in “*edss*” (actually referring to *Expanded Disability Status Scale*) which is mapped to “*edss [Amino Acid]*.”

Compared to MetaMap, it can be said that MaxMatcher fails in finding the correct medical concepts for compounded words. For example MetaMap maps “*chlamydia pneumoniae*” to “*Pneumonias, Chlamydial (Chlamydial pneumonia) [Disease or Syndrome]*” and “*Rickettsia*” to “*Rickettsia [Bacterium]*,” whereas MaxMatcher identifies no exact matching for either term.

Similar to MetaMap, verbs are frequently not recognized or are incorrectly mapped by MaxMatcher. For example, “*write*” is mapped to “*write [Occupation or Discipline]*.” In addition, nonmedical locations are also mapped wrongly, as in “*toilet*” which is mapped to “*toilet [Therapeutic or Preventive Procedure]*,” or “*baths*” which is mapped to “*baths [Therapeutic or Preventive Procedure]*.” Given that there are different meanings of terms and MaxMatcher only provides one mapping suggestion, it is not surprising that words can be incorrectly mapped.

This is in contrast to MetaMap which provides multiple possible mappings when several semantic types are possible.

Contractions may be wholly unrecognized by Dragon Toolkit. For example, “*don’t*” is mapped to “*don [Organic Chemical]*.” In addition, MaxMatcher also has difficulties in identifying terms referring to medical devices. For instance, the term “*pump*,” as in “*I have a baclofen pump that usually controls the pain & spasms*,” is not detected by Dragon Toolkit when in contrast MetaMap maps this term correctly as a medical device. Finally, qualitative concepts or adjectives often remain undetected by MaxMatcher. This is unfortunate because such modifiers can be of important medical significance as in “*increasing severe spasms & intense pain all day & night*” where the terms “severe” and “intense” carry much weight.

Discussion and Future Challenges

The assessment of the mappings showed that the NER tools still have problems in processing medical social-media data. In particular, both tools fail in mapping or produce wrong mappings for verbs, personal pronouns, adjectives, and connecting words. Clearly, these terms or at least their meaning and the relationships they infer are relevant for interpreting the content of a sentence and text. Since persons are describing their own personal experiences and observations in medical social-media data, the language they use inevitably includes to a large extent verbs that describe activities of persons and personal pronouns; consequently, it is crucial not to lose the meaning of these personal accounts from patients or healthcare professionals while engaging in automatic processing of blog or forum content. Whereas missing or wrong mappings are not necessarily an algorithmic problem, they might be a problem of the underlying knowledge resource. For example, there is no concept representing the verbs *warn*, *recommend*, and *cause* or the adjectives *horrible*, *miserable*, or *ineffective* in the UMLS, the language resource on which the tested tools are based. This is due to the fact that the terminology has been developed to formalize clinical knowledge, and thus the meanings of verbs or adjectives that are commonly used in medical social media are unfortunately not covered by this terminology.

One must take into consideration that authors of drug ratings, medical procedures, and other social-media content often have no medical training. As a result they often do not use the proper medical terms, but paraphrase these concepts instead. People frustrated with their medical conditions may use a metaphor to refer to their maladies. For example, a cancer patient wrote: “*The beast is going to kill me.*” While the metaphor “beast” is not normally considered as synonym for *cancer*, this is what the patient used to refer to his illness. What we can see from patients’ everyday usage of language to describe maladies is that the classical synonyms for medical terms that exist in biomedical ontologies may be wholly insufficient for the data considered here. Consideration of metaphors, paraphrases, and other ways that the lay population refers to illnesses and diseases could be a substantial extension

of these ontologies when applying such extraction tools that rely upon biomedical ontologies to mine medical social-media postings. Given that relevant meanings that conform to how patients articulate their symptoms are readily provided in more common vocabularies such as WordNet or CHV, some of the possible ways of improving the quality of mapping tools when processing medical social-media data are to consider additional knowledge resources or in the alternative to exploit a more general terminology. CHV which link everyday words and phrases about health (e.g., *heart attack*) to technical terms or jargon used by healthcare professionals (e.g., *myocardial infarction*) (Zeng and Tse 2006; Zeng et al. 2007) might in fact serve as a template for improving mapping tools for use in medical social media. In fact, the open-source, collaborative CHV initiative¹⁵ tries to develop a CHV for consumer health applications which is intended to complement existing knowledge in the UMLS.

Interestingly enough, in addition to terminology extensions such as those found in the CHV that augment the nomenclature of the UMLS, other improvements can likewise be made to mapping tools that are used in the medical social-media setting: (1) By including general terminological resources such as WordNet, meanings of adjectives could be recognized and considered in the analysis. (2) Another possibility against wrong mappings of medical social-media postings is to enhance the underlying ontology, but this must be done cautiously as it is a very complicated process and could probably lead to problems in processing professional language. (3) A third possibility for an improved mapping or for improved NER is the extension of the mapping algorithm. Aronson et al. showed that it is possible to apply successfully an ensemble of classification systems originally developed to process medical literature on clinical reports (Aronson et al. 2007). Such approaches need to be assessed in the future to develop a better suited mapping tool for medical social media.

In fact, various mapping tools could be used together. For example, there are additional tools for mapping to medical vocabularies available, such as BioLabeler¹⁶ or Open Biomedical Annotator¹⁷ (Jonquet et al. 2009). There are also other NER tools that are using their own underlying terminology and do not refer to existing biomedical ontologies (e.g., OpenCalais,¹⁸ LingPipe,¹⁹ cTAKES²⁰). Further, Open Information Extraction techniques (Etzioni et al. 2011) could help in identifying relevant relations as they are expressed by verbs in medical social media. This extraction paradigm learns a general model of how relations are expressed based on unlexicalized features such as part-of-speech tags (e.g., the identification of a verb in the surrounding context) and domain-independent regular expressions

¹⁵<http://samwise1.partners.org/CHV>.

¹⁶<http://www.biolabeler.com>.

¹⁷<http://biportal.bioontology.org/annotator>.

¹⁸<http://www.opencalais.com/>.

¹⁹<http://alias-i.com/lingpipe/>.

²⁰<https://wiki.nci.nih.gov/display/VKC/cTAKES> + 2.5.

(e.g., the presence of capitalization and punctuation) (Miller and Pole 2010). By making use of such extraction techniques, it would no longer be necessary to specify in advance the relevant terms or patterns found in social media. This approach may prove more practical given the fact that medical postings are fast-changing, thus making it simply impossible to continuously update the language of social media and their underlying lexical resources manually. Such an approach of open information extraction could help to identify relations expressed by verbs in medical social media which is so far impossible to do using existing mapping tools. To avoid wrong mappings of personal pronouns or connecting words, negative lists could be exploited, i.e., lists that instruct the algorithms not to map the listed words at all.

In sum, there are a number of obstacles that automatic processing tools must overcome in order to make better use of the richness of data found in medical social-media postings. Nevertheless, some of the methods we've analyzed in this chapter augur well for getting closer to meeting such challenges head on. In the end, better data extraction methods for medical blog content insure a healthier patient population and a more efficient healthcare delivery system.

References

- Aase L, Goldman D, Gould M, Noseworthy J, Timimi F (2012) Bringing the Social-media Revolution to Health Care. Mayo Foundation for Medical Education & Research, United States, 2012
- Altarum Institute (2012) Social-media and Health Care: Applications for Aging and Advanced Illness Populations. Highlights from Duke University's 07–08 May 2012, Durham, U.S., <http://www.dukehsac.com/files/2012/09/CHAPI-Social-Media-and-Health-Care-Paper-1.pdf> [downloaded October 25, 2012]
- Aronson A (2001) Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. *Proc AMIA Symp* 2001:17–21
- Aronson AR, Bodenreider O, Demner-Fushman D, Fug KW, Lee VK, Mork JG, Névél A, Peters L, Roger WJ (2007) From indexing the biomedical literature to coding clinical text: experience with MIT and machine learning approaches. *ACL, Workshop BioNLP, Prague, Czech Republic*
- Barla M, Bielikova M (2010) Ordinary web pages as a source for metadata acquisition for open corpus user modeling. In: White B, Isaías P, Andone D (eds.), *Proceedings of the IADIS International Conference on WWW/Internet*. (Timisoara, Romania). IADIS, 2010, pp 227–233
- Boulos MNK, Maramba I, Wheeler S (2006) Wikis, blogs and podcasts: a new generation of web-based tools for virtual collaborative clinical practice and education. *BMC Med Educ* 6:41
- Chapman WW, Fiszman M, Dowling JN, Chapman BE, Rindflesch TC (2004) Identifying respiratory findings in emergency department reports for biosurveillance using metamap. *Stud Health Technol Inform* 107:487–491
- Cohen AM, Hersh WR (2005) A survey of current work in biomedical text mining. *Brief Bioinform* 6(1):57–71
- Denecke K (2012) An architecture for diversity-aware search for medical web content. *Methods Inf Med* 51(6):549–556
- Denecke K, Dolog P, Smrz P (2012) Making use of social-media data in public health. In: Alain Mille et al (eds) *Proceedings of the 21st World wide web conference, WWW 2012, Lyon, France, 16–20 April 2012*, pp 243–246

- Etzioni O, Fader A, Christensen J, Soderland S (2011) Open information extraction: the second generation, mausam. International joint conference on artificial intelligence, 2011, Barcelona, Catalonia, Spain
- Friedman C, Kra P, Rzhetsky A (2002) Two biomedical sublanguages: a description based on the theories of Zellig Harris. *J Biomed Inform* 35:222–235
- Grishman R (1998) Information extraction and speech recognition. In: Proceedings of the broadcast news transcription and understanding workshop, Lansdowne, VA, February 1998
- Hillan J (2003) Physician use of patient-centered weblogs and online journals. *Clin Med Res* 1(4):333–335
- Himmel W, Reincke U, Michelmann HW (2008) Using text mining to classify lay requests to a medical expert forum and to prepare semiautomatic answers, SAS global forum, San Antonio, TX
- Jonquet C, Shah NH, Musen MA (2009) The open biomedical annotator. *Summit on Translat Bioinform* 2009:56–60
- Kahn CEJ, Rubin DL (2009) Automated semantic indexing of figure captions to improve radiology image retrieval. *J Am Med Inform Assoc* 16:280–286
- Kovic I, Lulic I, Brumini G (2008) Examining the medical blogosphere: an online survey of medical bloggers. *J Med Internet Res* 10(3):e28
- McCray AT (2003) An upper level ontology for the biomedical domain. *Comp Funct Genomics* 4:80–84
- McCray AT, Burgun A, Bodenreider O (2001) Aggregating UMLS semantic types for reducing conceptual complexity. *Medinfo* 10(1):216–220
- Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF (2008) Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform* 2008:128–144
- Miller EA, Pole A (2010) Diagnosis blog: checking up on health blogs in the blogosphere. *Am J Public Health* 100(8):1514–1519
- Rizzo G, Troncy R (2012) NERD: a framework for unifying named entity recognition and disambiguation web extraction tools. System demonstration at the 13th conference of the European chapter of the association for computational linguistics (EACL'2012), Avignon, France, 23–27 April 2012
- Stewart SA, von Maltzahn ME, Raza Abidi SS (2012) Comparing metamap to mgrep as a tool for mapping free text to formal medical lexicons. In: Proceedings of the 1st international workshop on knowledge extraction & consolidation from social-media in conjunction with the 11th international semantic web conference (ISWC 2012), Boston, USA, 12 November 2012, pp 63–77
- Zeng QT, Tse T (2006) Exploring and developing consumer health vocabularies. *J Am Med Inform Assoc* 13(1):24–29
- Zeng QT, Tse T, Divita G et al (2007) Term Identification methods for consumer health vocabulary development. *J Med Internet Res* 9(1):e4
- Zhou X, Zhang X, Hu X. Dragon toolkit: incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In: Proceedings of the 19th IEEE international conference on tools with artificial intelligence (ICTAI), Patras, Greece, 29–31 October 2007

Chapter 3

Machine Translation: The Enterprise Point of View

Brian Garr and Vadim Berman

Abstract The opportunity to reduce human translation costs has always been the most prominent value proposition for the continued investment in commercial machine translation. But what about the value to large, multilingual and multicultural enterprises? What value would reducing such costs have for the company with worldwide technical support in Germany that must support a new team in Brazil with real-time text chat? Similarly, what value would it have for the hotelier with hotels in China who needs to integrate language into their world-renowned loyalty program? This chapter reviews some of the history of machine translation in the enterprise, where it has succeeded and where it has failed, and what the future looks like for enterprise use of machine translation.

Introduction

For decades, entrepreneurs and investors have been captivated by machine translation. They have been lured to invest in its development, left stable jobs for start-ups, and plowed life savings into commercial machine-translation companies. That's because translation is a major expenditure, worldwide. Furthermore, when you examine the technology expenditures by Fortune 500 companies, you will find that many of them have explored the potential of machine translation at one time or another.

Despite such persistent fascination, the size of the market for commercial, machine-translation products has not exploded like the rest of the translation market. Organizations and individuals across the globe spend over 30 billion dollars (USD)

B. Garr (✉) • V. Berman
Linguasys, Boca Raton, FL, USA
e-mail: brian.garr@linguasys.com

on all forms of translation.¹ That number includes expenditures on interpreters and human translation; a plethora of machine-translation-assisted acronyms like HAMT (human-assisted machine translation) and MAHT (machine-assisted human translation); and machine translation. Yet, the raw number for machine-translation-software revenue, globally, is probably no greater than 50 million dollars a year.² It is also divided among many small companies because no single machine-translation company owns the lion's share of the market.

In the 1990s, it looked as if a revenue gap like that would never occur. That was when machine translation met the personal computer and it suddenly became available to the masses. Names such as Systran, Globalink, Transcend, and MicroTac (eventually acquired by Globalink) dominated the consumer markets. Boxes flew off of the retail shelves and found their way into consumers' homes and corporations. Around the world, international companies, like The World Bank and IntelSat, believed that machine-translation products costing \$299 (USD) were going to revolutionize the translation industry and put thousands of highly trained translators out of work.

Those boxes usually found new homes at the back of consumers' bookshelves and in corporate storage rooms. The reason is that once it became clear that these products, when they performed at their best, only produced "gist" translations interest in them flagged and sales plummeted.³ In the typical "hype curve," as defined by Gartner, Inc. (see Fig. 3.1), machine translation in the mid-1990s was rushing at the speed of light from the Peak of Inflated Expectations to the Trough of Disillusionment, where it stayed for the balance of the decade.

History

Two technology breakthroughs that occurred in the late 1990s helped bring machine translation out of the Trough of Disillusionment and into a steady climb up the Slope of Enlightenment towards the Plateau of Productivity for business enterprises. The first was an observation about computing technology called Moore's law.⁴

¹Kelly, Nataly, DePalma, Donald A. &. Stewart, Robert G. (2012) "The Language Services Market: 2012" Common Sense Advisory.

²DePalma, Donald A. (2007) "Machine Translation Attracts Eyeballs, Not Software Revenue." December 20, 2007 <http://www.common senseadvisory.com>

³A typical example of a "gist" translation from the 1990s would have looked like "*The parties of opposition took the Treaty, which includes 95 dates with the first action of Peña Nieto by Mexico.*" This example is, but it gives you enough, if you know the domain of the topic, to gather that one of the first actions of the new Peña Nieto Government in Mexico was to commit to 95 milestones of some sort. Certainly enough knowledge, if you follow Mexican politics, to know that you want to understand more.

⁴Moore's law is the observation that the number of transistors on integrated circuits doubles approximately every 2 years.

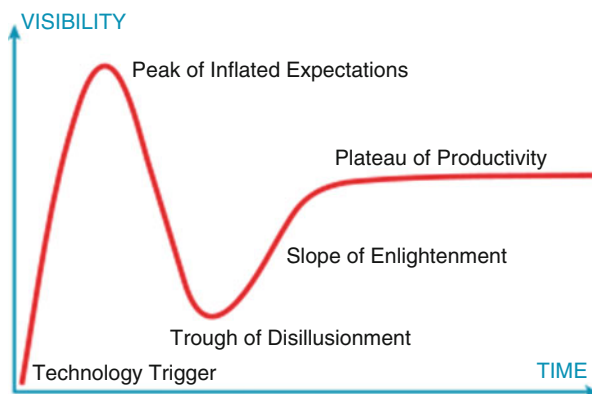


Fig. 3.1 Hype Cycle for technology from Gartner, Inc. (<http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>)

The steady increases in the speed and power of personal-computer chips enabled machine-translation applications to spend more time analyzing source content without appreciably slowing the speed of translations.

The impact of Moore's Law led to the development of commercial machine-translation products called "direct systems," whose name refers to the fact that there was a "direct flow" inside their executables between each language pair. Direct-systems products, like Transcend, became known for their phenomenal speed. They were notorious for performing "lightweight" analysis which was the result of having no mappable, discreet steps for getting from one language to another.

The rise of faster chips also produced the first "transfer systems." The most notable of them was Globalink's Barcelona technology which was designed to have a language-neutral, runtime engine and to have all data required for each language pair to reside within the data files. The Barcelona technology was state of the art for its time. Unfortunately, Globalink never completed the Barcelona language-neutral, runtime engine.⁵ Instead, the company pursued another path outside the norm for machine translation of that time. It allowed its retail package, Power Translator, to pay the bills while it created a client/server version of Barcelona with a lightweight browser-plug-in. Once these were developed, the company marketed machine translation to large corporations as a server-based offering. The concept of a PC-server-based machine-translation product that allowed all lexicons to be shared across the enterprise was revolutionary. One of the first early adopters of this server-based product was Intelsat in Washington, D.C. It is hard to predict what might have happened to the server-based version of Barcelona because, soon after its release,

⁵This was the case while the second author was involved in Globalink. That period extended through the sale of Globalink to Lernout and Hauspie in 1998.

Globalink, short of cash and with a depressed stock price, was sold to Lernout and Hauspie and the entire organization was disbanded.

The second technology breakthrough for machine translation was the use of a statistical technique called Hidden Markov Models (HMMs), which were already being used in speech-recognition systems. HMMs require training. This entails feeding the engine a great deal of data so that it can learn language patterns. Companies, such as Language Weaver, gained prominence using statistical techniques like HMMs to capture and translate source content. Soon there were two distinct camps: one following the statistical gods and the other followed the “tried and true” approach based on linguistic-rule patterns. The two sides did not like each other. At the beginning of the new millennium, if you were a participant in natural language technologies you were either a “statistical” person or a “rules” person, but not both. You picked your side and you suffered the consequences.

The State of the Art

There was a time in the 1990s when you could walk down the aisle of a 747 airliner full of people and you would be lucky to find one person who had actually heard of machine translation. A few things changed that dramatically. The first occurred on December 9, 1997, when Alta Vista, an early player in the Web-portal business, gave away free machine translation supported by Systran. It bore the name Babel Fish, in honor of the mythical fish from *The Hitchhiker’s Guide to the Galaxy* that could interpret all languages on the fly. Soon afterwards, Google released Google Translate, which was based on the statistical approach. Google invested a huge amount of time, resources, and money to “solve” the machine-translation problem. Even today, Google Translate does many things well but, like all machine-translation systems, it has problems with fidelity and accuracy. Nevertheless, the tremendous success of the Google search engine ensured that millions of people were exposed to machine translation and the idea that there could be quick, free translation. Large enterprises started to build test projects around Google Translate to see if they could slay the cost center for human translation. Unfortunately, they found that they could not. Once again, moderate disillusionment was followed by rampant disillusionment.

The situation changed little over the next two decades. Heading into 2010, the major pain-points for machine translation heading remained the same.

- It was too expensive to create a new language pair.
- Too much time was required to create a new language pairing.
- With few exceptions, everything was “paired” with English which served as a “pivot” language.
- No major leader emerged in machine translation, although IBM did release an enterprise product in 2000 called WebSphere Translation Server.
- The out-of-the-box experience was still pretty bad, although considerably better than it had been a decade earlier.

Why is machine-translation software lagging so far behind the rest of the computer world? Were there not enough grants and investments poured into its development? Is the translation of human languages some kind of esoteric art which can be mastered by humans but not converted into the binary computer logic?

The simple answer is that commercialization of linguistic technologies requires more effort than that of most other software. All linguistic technologies differ from software dealing with structured data because the basic tokens in linguistic systems comprise a potentially infinite set. For example, the number of digits and single characters used in a structured-data application, such as an accounting program, is finite; the basic unit of a linguistic system is the morpheme⁶ whose number for a given language is not only infinite, but morphemes can enter into infinite combinations that yield different results.

This doesn't mean that machine translation cannot be done. It would not take long to build an application to translate the sentence "Hello world" into different languages. It would take a bit longer to support a larger but still limited number of words (e.g., 10 or 50) in sentences with uniform structure. This type of highly constrained translation is still often passed-off as a successful proof-of-concept test at the R&D stage. Predictably, the major challenges start when you begin to scale up to handle the infinite set.

This means that the R&D stage has to evolve naturally into a commercialization stage. For commercial products, it's the user who is the final authority on whether a technology works well enough to be adopted. In the case of search-engine ranking or entity extraction, an error may be overlooked or ignored; in the case of machine translation, what might seem to be a reasonably high accuracy of 85 % is unacceptable to most users.

Here is the business catch-22: Given past experience with machine translation, very few enterprises are willing to invest in machine-translation R&D or to buy machine-translation software; yet, without investments and revenues the technology will not evolve fast enough.

The Carabao Technology

In 2010, a small US start-up called LinguaSys set out to change this conundrum. The purposes of their core technology, the Carabao Language Kit, are to resolve the need for large investment, shorten the time requirements for developing new languages, and serve as a playground for testing different approaches. The second author of this paper focused on architecture and flexibility. Rather than creating yet another "better" algorithm he crafted a system that essentially mimics human linguistic processes. That system is called Carabao.

⁶A morpheme is an atomic morphological unit of a language that cannot be further divided.

One important difference between Carabao and other machine-language approaches for the enterprise market is that it does not use English as a pivot language which would require it to perform two translations (to and from English) to get to the target language. Not having to do two iterations of translation creates a faster, yet higher quality, result. This is especially important for applications requiring short, quick communications, such as multilingual chat.

The system employs an “interlingua.” This means that it is not a “direct system” with special sets of executables for each pair of language. It contains internal structures designed to allow translations to occur between any language pairs and directly from one language (for example German) to another language (for example, Japanese). It is a universal, natural-language analysis and transformation engine.

The architecture of Carabao is different from that of most machine-translation systems, including the ones that use an interlingua data structure. Human language is treated as a way to describe real-world entities. Consequently, Carabao provides a superset of what is generally considered to be a machine-translation engine. The Carabao analysis step converts (transforms) the content being processed into a set of references to semantic entities. This step mimics human language understanding. Once created, these disambiguated entities are passed to the calling program to take the next steps. It doesn’t matter what those steps are (e.g., entity extraction, morphological analysis, or another analytical application). The final machine translation (and paraphrase) is an additional step called the “transformation stage” that generates the content of the target language, based on the language-neutral entities that were generated by Carabao’s analytic step.

As a superset of machine translation, the Carabao language can perform many natural language processes but the key is that all of these processes use the same language models. A new language model has the inherent ability to talk with all the other language models and perform all the features and functions as any other language model. This means it allows translation between any combination of languages found in the database. The linguistic data in Carabao is not arranged in pairs of languages because it is grounded in concepts rather than in words and phrases. These groups of concepts are called “families.” A family contains a set of synonym terms and, optionally, their inflections. They are akin to WordNet’s synsets.

Concepts are not specific to a language. This premise allows Carabao to assign unique ID numbers to each concept family and to keep those IDs the same across all languages. Sense equivalence across languages is signaled by these shared IDs. This also means that when a new term is added to a family it works for all combinations and in all languages. Because Carabao is concept based, its language models are not limited to mere lexicons. It also has data tables that contain semantic links between concept families, language-specific morphological data, language-specific agreement rules, thesaurus entries, a table of phonemes and their standard forms (for transliteration and phoneme normalization), and even some commonsense knowledge.

For example, we may want to look for a specific “case” in Japanese. But what kind of “case” did we have in mind? Is it a container to hold or protect something,

legal proceedings, or an instance of a particular situation? Needless to say, different meanings have different translations in the target language. We don't want to look them all up: not only the signal-to-noise ratio is extremely poor, but also the search results for the word senses we do not want might appear confusing to the user (it is not guaranteed that the original terms come back to our source language as "case"). In order to help the user pick only the relevant foreign terms, thesaurus entries guide the user to select the required word sense, and the foreign terms are generated based on this word sense only.

One of the problems with the traditional rule-based approach (and perhaps the main reason behind the push to use statistical processing) is that language-model builders normally must wear two hats. They must be skilled computer programmers and skilled linguists. The number of computational linguists is low and they typically either come from the world of linguistics and, therefore, know only "a little bit" of coding, or they are veteran coders who know only a small amount of linguistics.

The Carabao product addresses this problem by providing a linguistic workbench. Unlike many other linguistic workbenches it isn't a rudimentary tool designed to tweak the dictionary. The Carabao linguistic workbench is what developers use to construct the "core" model for each language. Since Carabao is concept based, the kernel of Carabao is linguistically abstract. That is, it does not know the meaning of linguistic terms like "noun," "verb," or "nominative case." Rather, Carabao's kernel gets that information from the linguistic database. All the linguistic logic is accessible to the linguists designing the linguistic database. Nearly everything can be managed from the linguistic workbench: the lexicon, special characters, punctuation marks, non-break rules, semantic network, detection of non-lexicalized entities, and more. The purpose of keeping the linguistic logic in the database is to ensure that linguists, who know more about the language than programmers, can modify the linguistic concepts, as needed. Since there is no need to ask programmers to modify the kernel, the language-development cycle is greatly reduced.

The Carabao linguistic workbench has a GUI interface (shown in Fig. 3.2). The purpose of creating a GUI tool was to further expedite linguistic development.⁷ This becomes even more important for translations between languages with markedly different linguistic and conceptual elements. Some languages contain grammatical concepts that are not defined in classical, Indo-European linguistics. For instance, most East Asian languages use "classifier words" that are nearly impossible to express in Indo-European terms. African Bantu languages use special noun classes,

⁷Despite all the resources that have been invested in natural language technologies surprisingly little effort has been put into streamlining the way the linguists work. While their peers in other areas of computing have complex IDEs with predictive input and visual diagramming tools, linguists working on machine translation still often have to hand-code formalisms developed decades ago in plain-text editors. This is comparable to attempting to build a skyscraper with a shovel, a hammer, and a chisel. It can be done, but in the age of earth-moving machines this is not a sound approach.

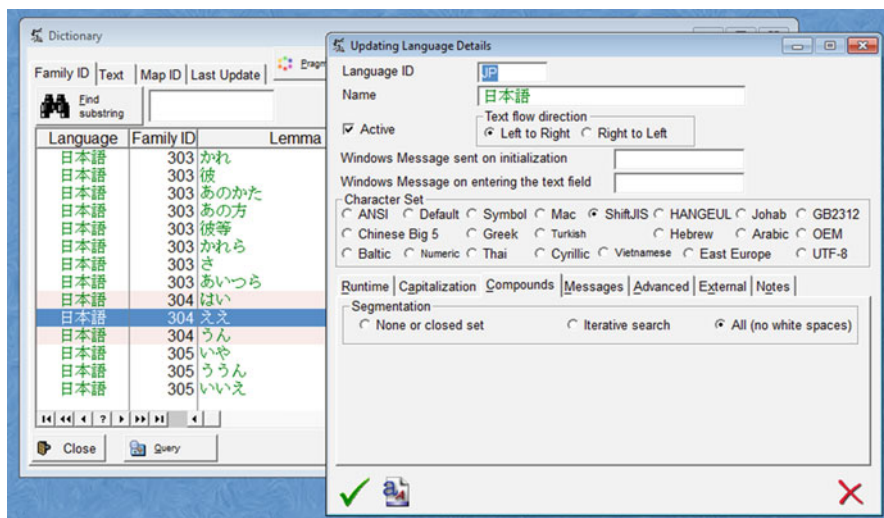


Fig. 3.2 GUI interface of the Carabao linguistic workbench

which are somewhat equivalent to the European noun genders, but not quite the same. Defining such linguistic concepts would take a great deal of work to explain to a programmer, embed into the kernel of a system, recompile, and then release a new version of the kernel. It takes less than a minute in the Carabao linguistic workbench.

In addition to the enhanced productivity and reusability of language models, Carabao's generic semantic network-based design allows the creation of innovative new breeds and combinations of applications. For instance, consider cross-language, information-retrieval based on word-sense lookup. Since every concept in Carabao is associated with a thesaurus entry, the user may refine a search term down to a specific sense.

No intervention by a native speaker is necessary to translate a query. The original content is searched—not a translated or an indexed gist. The same dictionary is used for both terminology lookup and machine translation. This means that there are obscure results or parroting of the originally specified keywords. Moreover, the same queries may be used with different search engines.

Enterprise Case Studies

Since the biggest known revenue-market for translation is human translators, it has naturally become the place to which machine-translation players have migrated. They are working to create a viable machine-aided human translation (MAHT) marketplace so that they can get some of those luscious 31 billion dollars spent

on translation. Too often, a critical market is ignored: large enterprises that need inexpensive and reasonably accurate translations in highly constrained environments. Large enterprises also expect quick responses and immediate solutions to problems that often involve sensitivity to culture as well as languages. We now look at several customer scenarios in large enterprises where Carabao machine translation has been successfully deployed.⁸

Financial Services

A Fortune 500, Financial-Services Firm offering mutual-funds is using machine-translation technology to translate its stock-options application, on the fly, for private customers. They are doing this because their clients' employees want to be able to check their stock options in their own language. That had been a problem because the Web site of the Financial Services Firm was in English. Today, the Carabao technology converts the Firm's Web pages into French, German, Spanish, Japanese, Thai, Malay, Vietnamese, Russian, and Brazilian Portuguese on the fly.

The key to solving the Firm's problem was the creation of customized language models for the highly technical domain of stock options. The customization element was extremely important, since phrases like "strike price" have precise, technical meanings. Once the language models were created the Firm integrated them into their Web-publishing process. "On-the-fly" translation of the client's stock-option Web site went live in early 2011. To date, there have been no complaints about translation quality from end users.

In this example, the constrained domain facilitated high-quality machine translation. This is extensible to comparable domains that involve the translation of Web pages that are full of labels for input fields. These labels tend to remain fairly static. With customization, the quality should always be high. For certain sentences, such as disclaimers, which are obviously very important to financial services firms, the use of a technique called "Translation Memory" (TM) can be highly effective because it provides a perfect human-quality translation every time. TM involves taking professionally translated documents and aligning them with their original language source. By aligning the two, the software can say "if you find an exact match between a full sentence in the new document, and the TM source, use the TM target sentence as the translation instead of going to the MT engine." In this way, Carabao's built-in TM component can be used for highly sensitive areas, such as disclaimers, assuring the clients that their preferred human translation will always be used when that specific sentence is used anywhere in their new documents.

⁸All the customer scenarios date from 2010 and later.

Hospitality

A large, international hotelier had a problem that is, most likely, prevalent in the hospitality and travel industry. Customers in China who wanted to join the hotelier's loyalty program were forced to enter their information in English. It was confusing for customers and may have deterred some Chinese customers from signing up for the program and its benefits. Even though this may be a common challenge facing international hospitality and travel, designing a high-quality solution requires intimate knowledge of the preferred language of the customer and the naming customs of the customer's culture.

Linguasys was not the only machine-translation vendor they consulted but we appeared to be the only vendor who understood that the hotelier did not have expertise in Chinese language and culture. We built cultural considerations into our solution.

Linguasys adopted a multistage approach. We knew that our expertise in Mandarin would help us provide value above and beyond the Carabao technology. That is why we began by having our Asian-language specialist help the hotelier better understand the "rules" of Chinese names, notably, when to use simple transliteration versus translation. Since Carabao technology can do both we were able to offer the client options and to recommend the best solution for their problem than machine translation alone.

We helped them create a workflow and integrated it into their system using our enterprise Web-services interface. Today, their Chinese clients can enter their information in Chinese and the staffs of hotels in countries across the globe know how to address them properly during their stays—even if not one member of the staff of the hotel speaks Mandarin or reads Chinese.

International Banking

Chinese Commercial Codes (CCC), also known as Chinese Telegraph Codes (CTC), was originally used for transmitting Chinese text over telegraph wires. In these codes, series of numbers represent different Chinese characters. Today, CCC is often employed in bank payment, FX, and trade-finance descriptions and is also used to specify the name of a Chinese individual or company without having to use Chinese characters.

Security has caused ripples in this process for financial services institutions. The reason is that names on a "watch list" are usually listed in non-Chinese or Latin script only. A dangerous consequence is that a name written with CCC may be accepted despite being on a "watch list" in another script. Regulators consider this practice of ignoring CCC as a gap in compliance and are starting to crack down. This is forcing financial services institutions to "look up" names sent in CCC to determine whether they are on a watch list. The lookup process is costly in terms of time and money.

A top international bank operating in the Asia Pacific region wanted to streamline the operation of looking up CCC names and thereby reduce the cost of transactions. Carabao's ability to do both transliteration and translation was central to satisfying the bank's needs. Linguasys provided a multifaceted solution. As the first step of the process Carabao converts a CCC numeric code into its corresponding Chinese characters. Then Carabao translates those Chinese characters into a business or a person's name. Both steps are displayed prominently on the screen of the bank's Web-based application so that human analysts involved in the financial transactions can verify its accuracy and take steps to correct any errors. The bank reported that this solution increased the number of transactions that they handle per minute with no loss of accuracy.

Government

The annual International Special Operations Forces conference in Tampa, Florida, attracts Special Ops forces from over 60 countries making it a truly special conference. In 2012, the military organizers of the conference worked with one of Linguasys' partners to create a unique Web site for the conference. A key differentiator for this Web site was its ability to have a central, virtual meeting place—a single location where anyone participating in the conference could go to provide and understand all conference- and location-related communications—in their own language.

The solution combined the Linguasys translation technology with IBM's Same-time Advanced server. The Web site was promoting an "Ask us" chat portal where attending guests could ask anything, such as "where can I find a good Sushi restaurant within walking distance of my hotel," or "what day and time is the big amphibian landing demonstration?" Furthermore, when a user logged onto the Web site and chose a native language, they could read previous questions and answers regardless of the language that had been originally used because everything was translated into their native language.

What's Next?

What we see for the future is the faster and cheaper creation of robust language-models and the ability to use the inherent capabilities of a machine-translation engine like Carabao for a host of natural language processing activities. Among those activities are entity extraction and sentiment analysis. These two operations are typically seen as functions of analytics. Are entity extraction and sentiment analysis also operations of machine translation? In some cases the answer is "yes"

but the answer is probably “no” for the machine-translation systems that are purely statistically based. In a statistical system the concept of “meaning” is foreign. Therefore the concept of “sentiment” is also foreign. For systems that are either hybrid or rules based, the concept of “meaning” is very much alive and well and most of those systems are already two-thirds of the way to creating a viable sentiment-analysis system.

The analytics industry, which itself is almost brand-new, is an important area where machine-translation vendors can see the addressable market expanding well beyond the \$50 million ceiling that has been in place for about a decade. According to Seth Grimes, an analyst who covers the Text Analytics marketplace, the opportunity is upwards of \$835 million globally, with an annual CAGR in the neighborhood of 25–40 %.⁹ A good chunk of that opportunity is addressable by high-end hybrid and rules-based machine-translation systems. In order to catch some of the new opportunities for translation that are coming from areas of analytics, notably entity extraction and sentiment analysis, you need to understand sentiment. This is why most of the machine-translation vendors using statistical approaches are now starting to move towards a hybrid approach.

It is vitally important that the machine-translation industry continue a strong tradition of not overselling the technology. There are many places where machine translation is simply not appropriate and must not be used. Using machine translation to translate medical procedures, for example, is a bad idea because there is no guarantee of 100 % fidelity without human review. While some current uses of machine translation are simply “bad,” some are acceptable, but a poor choice. For example, many analytics vendors choose a machine-translation partner for the purpose of translating text in a designated foreign language into English. They then run their analytics against the translated English corpus. While this approach may give semi-useful results, we believe that it is difficult to put any real credibility into the results. The proper way to analyze foreign language content is in the target language. This will always yield higher quality results.

But the opportunities extend beyond analytics. They are part and parcel of the growth of “big data” and the requirement to aggregate large quantities of information written in non-English languages. Those tasks will be shouldered primarily by the vendors of machine-translation products and technology because they already have rich repositories of non-English data that are well tagged and defined. Machine translation will also continue to make inroads into areas where pure human translation is being used, although that road will be long and difficult. We expect human translators to continue to vilify machine translation and characterize it as utterly useless. At the same time, major corporations are already requiring their localization firms to use machine translation to speed up work and reduce the final price. Sometimes these demands are a by-product of

⁹Grimes, Seth. (2011) “Text-Analytics Demand Approaches \$1 Billion” Information Week Software. May 12, 2011. <http://www.informationweek.com/software/business-intelligence/text-analytics-demand-approaches-1-billi/229500096>

inflated expectations by accountants and CFOs will continue to see it as a panacea for creating higher profit margins. As we mentioned earlier, wholesale replacement of human translators should be done with care and not solely to cut costs.

We expect these technological advances, partnerships with other domains, and overall market growth to continue through the rest of this decade and beyond.

Chapter 4

Speech-Enabled Unified Communications: Overcoming the Multilingual Challenges of the European Market

Detlev Artelt and Hassan Sawaf

Abstract There is a growing need for innovative, multimodal interfaces that can exploit the power of new devices and address the needs of an increasingly mobile world. This chapter describes creative and utilitarian solutions for both consumers and industry that are provided by speech-enabled, unified communication. There are, however, challenging real-world obstacles to implementation. A closer look at Europe, an important potential market for these kinds of solutions, highlights the challenges of overcoming multilingualism and market practices, such as high roaming fees and how such barriers are being addressed.

Introduction

People all over the world are becoming more mobile every day. If the acceleration of innovation for mobile devices continues on its current pace, it will reach every corner of the world. Nearly everybody will carry a huge amount of processing power in their pockets. This trend is even more impressive because it is combined with Internet connectivity which is bringing the world to everyone who owns a mobile device.

Processes and applications that must work in this new world have to be as usable on the road as in the office or at home. The trend towards increased utilization of mobile devices for interactive searching, data entry, and transactions of many types also requires the ability to process different kinds of information quickly and efficiently.

Clearly, mobility and the freedom to work and play from any location are transforming people's lives and the way that businesses operate. For speech and

D. Artelt (✉) • H. Sawaf
Aachen, Germany
e-mail: da@aixvox.com; hassan@sawaf.de

natural-language processing (NLP) this entails the ability to recognize whole sentences and align the given information with the context. It also involves seamlessly blending speech, natural language, and other input–output modalities for which Apple’s SIRI is a prime example.

This chapter begins with an introduction to Unified Communications (UC). That introduction then examines the value of speech-enabled UC for two consumer-focused functions on mobile devices. The section then addresses business and industrial applications. Section III focuses specifically on the European market. It describes three aspects of the European mobile market for speech-enabled UC: personal, business, and entertainment. Each is accompanied by an example. The chapter ends with a look into the future.

Introduction to Unified Communications

The concept of UC integrates different channels, such as speech, chat, video, telephone, fax, and Web, into a single “unified” interaction environment. The user can easily select the interface modality that best communicates whatever is required. This means that all communication in text, voice, and pictures is handled by a single interface which simplifies the day-to-day communication and ensures that tools like Skype, Online Meetings, and Webinars will be used more often. Typical providers of such UC solutions are Avaya, Microsoft, Cisco, and Siemens Communications. What their solutions have in common is that they facilitate the ability to work with subjacent technology and they are very user friendly. What is also interesting is that they are all evolving in the same direction: using speech as a control mechanism. It appears that for UC speech represents an even easier way of using and simplifying communication.

We now move to specific discussions of mobile consumer and industrial applications of UC. The focus is on speech-enabled UC. Table 4.1 provides a comparison of the overall consumer and industrial markets.

The Value of Speech-Enabled UC for Consumers

The movie “13th Street” illustrates a creative use of UC for entertainment. Movie goers with cell phones are asked to provide their mobile numbers as they enter the theater. At one point in the movie the protagonist of the movie is trapped. She finds a cell phone and hits the recall button. It connects her with someone in the audience who tells her how to escape using commands, like “go left” or “go down.” Technologically, the commands issued by the audience member are sent from the cell phone to a centralized server with speech-recognition (SR) technology. The system is able to recognize the relevant words and sends the information to the movie projector, which plays the appropriate response to each command.

Table 4.1 Overview of mobile consumer and industrial markets

| | Industrial speech applications | Consumer speech applications |
|---|---|--|
| User | Familiar with the commands Application is work tool | Often novice Application may be connected to private activity |
| Market driver | Is required to use the application | Switching to alternative applications/channels is easy |
| VUI design | Cost driven: ROI, productivity Small margins (e.g., logistics). B2B Efficiency driven: Short prompts, sticks to existing/familiar process Intelligibility, speed Often either strong system or strong user initiative | Revenue driven B2C Emphasis on pleasurable user experience |
| Price model | Per device Strong bias to capex Value range several 100 to several 1,000 dollars per device | Persona, studio prompt recording Helpful prompts. Mixed initiative, or mildly user/system driven Per voice channel Mix of capex and opex, with bias towards opex for hosting, telephony costs |
| Deployment | Often embedded on ruggedized PDAs | Telephony or cloud (client based), in the latter case with a bias to customer OS and devices: iOS, Android, Windows Phone 7/8 |
| “Why voice?” (voice channel adoption factors) | Issues: Device, hardware, audio I/O OS usually Microsoft Windows CE (5, 6) or Windows Mobile (5 through 6.5) Hands-free, eyes-free operation | Naturalness, universality of voice modality |

Later in the movie, the protagonist calls someone else in the audience to get more instructions. “13th Street” was a commercial success and won awards at film festivals, including Cannes. This example shows how powerful UC using SR can be.

Most speech-enabled UC is directed at task simplification. We turn to two core mobile functions for individual/consumer users: presence management and personal assistant.

Presence Management

An important UC component is the visualization of the “readiness to communicate.” This means that it is possible for every user to indicate on which communication channel he or she is currently available for mutual communication. This not only refer to the fixed-line telephone network. It is “unified.” That is, it spans the entire spectrum of communication, from voice (e.g., cell phones) and softphones (e.g., Skype) to text (e.g., chat, SMS, instant messaging.) and beyond, in nearly real time. This function is called “presence management” and speech can be used in various ways for core mobility function.

An important application of presence management occurs every day for many of us. In the course of a normal workday, one is interrupted by (possibly numerous) phone calls or interactions with other people. Furthermore, a great deal of time is wasted trying to reach other people. Presence management enables a user to see the current status of colleagues. One way this is indicated is through colors. If the person being sought has a communication status of red, no time needs to be wasted by trying to call them. This function is available for not only office phones but also cell phones and softphone in the home office.

Such a presence-management tool can use the individual’s calendar to determine status or it might notice when a landline connection is busy and use that information to change the person’s attendance status for cell and softphone to “busy.” This kind of automated presence management can prevent frustration and wasted time.

Unfortunately, there are still situations for which people must change their “presence status” manually. Such situations may even occur several times a day. Most busy people are loathe to engage in the manual operations required to change their presence status. Consequently, this is an important way in which speech technology supports presence management. A user can easily change their status by simply placing an order on the cell phone. That is, by saying “Change presence status to ‘Meeting’ Duration 1 h.” Use of speech for this type of presence management would produce a huge improvement in acceptance and use of UC and, thanks to smartphones, it will be accessible from any end device.

Personal Assistant

A personal assistant is a device interface standing between the user and the functions on a device. The most well-known and highly touted personal assistant is

Apple's SIRI. A speech-enabled personal assistant like SIRI ensures that presence status can be easily managed via cell phone with speech. But the value of speech-enabled personal assistants goes well beyond presence management. A personal assistant like SIRI enables users to manage many tasks by voice and can be useful in many situations.

For example, most people organize their day-to-day work using a calendar. In doing so, they store a huge amount of relevant information in their calendar systems. Keeping a calendar up to date requires a great deal of work and involves quite a bit of typing. Therefore updating a calendar would be fast and simple if all one had to do to make a new appointment was to say a sentence like "Calendar: New appointment with Bill and Judith for April 14th in San Francisco at Location Hotel Hyatt Regency."

Using a voice-enabled UC solution makes it possible to create new calendar entries using a cell phone as well as a computer. This means that it doesn't matter where the user is at that moment, because the cell phone receives the new entry and saves it in the local calendar. The next time that cell phone connects its calendar with the company's network (e.g., Exchange, Notes) all information will be harmonized.

Personal assistants like SIRI offer conversational interaction with many applications, can answer questions posed by the user, and give instructions to applications that would otherwise require a manual entry.

The greatest benefit of personal assistants is their ability to span applications and functions on and/or used by the device, such as getting the weather forecast, making a note about something, writing an e-mail, or placing a telephone call. It is very easy to use and it works as long as there is a connection to the Internet because the audio data of the speech is transferred to a server where it is converted to text and/or commands and then sent back to the end device.

Speech-Enabled UC as a Business Tool

Companies can also benefit from speech-enabled UC. One example is by using a voice-assisted, corporate mail-sorting. In mail room of larger corporations, incoming mail is manually sorted into pigeon holes according to the information on the envelope. This information is usually limited to the first name and last name of the recipient and, possibly, their division or other location indicators. Using a voice-enabled UC, a mailroom clerk needs only to read information on the envelope. If the information gathered is sufficient, the voice application will respond with the corresponding pigeon-hole number for the recipient. If it is ambiguous or if there are other problems, the voice application will respond with a request for further discriminating information.

The biggest savings companies get from UC are by integrating communication into their business processes. If the company uses a CRM or an ERP solution, it is possible to integrate the communication completely into this system. Because many users save all their contact details and contact history in the CRM, there is no

need to dial a telephone number in order to call someone. For example, a supervisor needing to contact a distributor when the supply of a production material is low can contact the distributor by simply speaking the distributor's name into his or her cell phone which will—thanks to SR—call the right person. This process saves time, accelerates work productivity, and, most importantly, enables just-in-time processes. The result is better production performance.

Another useful application of speech-enabled UC occurs every day in warehouses when employees called “pickers” interact with a speech system to fulfil a customer's order. The speech system utilizes a catalogue of items and quantities in the order (called a “picking list”). Following instructions from the speech system, the picker manually locates the items on the picking list, retrieves the correct quantity of each item, and places it on a trolley, in a forklift, or in a container. Such applications require speed and traceability. These are the very things that speech-enabled UC provides.

Plant scoring is an application of speech-enabled UC which is done in an open field or in a greenhouse. Plants that have been treated with a new chemical (e.g., a pesticide or a fertilizer) must undergo health-status assessments by specially trained plant biologists. The operator follows a predefined procedure but, instead of writing down their findings or typing them into a computer, the operator can record the plant's condition by voice entry which can be immediately uploaded and processed by the central system. Since adjacent plants often display the same conditions, voice shortcuts can be used for multiple entries and path changes.

There are many other examples of speech-enabled UC in inspection, maintenance, logistics, and manufacturing. They include returned goods inspection and sorting, tree health inspection in forestry, spare parts picking, simultaneous inventory and picking, and goods reception.

There are a number of companies involved in supplying speech-enabled UC systems for these types of logistics applications. Over the years, they and their partners have developed an entire ecosystem consisting of UC input and output technology providers, headset and local hardware device providers, and warehouse management system suppliers.

Among the companies participating in this ecosystem is Dawin, a provider of voice-assisted checklists. The Dawin checkMaster is a customizable application that allows the user to design and edit electronic checklists without any programming. Checklists can operate both on- and off-line on PDAs, PCs, and tablets. A typical checklist consists of number/text/date/structured data input fields or multimodal input fields, such as GPS-position acquisition, a photograph, audio snippets, touchpad signature, or a sketch. Figure 4.1 shows the checkMaster's graphic interface and Fig. 4.2 contains a typical checkMaster picking dialogue.

The portable devices used by operators in warehouses and factories are usually ruggedized, and may be subject to extra robustness compliance requirements, e.g., IP 64 or 65 (falls, shocks, breakability). When running a voice application, they are usually stowed in a belt pouch to enable hands-free, eyes-free operations. Typical PDAs include Motorola MC70, Psion Ikon, Pidion Bip 5000/6000, LXE MX7/MX8, Datalogic Memor, featuring around 256 MB RAM and ROM, and an



Fig. 4.1 Photos of Dawin checkMaster (provided by Dawin gmbh)

800 MHz ARMv4i processor. Manufacturers of ruggedized PCs include Handheld, Panasonic, and Fujitsu.

Audio acquisition is typically done via wired or wireless headsets especially in noisy environments, like a factory floor. Wires often need to be worn under protective/professional clothing to avoid accidental strapping. Wired headsets offer wideband speech acquisition and low weight—an important factor when the headset is worn all day long. Wireless headsets are preferred when it is possible to stow the device in a rucksack or a belt case, or attach it to a trolley or a forklift. However, requirements for these wireless headsets are challenging. They include the following:

- Low weight and good head/ear fitting are stringent criteria for user/trade union adoption.
- Robustness, read below.
- Long battery life: For speech applications used all day long, a battery change incurs several minutes of service interruption.
- Wideband audio acquisition, read below.
- Noise cancellation, typically implemented as DSP-level differential noise subtraction.

Robustness requirements may involve the ability to withstand rough handling, bumping, and dropping. Sometimes they have specific requirements. For example, picking in deep-freeze warehouses operating at $-28^{\circ}\text{C}/-18^{\circ}\text{F}$ requires resistance to breath condensation which may freeze on the microphone foam-pad. Outdoor operation will need to withstand rain, wind, and other weather conditions. Such requirements are more difficult to meet.

Icons : 🗨 system speaking 🗨 operator speaking.

- 🗨 Hello. Please say your first name followed by your name.
- 🗨 Neil Jones.
- 🗨 Hello Neil Jones. Please say the picklist number or scan.
- 🗨 25 89.
- 🗨 Please confirm 25 89.
- 🗨 Yes.
- 🗨 List contains 17 references, with a total weight of 32 kilograms. Please take a large combined trolley and read its check number.
- 🗨 238
- 🗨 238. Go to alley Charlie position 6 and confirm check number.
- 🗨 (User goes to position, and reads out the two last digits from the position label) 59
- 🗨 OK. Load 3 boxes and confirm.
- 🗨 3 loaded.
- 🗨 OK. Go to alley Delta position 2 and confirm check number.
- 🗨 (User goes mistakenly goes to position 3). 21
- 🗨 21 is not correct. (speaks slower for better understanding) Please check you are at alley Delta position 2, and confirm check number.
- 🗨 (User realizes his mistake and says correct check digits number). 94
- 🗨 OK. Load 14 boxes and confirm.
- 🗨 10 loaded.
- 🗨 10 loaded, 4 left.
- 🗨 4 loaded.
- 🗨 OK, finished. Go to alley Echo position 5 and confirm check number.
- 🗨 (User goes to position) 84
- 🗨 OK. Load 12 boxes and confirm.
- 🗨 6 loaded.
- 🗨 6 loaded, 6 left.
- 🗨 Exhausted.
- 🗨 Position exhausted, can you confirm please?
- 🗨 Yes.
- 🗨 OK. Go to alley Golf position 3 and confirm check number.
- 🗨 (etc...)
- 🗨 OK. Picklist finished. Please take trolley to station North 3 and scan.
- 🗨 (User goes there, releases trolley for further processing, and scans station label to double check position)
- 🗨 Thank you. Order processed. If you would like to process a new pick list, please say the picklist number.
- 🗨 (etc...)

Fig. 4.2 checkMaster picking dialogue

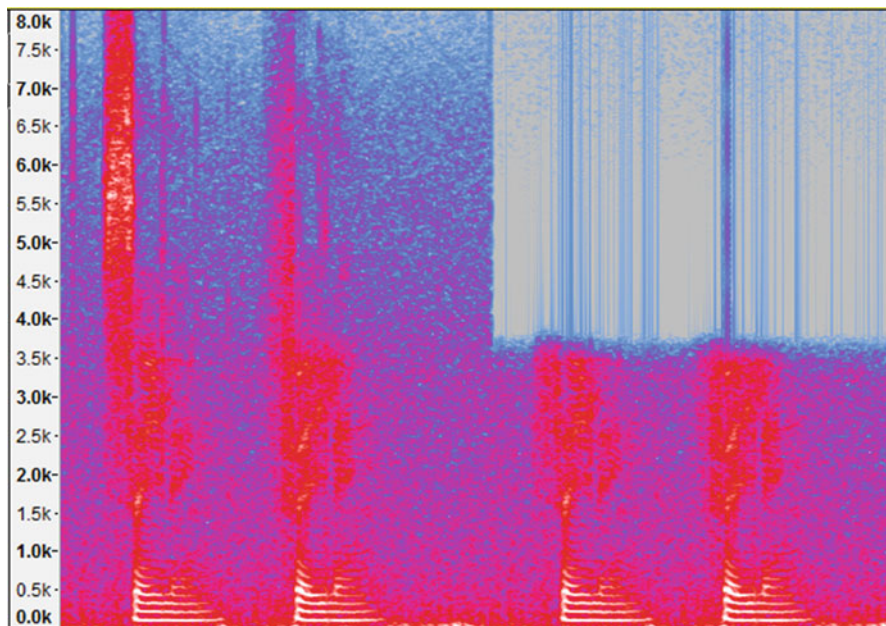


Fig. 4.3 Spectrogram of the words “sailing, failing”

The most popular wireless audio technologies are Bluetooth and DECT. Bluetooth chips are embedded in most ruggedized PDA. The SCO (synchronous connection oriented) Bluetooth protocol stack only supports (full-duplex) 8 kHz sampling (telephony rate), while the high-fidelity rate (44.1 kHz) using the A2DP (advanced audio distribution profile) is reserved for simplex audio broadcasting.

Wideband audio acquisition is a performance factor for SR, as illustrated by the spectrogram in Fig. 4.3. It contains spectrograms for the English words “sailing” and “failing” at 16 kHz sampling rate (aka wideband) and 8 kHz sampling (narrowband, telephony-speech quality). The spectrograms show that narrowband sampling discards discriminating parts of the audio signal. For example, the difference between “s” and “f” is clearly shown in the region which is beyond the 8 kHz-sampled speech. This illustrates why wideband sampling brings higher quality SR and better audio, overall.

Another issue is streaming. There are few wideband, Bluetooth audio-transmission protocols. They are proprietary, lossy, and typically developed for military applications, and require both a sender and a receiver chipset. Even if corresponding headsets were available, the redesign of existing PDAs to include such a chipset would likely be cost prohibitive. Attaching an external chipset to the PDA audio-jack is cumbersome and subject to disconnection. DECT supports a wideband protocol at practically 16 kHz sampling rate, but it is rarely implemented in ruggedized PDA.

Connectivity issues for streaming effect both headsets and devices, and are often the result of the high spatial density of multiple devices operating in the same environment. In addition, large metal racks and walls are ubiquitous in warehouses, causing Wi-Fi blind spots. Wi-Fi and Bluetooth counter such interferences by channel hopping. Furthermore, broken-up, streaming audio is often a direct consequence of spectrum overload. That requires a solution at the network-design level. This ends up being complicated and expensive to solve, especially as warehouse or industrial networks are rarely designed with streaming signals in mind; more often, they are designed for Telnet or some IP packet-oriented exchange with minor risk of collision.

Noise is a challenge for speech-enabled UC. Industrial speech-applications are often used in high-noise environments that may include nonstationary machines, vehicle or material noise, and background speech or music (workers listening to the radio on the workplace). Efficient noise cancellation and voice-activity detection algorithms at the DSP or the software level exist and mitigate recognition errors. DSP-level solutions (e.g., built into wireless headsets) might sometimes produce a repaired signal. While such signals may be highly intelligible for humans they defeat the SR's front-end signal-processing and may result in a lower performance than achieved using software-level processing.

"Speaker-dependent" SR, which is designed to create a voice model for each individual user, can be a solution to noise issues. When combined with word enrolment, or pronunciation tuning, it may improve SR performance, especially for speakers with strong regional dialects and those with non-native accents—a common by-product of today's migrating workforce.

Another issue that faces any new technology is managing customer expectations which range from bias against a technical solution or inflated expectations about that technology. These problems have been mitigated by the movement of speech technologies into the mainstream. The result is greater acceptance of speech-enabled UC.

A related issue is audio logging which is used to tune speech applications. When it is done on cards it slows down the PDA's operation. When that happens, perceived latency is often blamed for any productivity losses that might occur. A typical solution consists of logging onto select devices and collecting audio material over FTP when those devices are idle and network-powered. Compared to a telephony environment, where statistics and recordings are a matter of course, the lack of hard audio performance data (let alone statistics) for embedded deployments can cause controversies and lead to misguided debugging or tuning.

Some industrial applications need to be multilingual. For instance, mail sorting applications in multilingual locations such as Belgium or Canada contain street, city, and person names from two or more languages. Furthermore, a multilingual workforce may require VUIs in the user's native language. Fortunately, current speech-synthesis and SR engines can run in parallel, and cross-lingual situations are efficiently handled by phonetic transfer matrices which encode correspondences between one language's phonemes and another's.

This leads to the next part of this chapter: the European market for speech applications.

The European Market

This section addresses two major challenges to using speech-enabled UC in Europe: multilingualism and exorbitant roaming fees.

Europe: The Continent of Languages

Speech-enabled systems were originally designed with the US market in mind. Consequently, anyone wanting to build a voice application for the European market faces a major problem: the number of languages they must support—especially if they plan to cover the entire continent. With more than 225 indigenous languages, Europe is rich in multilingualism. Even when one eliminates languages and dialects spoken by very small minorities there are still 23 official and working languages in the European Union (EU) alone. These numbers show that one of the biggest challenges for SR in Europe is to overcome the language barrier. This is why, over the years, the European Commission has provided a great deal of financing for research groups working on natural language processing, speech technology, machine translation (MT), and information extraction. Enlargement of the EU and globalized markets for European goods have led to even more EU support for NLP. Private industry in Europe also recognizes the importance of addressing Europe's multilingual challenge.

The Need for MT

One of the primary areas of focus for both the EU and private industry is MT. The need for a user-friendly and easy-to-access translation device is critical because even the three largest languages of Europe are spoken by minorities. There are approximately 450 million people living in Europe. Twenty percent of them identify German as their mother tongue, 12 % are native English speakers, and only 11 % are native speakers of French. Thus, as soon as European citizens leave their homelands they face a language problem. If they travel to rural areas or must speak with elderly people the chance they will encounter a language barrier increases even more. This problem also effects speakers of English, which is the most widely used language in Europe. Of course, this language problem applies to other continents as well. The growing and prospering Asian market is already demanding speech applications in various Asian languages, including—but not limited to—Mandarin Chinese, Japanese, and Korean.

There are a number of ways in which multilingualism has been addressed in Europe. One is to develop MT solutions, some of which are designed specifically for mobile devices. Such solutions are appropriate given the rise of smartphones and other mobile devices. Other applications of MT are designed for larger, real-time applications, such as closed-captioning of media broadcasts.

MT for Closed-Captioning

Closed-captioning requires transcription. Generally, 1 h of spoken material requires between 2 and 3 h of transcription which may be followed by postediting to achieve results that can be used in real-world applications, such as media broadcasts. When the closed-captioning requires translation the time required for humans to produce good captions and subtitles explodes. For example, a skilled human Arabic-to-English translator can generate 400–500 words per hour. Just the translation of 1 h of broadcast data might require 12 human translators (plus at least one editor). The cost and time requirements have led EU researchers to automated SR and MT for closed-captioning.

The idea of automated closed-captioning and subtitling for user-generated videos is not new—even for the Internet. Google has offered this feature for videos uploaded to YouTube since 2009. Many services also offer human postediting to improve the quality of automated services, to improve the output quality of a transcript for the closed-caption text and translation for the subtitle text.

The two major challenges to this approach are that the SR and MT technologies are not tightly coupled and that the solutions do not weave user feedback into the actual workflow.

Consequently, the quality of automated MT systems for closed-captioning remains poor. When human post-editors are added, the quality can improve dramatically but service becomes expensive. The twin challenges of quality and cost have prevented media networks from using captioning solutions or have forced them to pick-and-choose the multi-language content to provide to their consumers.

The Science Applications International Corporation (SAIC) has developed a product, called the Omnifluent Media Subtitling System (OMSS), that, it believes, overcomes these problems. OMSS generates high-quality, closed captions and subtitles and then streams them to live TV shows. The product uses automated SR and MT and human post-editors. When a translation is completed, the application feeds the edits and comments of the post-editors to the automated components as a way of improving these components and the overall system.

Figure 4.4 shows the main components of their system:

The process begins with video and audio captioning which, as its name suggests, provides captions for the live audio stream. It then streams it to the subsequent processes. The segmentation of the audio process is included to improve SR and MT quality. The segmentation includes handling of a range of challenges, including speaker change, channel change, various environments (e.g., studio interview vs. interview on the street), language change, prosody, noise, and music and jingle detection.

The text is extracted from the audio stream using a large-vocabulary SR subsystem which is described in greater detail, below. This system has post-processing features designed to improve overall readability, e.g., capitalization, punctuation, and optional speaker tagging.

Another integral process is postediting of SR by humans. Post-editors who edit and annotate the transcript.

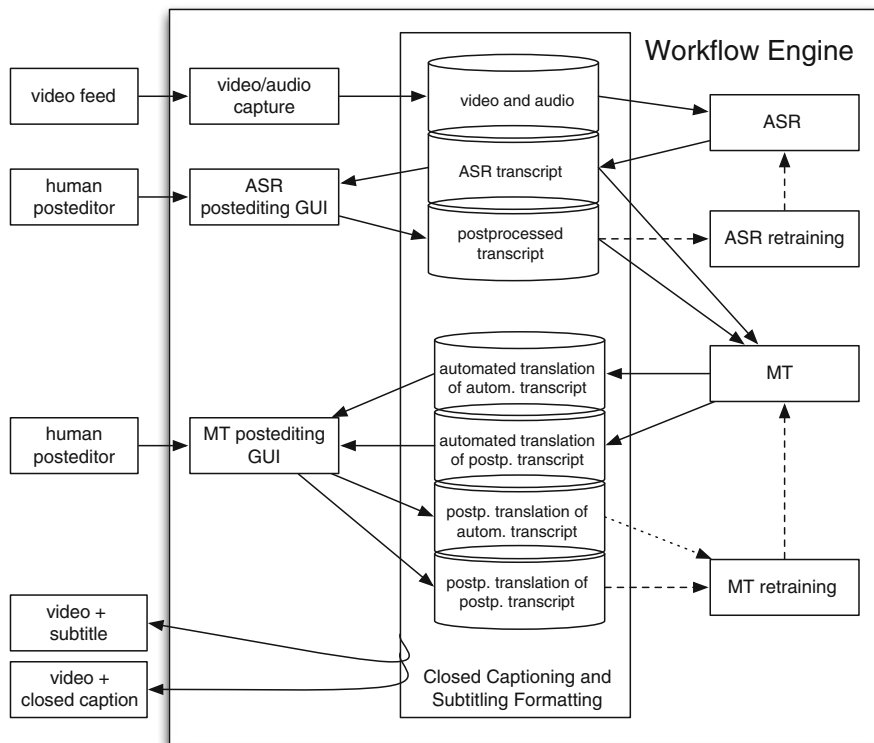


Fig. 4.4 Diagram of main components of the Omnifluent Media Subtitling System

The other major component is MT which, like SR, includes postediting. The MT subsystem converts the post-edited SR transcript into the targeted language. It uses linguistic and meta-linguistic information (e.g., prosodic features and hesitations). The output of MT may be post-edited by humans prior to publication.

The closed-caption and subtitle formatting process adheres to encoding standards. Depending on the needs of the user and consumer this encoding may include nonspeech events or speaker-identification tags. The workflow management process modifies the output so that it conforms to time constraints, resource availability, and source-data quality.

As mentioned earlier, SR is a core component of SAIC's OMSS. It can be implemented software-as-a-service (SaaS)-based, premise based, or as a hybrid combining the two methods. It includes highly trained models for all the languages it supports. For example, the models for Arabic were developed from over 2,000 h of manually transcribed data plus more than 100,000 h of automatically transcribed data that was used for unsupervised training. Since it contains models that are dialect specific, the Arabic language model is dialect adaptive. As mentioned earlier, the SR

system is capable of learning from corrections made by the post-editors. This can be done on three different levels:

Acoustic adaptation techniques incorporate new data about accents and dialects. New words from post-edited transcripts can be automatically added to the pronunciation dictionary and the system can adjust pronunciation using the postediting of the SR output.

The language model can be adapted using the corrected and uncorrected SR. The adaptation happens continuously by using subsampling techniques against a large background corpus. The adaptation uses a range of techniques, including category (e.g., politics vs. sports) style, dialect, and domain.

MT, the other core component, is a hybrid system using statistical search and rules. The hybrid system employs a combination of multiple probabilistic translation models, including phrase-based and word-based lexicons, as well as reordering models and target n-gram language models.

Much like the SR models described above, statistical MT algorithms are trained on large quantities of data which generate probabilities for translation options. During operation, the MT component chooses the phrase or the sentence in the target language with the highest probability to be a translation for the source input.

Syntactic and semantic information are derived from a rule-based engine that generates syntactic trees with semantic and syntactic annotations.

The hybrid MT approach enables OMSS to exploit systematic phenomena that are best captured using abstract rules rather than lengthy statistical sampling and training. This is especially useful for rules that address classes of words or special word dependencies that can be distributed over an entire sentence.

Within this hybrid approach the statistical search process has access to the complete information database that serves the rule-based engine.

The hybridization is accomplished by treating all pieces of information as feature functions in a log-linear framework. Incorporation of these rule-based and statistical knowledge sources is achieved by adding feature functions and by allowing a training algorithm to train the weights of features in context with other features. Translation quality is measured by an error criterion.

OMSS can learn from example sentences and from corrections made by a human post-editor. It can also learn from abstract knowledge coded into language rules by human linguists.

In addition to this learning process, the phrase tables and language models are optimized using adaptation techniques. These techniques include subsampling; multiple parallel-processed, domain-specialized phrase tables and language models; and category- and class-based models. The system is capable of learning about new dialects through phrase tables with post-processed translations combined with automatically generated translations.

Figure 4.5 provides the workflow for increasing the overall processing speed from capturing the audio to generation of the closed captions and subtitles. As the system progresses and improves in quality, the postediting for both SR and MT can be focused on utterances and sentences that have a lower confidence levels.

In media broadcasts some content is known beforehand (e.g., scripts that a news anchor reads from the teleprompter). This content must be processed by the

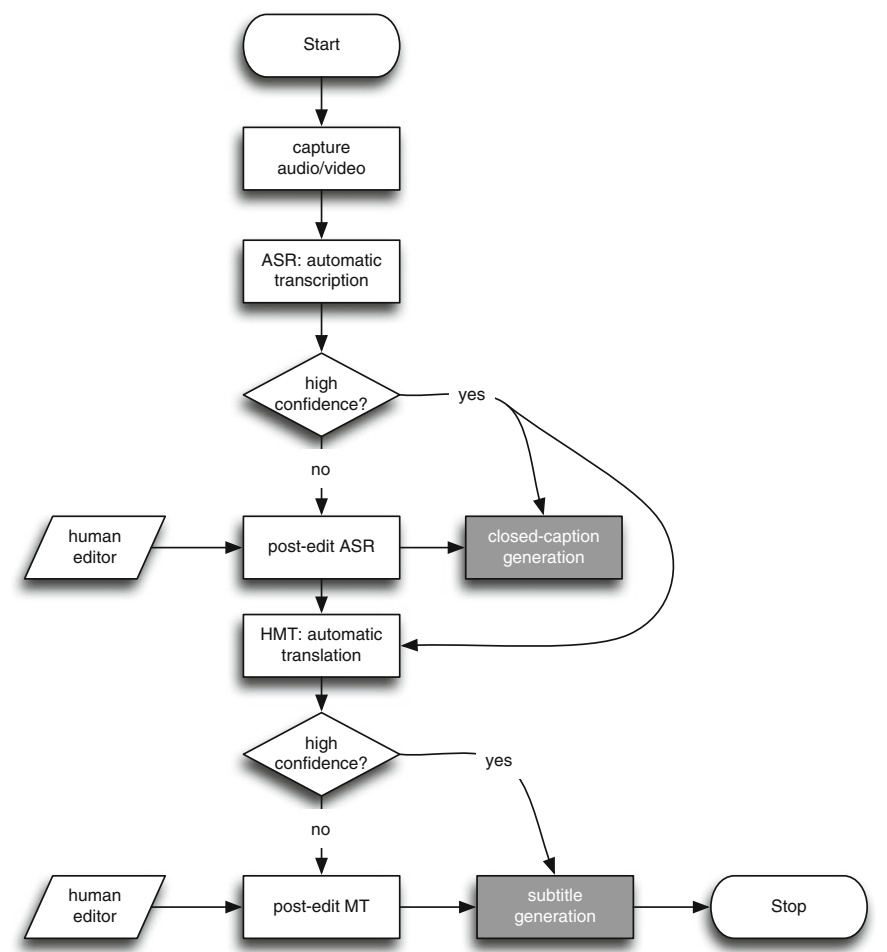


Fig. 4.5 Diagram of statistical search process

workflow separately from the rest of the live broadcast. This adds overhead to the workflow management process and introduces the problem that the uttered speech might not be exactly what appears on the teleprompter. At the same time, the known content facilitates and speeds the automated and postediting processes and, overall, ensures a high level of accuracy.

The OMSS model makes it possible to process data from data capture to delivery of closed captions and subtitling in less than 1 h. In the future, systems like OMSS will add additional functionality that will further increase speed and efficiency. Among that functionality is use of additional knowledge sources for MT and an even tighter integration of SR and MT.

Roaming

Roaming is an issue facing users of wireless mobile devices. It refers to use a mobile device outside of a subscriber's home location. It is a basic wireless need and a service provided by wireless carriers in Europe and elsewhere.

Unlike the USA, each state in Europe—and there are around 50 different ones—has at least three telecommunications-service providers. Traveling through Europe often requires jumping from one provider to another. From a technical standpoint, this is not a problem. Rather, it is a market issue because service providers demand huge fees from roaming users—even when those users are their own subscribers. For example, Vodafone operates in both Greece and the Czech Republic but gives no discounts to its Greek subscribers who are using the Vodafone network in the Czech Republic. Those users must pay the same roaming costs as roaming users who are not Vodafone subscribers and the fees are applied to both making and receiving cell phone calls. Other service providers have comparable policies.

The reason for this is that they counterbalance the providers' inland business losses. The competitive inland market has forced providers to lower prices and to outbid competitors leading to losses for those services and the use of inflated roaming costs to recoup those losses.

This issue is not simply an inconvenience; it is a threat to business because it defeats the path Europe has been taking to eliminate barriers to business development caused by national boundaries.

Since 2007, the European Commission has enacted laws requiring providers to adhere to a set price-frame but roaming costs remain high—particularly for Internet access. In 2007, roaming calls within the EU cost 49 cents per minute (in US dollars) and roaming reception costs 24 cents per minute, plus tax. Today, prices dropped somewhat: 35 cents for placing a roaming call, 11 cents for receiving a call while roaming, and 11 cents for SMS. Access to the Internet, however, is 83 cents per MByte (Fig. 4.6).

The high cost of Internet access has put a brake on SR solutions that require the use of a data network. Consequently, there are few options for roaming users and companies interested in providing cloud-based or other device-independent solutions.

Purchasing a local SIM card is the best available option. They offer low roaming rates. Another option is to call a prepaid service, such as Global SIM in the UK. Such services come with a SIM card with a local UK number. To make a call, the user simply dials the number they want to call and hangs up. After a short while their phone rings. When they answer it they will be connected to the called party at a lower roaming rate.

While this may work for standard mobile calls it is less effective for SR and MT solutions. Most of the current voice solutions use a server in a hosting facility where the speech is processed. Hosting works well for SR technology and solution providers because a service of this type is easy to set up and monitor. It also provides huge amounts of audio that can be used to optimize the SR.

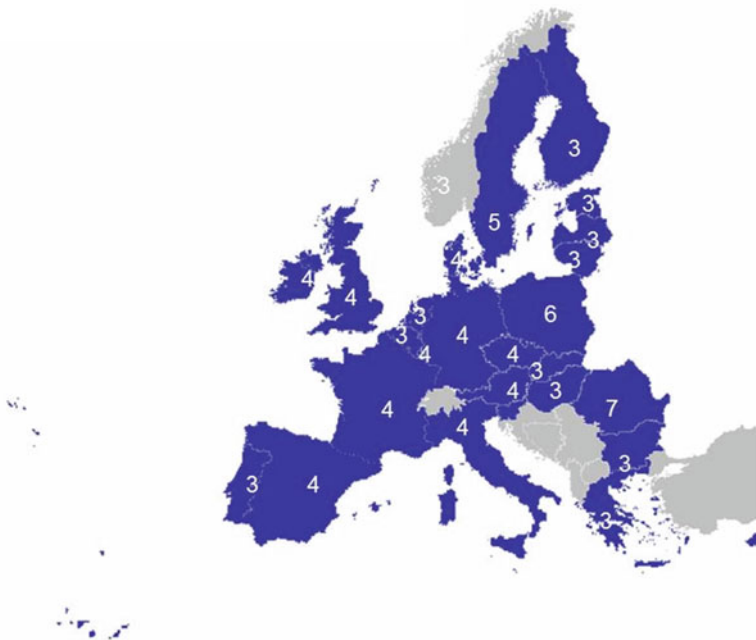


Fig. 4.6 Number of telecommunications-service providers per country (by aixvox gmbH)

It is not, however, the best way to offer a solution to mobile users because roaming costs for Internet access deter mobile users from using an SR or an MT solution that is not fully embedded in the device. Added to this is the problem of network coverage. The most wonderful mobile cloud service is useless if there is no connection available or if the connection is too expensive.

One possible solution for MT is a product called JIBBIGO. JIBBIGO was created several years ago by scientists from Germany and the USA. It offers speech-to-speech translation for 12 languages, including English, German, French, Spanish, Italian, Arabic, Mandarin Chinese, Japanese, Korean, Thai, and Tagalog. JIBBIGO is still in an early stage of development. Currently, it can only translate from those other languages into English (Fig. 4.7).

The goal of JIBBIGO is to provide seamless MT. For example, a traveler from France who speaks no German would be able to tell the device, “*Je voudrais commander un diner délicieux et quelque chose frais à boire.*” The SR and MT on the mobile device would produce a spoken or a text output of the correct translation, “*Ich möchte gern ein leckeres Abendessen und etwas kühles zu trinken*” and, hopefully, the waiter would bring something tasty for dinner along with a cold drink. JIBBIGO would also support synthesized, spoken output of the translation (Fig. 4.8).

Fig. 4.7 JIBBIGO (provided by JIBBIGO)



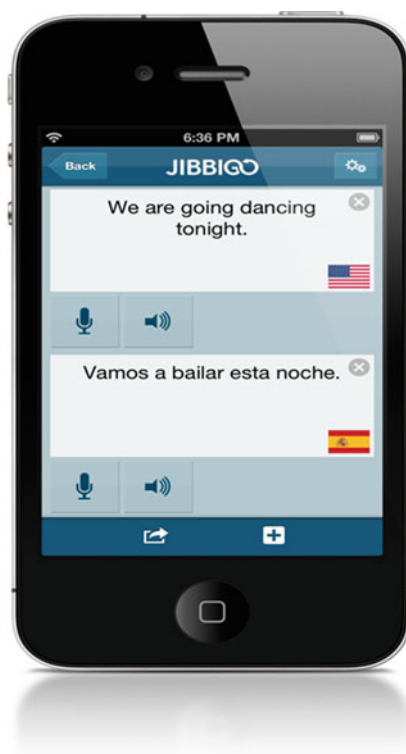
The true value of devices like JIBBIGO is that they don't need an Internet connection or a server to operate. For Europe, this also means that there would be no roaming costs for MT.

The independence of this application from the Internet or a server is not only useful for tourists and traveling businesspeople, but it may also be applied to humanitarian operations. Many of those operations take place in areas where there is no Internet connection, inside or outside of the EU.

Conclusion

This chapter has shown that the use of speech technology and MT is on the rise. Better, easier, and more user-friendly speech applications enter the market every day and most of them can be accessed by a cell phone. This means that they are accessible everywhere, which is very important because mobility is becoming universal. Hence, they form a growing part of our daily routine. When used in combination with other technologies, they produce UC which brings the power of

Fig. 4.8 Sample translation
(picture from JIBBIGO)



SR and other individual technologies to another level. Fortunately, this trend is also addressing the two main barriers to SR, MT, and UC in Europe: multilingualism and roaming costs for Internet access.

Part II

Advanced NLP Methods and Applications

Chapter 5

Exploiting Lexical Sensitivity in Performing Word Sense Disambiguation

Oi Yee Kwong

Abstract State-of-the-art research on word sense disambiguation (WSD) has demonstrated the superiority of supervised learning systems and necessity of multiple knowledge sources. However, despite the complex interaction observed between these two external factors, the intrinsic reason underlying such phenomena is not sufficiently understood. This calls for more qualitative analysis of disambiguation results from an interdisciplinary perspective. In this chapter, we explore the long realised lexical sensitivity issue in WSD in terms of concreteness, with reference to the context availability model in psycholinguistics and the Sketch Engine popularly used in lexicography. It will be shown that the “difficulty” of disambiguating a particular target word is a function of its information susceptibility, which depends on how the senses of the word were distinguished in the first place, thus leading to varied effectiveness of individual knowledge sources as observed. WSD could thus be treated as the reverse engineering of lexicography so that the use of knowledge sources, and the feature selection, could then be more informed with respect to individual words and their senses, and the combinations of algorithms and knowledge sources could be applied in a real lexically sensitive way.

Introduction

Word sense disambiguation (WSD) has always been considered an important intermediate task in natural language processing, with its potential contribution typically portrayed for applications like information retrieval and machine translation systems. It is also expected to benefit speech recognition and generation systems. The ability to identify the meanings of words in a discourse will enable better

O.Y. Kwong (✉)

Department of Chinese, Translation and Linguistics, City University of Hong Kong,

Tat Chee Avenue, Kowloon, Hong Kong

e-mail: Olivia.Kwong@cityu.edu.hk

distinction between homographs, such as “tier” is pronounced as /tiə/ and /taɪə/ when it means a level or a person, respectively, and homophones, such as /rait/ should spell as “write” in “to write a letter” but “right” in “to right the wrongs”. Accurate WSD is therefore critical for the intelligible pronunciation of homographs from text to speech, and the correct transcription of homophones from speech to text, in speech processing applications.

Current mainstream research on WSD often treats the problem as a classification task. Systems attempt to assign the most appropriate sense in a given sense inventory to individual words in a text. State-of-the-art WSD systems, as exemplified in recent SENSEVAL and SEMEVAL evaluation exercises (see, e.g., Edmonds and Cotton 2001; Mihalcea et al. 2004), have demonstrated the superiority of supervised approaches, where machine learning algorithms are trained on sense-tagged examples, using a wide range of features extracted from the text approximating a variety of knowledge sources deemed useful for disambiguation. Ensembles of different types of classifiers based on different feature sets with some voting scheme often report better performance than individual classifiers alone, though the advantage may just be marginal. While complex interactions between learning algorithms and knowledge sources have been observed (e.g. Yarowsky and Florian 2002), the nature underlying such interactions, which points to the *lexical sensitivity* issue of WSD, is somehow under-explored. In particular, more qualitative analysis of disambiguation results from an interdisciplinary perspective will be needed for a better understanding of the issue. As Krahmer (2010) pointed out, mutual learning between computational linguists and psychologists is crucial.

The varied difficulty of disambiguating individual target words has been attributed to a number of factors, including the number and granularity of the senses, the size of the training datasets, the part-of-speech (POS) of the target words, etc. In this chapter, we intend to explore the difficulty of WSD, or otherwise the differential effectiveness of various disambiguating knowledge sources, with respect to the more intrinsic properties of the senses, including the concreteness of the concepts concerned as well as the rationales underlying the sense distinctions of individual words made by lexicographers in the first place. To this end, we refer to the context availability model in psycholinguistics (Schwanenflugel 1991), which is used to explain human comprehension processes in general and more specifically to account for the concreteness effect in human word processing, to analyse system performance on individual target words. We also make use of the Sketch Engine (Kilgariff et al. 2004), a widely used lexicographic tool, to assist in uncovering the readiness of individual senses to be associated with some characteristic linguistic context. We suggest that WSD difficulty be considered a function of the *information susceptibility* of individual target words, which depends on how the senses were distinguished in the first place. The characteristic contexts underlying the sense distinctions and the distinctness of such contexts are thus responsible for the usefulness of different knowledge sources in subsequent disambiguation. Thus the lexical sensitivity issue could be more positively addressed by treating WSD as the reverse engineering of lexicography. The use of knowledge sources, and thus the feature selection, could then be more informed with respect to individual words and their

senses, and the weightings for different combinations of algorithms and knowledge sources could be adjusted in a real lexically sensitive manner.

In Section “WSD: State of the Art”, we first briefly review state-of-the-art WSD approaches, with particular focus on the complex interaction between learning algorithms and knowledge sources revealed in recent evaluation exercises and various comparative studies. In Section “Semantic Characteristics and Lexical Processing”, we present the context availability model and discuss how it accounts for the concreteness effect in psycholinguistics. We apply this model to account for the lexical sensitivity of WSD, and hypothesise that the availability of distinct and characteristic *linguistic* context for individual senses of a target word is a critical factor for the success of its disambiguation and is responsible for the differential effectiveness observed for different knowledge sources. Section “Qualitative Analysis of System Performance” reports on our qualitative analysis of the results from a simple WSD experiment on the noun samples in the SENSEVAL-3 English lexical sample task, with an introduction to our use of the Sketch Engine, a corpus query system popularly used in lexicography, as a tool for comparing the linguistic context availability among word senses. The chapter is concluded with future directions in Section “Conclusion and Future Directions”.

WSD: State of the Art

In this section, we mainly focus on the evaluation of state-of-the-art WSD systems and comparative studies exploring the differential effectiveness of individual algorithms and knowledge sources across different experimental settings. Readers are suggested to refer to Ide and Veronis (1998), Agirre and Edmonds (2006), Navigli (2009), and Yarowsky (2010) for the history of WSD and detailed surveys of WSD methods.

Recent evaluation exercises like SENSEVAL and SEMEVAL have enabled WSD systems based on different approaches to compare with one another with respect to a common dataset and sense inventory. In these exercises, the superiority of supervised systems is particularly obvious. For example, the best performing system in SENSEVAL-1 was a supervised system based on hierarchical decision lists, using a rich set of collocational, morphological and syntactic contextual features with different weightings, which achieved an accuracy of 78.1% on the test data (Kilgariff and Rosenzweig 1999; Yarowsky 2000). Other studies have demonstrated the use of different learning algorithms for the task, such as Naïve Bayes, instance-based learning, Support Vector Machines and inductive logic programming, amongst others (e.g. Leacock et al. 1998; Lee et al. 2004; Stevenson 2003; Specia et al. 2010). According to Mihalcea et al. (2004), several of the top performing systems in SENSEVAL-3 were based on combinations of multiple classifiers with some voting scheme.

Another obvious characteristic of state-of-the-art WSD systems is their reliance on multiple knowledge sources for disambiguation, almost without exception.

The need for multiple knowledge sources has already been realised in early AI-based and knowledge-based WSD systems (e.g. Small and Rieger 1982; McRoy 1992). Stevenson (2003) combined many weak knowledge sources for disambiguation. It is thus well observed that sense distinctions could be based on different linguistic properties, and multiple knowledge sources like POS, topical relation, selectional restriction, domain information, and local collocation are necessary for disambiguation. None of them is sufficient for the task, but each of them may contribute in one way or another.

Learning Algorithms and Knowledge Sources

Two critical factors for the success of supervised WSD systems are thus the learning algorithms adopted and the knowledge sources used.

Individual learning algorithms varied in their disambiguation performance. For instance, Màrquez et al. (2006) compared five machine learning algorithms widely used in previous studies, namely, Naïve Bayes (NB), k-Nearest-Neighbor (kNN), Decision Lists (DL), AdaBoost (AB), and Support Vector Machines (SVM). They were trained on the same set of data and tested on examples selected from the DSO corpus. Knowledge sources were in the form of 15 local feature patterns (with words and POS) and topical context as bag of words (content words in the sentence). The most-frequent-sense classifier was used as the baseline. It was found that all algorithms outperformed the baseline (46.55%), with SVM (67.07%) and AB performing significantly better than kNN, which in turn performed significantly better than NB and DL (61.34%).

Multiple knowledge sources are indispensable in WSD systems. Agirre and Stevenson (2006) summarised from many WSD studies the different knowledge sources available or extracted from various lexical resources and corpora, and their realisation as different features in individual systems. They generalised that all knowledge sources seem to provide useful disambiguation clues. Each POS profits from different knowledge sources, e.g. domain knowledge and topical word association are most useful for disambiguating nouns while local context benefits verbs and adjectives. The combination of all knowledge sources consistently gets the best results across POS categories. In addition, some learning algorithms are better suited to certain knowledge sources, and different grammatical categories may benefit from different learning algorithms.

Such a complex interaction between learning algorithms and knowledge sources was exemplified in the comprehensive study by Yarowsky and Florian (2002). They compared the relative system performance across different training and data conditions with SENSEVAL-2 data on four languages. The results clearly show the interaction among feature sets, training sizes, and learning algorithms. They concluded that “there is no one-size-fits-all algorithm that excels at each of the diverse challenges in sense disambiguation”. For example, discriminative and aggregative algorithm classes often have complementary regions of effectiveness

across numerous parameters; the former such as decision trees tend to perform well with local collocations or syntactic features, whereas the latter like Naïve Bayes tend to perform well with bag-of-word features. Some algorithms are more tolerant than others of sparse data, high degree of polysemy and noise in the training data.

Lexical Sensitivity

Despite such findings on the complex relationship between learning algorithms and knowledge sources, which possibly lead to the use of ensembles of classifiers with diverse knowledge sources in state-of-the-art systems, there are nevertheless some questions regarding their differential effectiveness left unanswered. In particular:

- While we find that target words of different POS categories favour different knowledge sources for disambiguation, how can we also account for the intra-POS variation in disambiguation effectiveness of the same and most-favoured knowledge source?
- Although senses are often considered an artefact produced by lexicographers and their granularity might affect the difficulty of disambiguation, if their distinction was somehow based on sufficient contextual and usage differences in the first place, why would systems fail to distinguish them?

Attributing the varied system performance to the different models produced by individual algorithms, training data size, representation of the knowledge sources as features, POS categories or even the number, granularity and dispersion of senses still may not give a satisfactory account for the notorious issue of *lexical sensitivity*. As Resnik and Yarowsky (1999) remarked, WSD is a highly lexically sensitive task which in effect requires specialised disambiguators for each polysemous word. However, in what way precisely is the combination of algorithms and knowledge sources sensitive to individual (groups of) lexical items? Factors like the number of senses and how closely they are related will have an impact on the difficulty of disambiguation, and the varied difficulty may be reflected from the system performance (Chugur et al. 2002; Pedersen 2002), but there is still more to learn. To shed light on the above questions, more qualitative analysis of disambiguation results from an interdisciplinary perspective is necessary, to investigate the susceptibility of individual words to different knowledge sources and if there are any intrinsic properties shared by the lexical items which can be effectively disambiguated with a particular kind of knowledge source.

Semantic Characteristics and Lexical Processing

Where lexical sensitivity of WSD is concerned, previous studies have made important observations such as verbs are in general more difficult to disambiguate than nouns and adjectives (e.g. Agirre and Stevenson 2006) and topical contexts

are more useful for disambiguating nouns while syntactic information and local collocations benefit verbs and adjectives more (e.g. Leacock et al. 1998), not to mention the obvious impact of sense number and granularity, or frequency. However, the conclusion of disambiguation performance with respect to POS alone may be overgeneralised and does not satisfactorily account for the lexical sensitivity issue. Even when a particular knowledge source is found to be beneficial to target words of a particular POS, it is often only a general remark and does not necessarily mean that the observed improvement is found for all target words in that POS across the board. If POS fails to neatly group the words with respect to their response to particular disambiguating information, we will have to look for other intrinsic properties of words to explain the phenomenon.

The Concreteness Effect

In his discussion on the semantic characteristics of isolated words which may influence human lexical processing, Taft (1991) listed concreteness as the first characteristic, followed by polysemy and subjective familiarity. Many psychological studies have also supported that concrete words are easier to learn and understand than abstract words (e.g. Paivio et al. 1968; Kroll and Merves 1986; Schwanenflugel 1991). Studies on lexical access have shown that concrete words and abstract words might be separately stored in the mental lexicon, and the reaction time is often shorter for the former in lexical decision tasks and naming tasks (e.g. Bleasdale 1987; Kroll and Merves 1986). Studies on children's spoken and reading vocabulary have shown that abstract words are acquired later than concrete words during childhood (e.g. Yore and Ollila 1985).

While polysemy (in terms of sense number and granularity) and familiarity (in terms of frequency or prior probability) have also been addressed by computational linguists to account for differential system performance, the concreteness effect is somehow seldom discussed in the WSD literature. Nevertheless, Jorgensen (1990), for a word usage sorting task, suggested that concreteness of a word may increase agreement between judges and concrete words are easier to define. Kwong (2008) attempted to sort out the relation between concreteness and system performance in SENSEVAL-2, though the findings were not particularly conclusive, partly because of the confusion from discussing concreteness at both the sense and word level. Yuret and Yatbaz (2010) have mentioned that the abstract classes were responsible for most of the errors in their supersense tagging with unsupervised method. More in-depth analysis of the concreteness effect is therefore needed, especially with respect to mainstream supervised WSD.

Context Availability Model

Psychologists have put forth various plausible explanations to account for the concreteness effect observed in human lexical processing. For example, the dual-coding

model suggests that concepts are represented simultaneously in two systems, a verbal system with mostly linguistic properties and a nonverbal system containing images, and abstract concepts are primarily represented in the verbal system (Paivio 1986). According to the model, abstract concepts are more difficult as they are not as readily represented in the form of an image. This model makes the consistent prediction that concrete concepts will be better comprehended because they are more likely to have both the verbal and the image representations available.

The context availability model, on the other hand, suggests that the advantage of concrete words comes from their stronger and denser association to contextual knowledge than abstract words (Schwanenflugel 1991). The availability of contextual information enables a person to draw the relations between concepts that are needed for comprehension. Such contextual information may come from a person's prior knowledge or from the stimulus environment. According to this model, lexical decisions tend to take longer for abstract words because related contextual information that is used in deciding that an item is a word is less available for abstract words. Schwanenflugel et al. (1988) thus pointed out that unlike what the dual-coding model will predict, the lexical decision times for abstract words are not necessarily longer than those for concrete words, especially when abstract concepts are also presented in relevant contexts. They performed a series of experiments, controlling the context availability of the stimulus words, both concrete and abstract, and found that under such a condition the lexical decision times for both word types did not significantly differ. In addition, they compared rated context availability to other predictors of lexical decision time such as imageability, familiarity and age-of-acquisition to determine the relationship between context availability and these other variables. They concluded that concreteness effects in lexical decisions are not entirely due to the difficulty to retrieve an image for abstract concepts, but are rather attributable to the ease of retrieving related contextual information from prior knowledge for individual words, that is, context availability matters.

Context Availability and Information Susceptibility

Schwanenflugel (1991) also remarked that the context availability model has an advantage over other views because it is in basic agreement with the discussions about the nature of concepts and word meanings in general, such as Barsalou (1982), Roth and Shoben (1983), Greenspan (1986), etc., who have focused on the episodic, contextually based character of word meaning. There is a general movement away from thinking of concepts as having fixed, definitional cores that become activated on all instances of word use and they suggested that not all knowledge associated with concepts is equally accessible during every instance of word use. Such emphasis on the contextually based character of word meanings is obviously in line with current mainstream practice in WSD. The following comment particularly highlights the relevance and potential applicability of the model in our investigation of lexical sensitivity in WSD:

... Some knowledge is said to be context-dependent (accessible only in particular contexts) and other knowledge is said to be context-independent (accessible in all contexts of word use). It is possible that words rated low in context availability largely possess context-dependent knowledge which is relatively inaccessible when the words are presented in isolation. However, when such words are presented in supportive contexts, this context-dependent information becomes highly available for deriving meaning, eliminating potential differences in comprehension between abstract and concrete words. (Schwanenflugel, 1991: p.246)

Hence we try to apply the context availability model to our study of lexical sensitivity in WSD, by investigating the relationship between the effectiveness of various knowledge sources (in terms of the disambiguation performance) and the availability of *characteristic linguistic context* distinguishing one sense from the others for a particular target word. To do this, however, we will have to introduce a variation to the model. In an earlier study (Kwong, 2008), we have observed that concreteness seems to be related to the difficulty of WSD, although the precise effect is still to be accounted for. For one set of data, concrete words are apparently more easily disambiguated; but for another, highly concrete words and highly abstract words are easier for WSD than those in the middle of the concreteness continuum. This inconclusiveness is partly attributable to the confusion between lexical concreteness and sense concreteness, which is also a major inadequacy in psycholinguistic studies of the concreteness effect. On the one hand, the existence of polysemy means that a word can have multiple senses, but when psycholinguists attempt to norm the concreteness ratings from human subjects, there has been no control on how they actually come up with a rating for the word as a whole. For instance, the more familiar or common senses of the word might have a greater influence on the rating. On the other hand, especially in view of the phenomena of sense extensions and metaphorical usages, polysemous words may consist of a mix of both concrete and abstract meanings, and it would make better sense to discuss the concreteness effect at the sense level instead of, or at least in addition to, the word level. This is particularly critical when word sense disambiguation is concerned.

Hence we start with sense concreteness, and investigate the difficulty as well as the effectiveness of various knowledge sources in disambiguation. We attempt to explain the lexical sensitivity phenomenon in WSD with respect to context availability, and to this end we apply the context availability model at the sense level, and focus on the availability of *linguistic context*. We hypothesise that the differential effectiveness of individual knowledge sources is a result of the varied availability of *characteristic linguistic context* which serves to distinguish one sense from the others for a particular target word in the first place. This difference thus leads to different information susceptibility of individual target words, which is in turn reflected in the disambiguation performance, indirectly as the difficulty of WSD, giving rise to the long-standing issue of lexical sensitivity. In the current study, we therefore pursue qualitative analysis of WSD results with respect to the concreteness of words and senses, and explore the potential use of context availability to predict WSD difficulty and the effectiveness of individual knowledge

sources. It is hoped that by proposing a semantic characteristic for viewing the differential system performance, a better understanding of the lexical sensitivity issue in WSD can be achieved.

Qualitative Analysis of System Performance

To obtain a better and clearer map of the effectiveness of various knowledge sources deemed important for WSD, with respect to concreteness as a potential underlying factor, more qualitative analysis is needed. We first set up a simple WSD experiment, running different supervised learning algorithms (including Naïve Bayes, Decision Tree, and SVM) with various knowledge sources (including topical contexts, local collocations, and local syntactic contexts) and their combinations on the noun samples available from the SENSEVAL-3 English lexical sample task. The most frequent sense was used as the baseline. The disambiguation accuracy was analysed and compared across individual target words. For the experiments, we used the learning algorithms implemented in the WEKA package (Hall et al. 2009). All default settings were used. The data and the knowledge sources tested are spelled out in the following subsections. For tokenisation and tagging of the data, the tokeniser and tagger available with the Lund University dependency parser (Johansson and Nugues 2008) were used, although we did not use the parser specifically for this study.

Dataset

The data available for target nouns tested in the SENSEVAL-3 English lexical sample task were used. According to Mihalcea et al. (2004), the examples were extracted from the British National Corpus and the sense annotation was done using the Open Mind Word Expert system (Chklovski and Mihalcea 2002). The sense inventory used for the nouns is WordNet 1.7.1 (Miller 1995). Table 5.1 shows the target nouns with the number of senses and the distribution of concrete and abstract senses, as well as the number of training and testing instances for each noun. There are 20 items, with 3–9 senses, averaging at 5.35 senses.¹ The number of training examples for each sense varies considerably. The concrete/abstract classification of the senses was based on the lexicographer files in WordNet. Senses are organised under 45 lexicographer files based on syntactic category and logical groupings, and 26 of them are relevant to noun senses. We considered 7 of them concrete classes and the remaining 19 abstract classes. The concrete classes thus include *animal*,

¹These only cover the senses with training examples, not all senses listed in the sense inventory, hence the slight difference from the figures stated in Mihalcea et al. (2004).

Table 5.1 Distribution of senses and tagged instances for the target nouns

| Target word | Senses | Concrete | Abstract | Training | Testing |
|--------------|--------|----------|----------|----------|---------|
| Argument | 5 | 0 | 5 | 221 | 111 |
| Arm | 5 | 4 | 1 | 266 | 133 |
| Atmosphere | 5 | 1 | 4 | 161 | 81 |
| Audience | 4 | 0 | 4 | 200 | 100 |
| Bank | 9 | 4 | 5 | 262 | 132 |
| Degree | 7 | 0 | 7 | 256 | 128 |
| Difference | 5 | 0 | 5 | 226 | 114 |
| Difficulty | 4 | 0 | 4 | 46 | 23 |
| Disc | 4 | 3 | 1 | 200 | 100 |
| Image | 6 | 3 | 3 | 146 | 74 |
| Interest | 7 | 0 | 7 | 185 | 93 |
| Judgment | 7 | 0 | 7 | 62 | 32 |
| Organization | 4 | 0 | 4 | 112 | 56 |
| Paper | 7 | 1 | 6 | 232 | 117 |
| Party | 5 | 1 | 4 | 230 | 116 |
| Performance | 5 | 0 | 5 | 172 | 87 |
| Plan | 3 | 1 | 2 | 166 | 84 |
| Shelter | 4 | 2 | 2 | 196 | 98 |
| Sort | 4 | 1 | 3 | 190 | 96 |
| Source | 7 | 3 | 4 | 64 | 32 |

artefact, body, food, object, person and *plant*. The abstract classes are *act, attribute, cognition, communication, event, feeling, group, location, motive, phenomenon, possession, process, quantity, relation, shape, state, substance, time* and *Tops* (the unique beginner for nouns).

Knowledge Sources

In this study, we focus on three types of disambiguating information: topical contexts, local collocations and shallow syntactic information. They are realised in the form of bag of words, single words and word combinations in surrounding context and the POS of neighbouring words, respectively, as binary features for the learning algorithms.

Topical Contexts

Topical contexts capture the broad conceptually related words, which are expected to reflect the topic or the domain in which a sense often occurs. Upon testing a few different configurations, for this study we collected from the training examples

all the noun and verb lemmas within a window of ± 50 words from the target as features. Then in each testing instance, if any of those lemmas are found in a window of ± 50 words from the target, the corresponding feature will have value 1, otherwise 0. For example, in the following testing instance for the word “disc”, the lemmas for the underlined words match with the features collected from the training data, and their values will be 1.

Video-recording can be used, as with films, to create an atmosphere of reality and convey moving images, but these are requirements seldom met with in library instruction. Short video-recorded programmes might well be used to illustrate lectures, as a means of creating interest. Video-recording can make use of tape, film or discs for the actual storage of the recorded material. One of the problems facing libraries in the use of video material has been the lack of standardization between different systems. It would appear that cassette systems are easier to use, as audio-cassettes, in library education.

Local Collocations

The collocation patterns were approximated by the lemma unigrams and bigrams in the local context of the target word, within a window of ± 3 words. Although we considered only up to bigrams on each side of the target word, 3-word windows were used. This is to allow some flexibility in case additional modifiers like adverbs and adjectives are present. From the training instances, unigrams w_{-3} , w_{-2} , w_{-1} , w_1 , w_2 and w_3 and bigrams $w_{-3}w_{-1}$, $w_{-2}w_{-1}$, $w_{-3}w_{-1}$, w_1w_2 , w_2w_3 and w_1w_3 were extracted as features. These features were matched within a 3-word window on the left and the right of the target word correspondingly in a training instance. The word form of the target word was also included. For the example above, the word form of the target is < discs >, and < film >, < or > and < film_or > are matched from the left context, while < for >, < the > and < for.the > are matched from the right context.

Shallow Syntactic Information

For this knowledge source, we collected features from the POS of the neighbouring words and the target word itself in the training instances, namely, p_{-3} , p_{-2} , p_{-1} , p_0 , p_1 , p_2 and p_3 .

For the example above, the POS corresponding to these positions are “,” (comma), NN, CC, NNS, IN, DT and JJ, respectively.

They were rendered as binary features.²

²We had also tested the use of POS unigrams and bigrams like what we did for the local collocations, but it turned out that the POS unigrams at fixed positions seemed to produce better results with the current dataset. So in this study, we did not use POS bigrams.

Procedures

WSD results were first obtained with individual classifiers using various combinations of the knowledge sources. The results were then subject to comparison and error analysis, with respect to the two questions we raised in Section “Lexical Sensitivity” regarding the lexical sensitivity issue, namely:

- How to account for the intra-POS variation for the effectiveness of different knowledge sources
- What might be the discrepancy with how the senses were distinguished by lexicographers in the first place

As mentioned, we will cut in from the perspective of context availability, in an attempt to account for the disambiguation results on concrete and abstract senses. To more systematically discuss context availability, we consider the sense distinctions from the lexicographers’ perspective.

Lexicographers distinguish senses by many criteria, including the most important considerations as follows:

- Syntactic patterns
- Collocation patterns
- Colligation patterns
- Domain

If one considers senses the artefacts from lexicography (e.g. Kilgarriff 2006), it makes sense to think about WSD from lexicographers’ point of view, because whether they rely on sufficient characteristic contextual difference to distinguish the senses to start with will directly affect the difficulty of subsequent word sense disambiguation and the usefulness of various knowledge sources for this purpose. Hence we try to assess context availability with the Sketch Engine, an important tool for computational lexicography.

The Sketch Engine

The Sketch Engine is a corpus query system which takes as input a corpus of any language and a corresponding set of grammar patterns, and generates word sketches for the words of that language, whereas word sketches are one-page automatic, corpus-based summaries of a word’s grammatical and collocational behaviour (Kilgarriff et al. 2004). The Sketch Engine has played an important role in modern computational lexicography, with its word sketch function most notably used by the Macmillan English Dictionary (Rundell 2002). Sketch difference is also one of the many functions available in the Sketch Engine. It provides useful summaries in how pairs of near-synonyms differ, allowing users to compare and contrast the grammatical and collocational patterns of two words with apparently similar meanings.

Table 5.2 Examples of word sketch relations for nouns

| Gramrel relation | Explanation |
|------------------|--|
| and/or | Indicates locally collocated nouns conjoined with and/or |
| object_of | Indicates verbs which usually take the noun as object |
| subject_of | Indicates verbs for which the noun is subject |
| a_modifier | Adjectival pre-modifier for the noun |
| n_modifier | Nominal pre-modifier for the noun |
| modifies | Head noun being modified by the noun |
| pp_%s | Prepositional phrase following the noun, the preposition itself and the noun within the prepositional phrase |

We capitalise on the sketch difference function for comparing and contrasting individual senses of a word, to identify important grammatical and collocational patterns within specific grammatical relations critical for their distinction, and hence map out the usefulness of various knowledge sources for their disambiguation, and explore any relation with the concreteness effect.

Creating Sense Sub-corpora in Sketch Engine

The training data from the SENSEVAL-3 English lexical sample task were first preprocessed. All examples were extracted and stored in different files according to individual senses. A corpus was created in Sketch Engine, treating each set of examples as a sub-corpus, and all other senses of the same word as another sub-corpus, to facilitate subsequent comparison of prominent contexts among senses. For each target noun, we obtained the sketch difference for each of its senses with the rest of its senses, and analysed for common patterns and unique patterns with respect to sense concreteness and difficulty of WSD. For the word sketch patterns, we used the default English Penn Treebank sketch grammar available from the Sketch Engine, and added three more patterns. Some typical grammatical relations (gramrel) specified in the word sketch patterns relevant to nouns are listed in Table 5.2. The sketch grammars were written using regular expressions, indicating the POS patterns surrounding the target word, and the actual lexical item involved in these syntactic and collocation patterns. For example, the pattern for extracting collocated nouns for the n_modifier relation is as follows:

```
2 : "NN . ? . ? " "NN . ? . ? " { 0 , 3 } 1 : "NN . ? . ? "
```

Figure 5.1 shows an example of the sketch differences between the second sense of the target noun “disc” (phonograph record) and its other senses (circular plate/magnetic disc/saucer) displayed by the Sketch Engine.



Fig. 5.1 Screen capture of sketch differences between Sense 2 and other senses of disc

Results and Analysis

As seen from Table 5.1, the target words have considerably different number of training and testing instances. Moreover, most of them are abstract. Of the 20 items, 9 only have abstract senses, and the rest have a mix of concrete and abstract senses. None is entirely concrete. Among the 107 senses for all words, only 24 are concrete senses. So the data is in some way biased in their concreteness. Although running WSD experiments on SENSEVAL data would allow better comparison with previous studies, ideally there should be better control over the concreteness distribution especially for the purpose of this investigation. For this study, we will just note this deficiency.

Table 5.3 Accuracy for individual algorithms with various knowledge sources

| Knowledge\Algorithm | NB | DT | SVM |
|---------------------|-------|-------|-------|
| TC | 0.583 | 0.601 | 0.664 |
| LC | 0.649 | 0.599 | 0.623 |
| SS | 0.582 | 0.547 | 0.565 |
| TC + LC | 0.630 | 0.590 | 0.657 |
| TC + SS | 0.587 | 0.616 | 0.683 |
| LC + SS | 0.661 | 0.607 | 0.652 |
| TC + LC + SS | 0.636 | 0.609 | 0.672 |

Knowledge Sources and Learning Algorithms

Table 5.3 shows the results obtained from various learning algorithms (NB for Naïve Bayes, DT for Decision Tree and SVM for Support Vector Machines) with different combinations of features modelling various types of knowledge sources (TC for Topical Contexts, LC for Local Collocations and SS for Shallow Syntactic Information). The figures refer to precision, which is the same as recall in this case since coverage is 100% for all target words. The baseline from the most frequent sense is 0.542.

All the results in Table 5.3 are above the baseline. However, contrary to what most previous studies might have observed, combining all knowledge sources does not necessarily give the best result. For instance, adding LC to TC and SS with SVM leads to a drop in overall performance. Such slight discrepancy among different studies may also depend on the specific dataset being used and the minor tunings done to the classifiers individually (we only used default settings). Many studies (e.g. Lee and Ng 2002) have reported the superiority of SVM over other learning algorithms. This is apparently the case especially when many different knowledge sources are combined. For individual knowledge sources, other algorithms may sometimes have an advantage. For example, NB gives better overall results when only LC and SS are used, whereas when TC is used, SVM performs better. This might be related to the dimensionality of the feature space, and SVM apparently copes better with high dimensionality than NB. Since DT looks for the best distinguishing features, it needs a set of most relevant and characteristic features, and high dimensionality may not be favourable.

Nevertheless, just looking at the overall results could sometimes be misleading. Table 5.4 shows the best and worst scores for individual target words, and Table 5.5 shows the algorithm and knowledge source combinations which produced the best scores for each word. The asterisks in the “Abs” column in Table 5.5 indicate that a target word contains only abstract senses.

It is obvious that the accuracy varies across different target words. For instance, the best score ranges from 0.478 for “difficulty” to 0.917 for “plan”. The number of training instances available for individual senses may be partially responsible, since the former has 4 abstract senses but only 46 training instances, while the latter has 1 concrete and 2 abstract senses with 166 training instances altogether. In some cases, such as “arm”, “organization” and “plan”, the most frequent sense is already quite

Table 5.4 Best and worst scores for individual target words

| Target word | Baseline | Best score | Worst score |
|--------------|----------|------------|-------------|
| Argument | 0.514 | 0.550 | 0.468 |
| Arm | 0.820 | 0.902 | 0.842 |
| Atmosphere | 0.667 | 0.716 | 0.667 |
| Audience | 0.670 | 0.800 | 0.670 |
| Bank | 0.674 | 0.833 | 0.788 |
| Degree | 0.609 | 0.758 | 0.680 |
| Difference | 0.404 | 0.605 | 0.377 |
| Difficulty | 0.174 | 0.478 | 0.261 |
| Disc | 0.380 | 0.780 | 0.620 |
| Image | 0.365 | 0.649 | 0.581 |
| Interest | 0.419 | 0.753 | 0.452 |
| Judgment | 0.281 | 0.594 | 0.469 |
| Organization | 0.732 | 0.768 | 0.732 |
| Paper | 0.256 | 0.564 | 0.350 |
| Party | 0.621 | 0.776 | 0.629 |
| Performance | 0.264 | 0.494 | 0.356 |
| Plan | 0.821 | 0.917 | 0.869 |
| Shelter | 0.449 | 0.622 | 0.480 |
| Sort | 0.656 | 0.677 | 0.646 |
| Source | 0.656 | 0.688 | 0.469 |

Table 5.5 Systems leading to the best scores for individual target words

| Target word | TC | LC | SS | TC + LC | TC + SS | LC + SS | TC + LC + SS | Abs |
|--------------|-----|----|----|---------|---------|---------|--------------|-----|
| Argument | | | NB | | | SVM | | * |
| Arm | | | | | | | SVM | |
| Atmosphere | DT | DT | | SVM | | DT | | |
| Audience | | | | | SVM | | | * |
| Bank | SVM | | | | | | | |
| Degree | | | | | | | SVM | * |
| Difference | | | | | | NB | | * |
| Difficulty | | | | | | SVM | | * |
| Disc | SVM | NB | | | | | | |
| Image | DT | | | | | | | |
| Interest | | | | | SVM | | | * |
| Judgment | DT | | | | | | | * |
| Organization | SVM | DT | | SVM | SVM | DT | | * |
| Paper | | | | | SVM | | | |
| Party | | | | SVM | | | | |
| Performance | | | | | | | NB | * |
| Plan | | | | DT | | | DT | |
| Shelter | | | | | | DT | | |
| Sort | | DT | | | | | | |
| Source | SVM | NB | | | | NB | | |

Table 5.6 Disambiguation performance w.r.t. concreteness

| Concreteness | Baseline | SVM (TC + LC + SS) | Best scores |
|----------------------------|----------|--------------------|-------------|
| Only abstract senses | 0.489 | 0.608 | 0.663 |
| Both abstract and concrete | 0.579 | 0.718 | 0.748 |
| Overall | 0.542 | 0.672 | 0.713 |

hard to beat. In fact, it has been considered a very powerful heuristic in itself (e.g. McCarthy et al. 2007). The best scores for words like “difficulty”, “disc”, “interest”, “judgment” and “paper” have gained most improvement (over 0.3 increase) from the baseline.

With respect to the learning algorithms, although SVM tends to have an overall advantage over the others, when it comes to the best performance on individual target words, DT and NB can serve equally well if used with the appropriate features. For example, using NB could give the best score for “disc” and “difference” with LC and LC + SS, respectively, and using DT with LC + SS could give the best score for “shelter”. Where there is high dimensionality, especially when TC is used, SVM often consistently gives better results.

Concreteness Effect

Table 5.6 shows the overall results from SVM with all knowledge sources and the best score for individual target words, with respect to the concreteness of the senses for the words. Although the SENSEVAL-3 data contain more words with only abstract senses, the results apparently suggest that words with only abstract senses are more difficult to disambiguate than those with a mix of concrete and abstract senses, as is evident from the lower scores for the former in general.

Considering the effectiveness of various knowledge sources on individual target words, as reflected in Table 5.5, words with entirely abstract senses are apparently more susceptible to local syntactic features in addition to topical and collocation features, while those with a mix of abstract and concrete senses are less so. If we only take the minimal feature set which could lead to the best scores for individual words, it is found that 7 out of the 9 words with only abstract senses take topical/collocation features with local syntactic features to achieve the best scores. On the contrary, only 3 out of 11 mixed-sense words need local syntactic features to reach the best scores.

For instance, all senses for “audience” are abstract. The best score for “audience” was achieved by SVM with TC + SS. The senses for the word are as follows, with the text in [] indicating the WordNet lexicographer file label and that in { } indicating the WordNet synset:

Sense 1: [Communication] {consultation, audience, interview}—a conference (usually with someone important)

Table 5.7 Confusion matrices for the disambiguation of “audience”

| (a) SVM with TC | | | | | |
|----------------------|---|----|----|---|--|
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 57 | 3 | 0 | |
| 3 | 0 | 18 | 8 | 0 | |
| 4 | 0 | 3 | 1 | 2 | |
| 2 or 3 | 0 | 7 | 0 | 0 | |
| U | 0 | 1 | 0 | 0 | |
| (b) SVM with SS | | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 49 | 10 | 1 | |
| 3 | 0 | 15 | 11 | 0 | |
| 4 | 0 | 0 | 0 | 6 | |
| 2 or 3 | 0 | 3 | 4 | 0 | |
| U | 0 | 1 | 0 | 0 | |
| (c) SVM with TC + SS | | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 57 | 3 | 0 | |
| 3 | 0 | 15 | 11 | 0 | |
| 4 | 0 | 1 | 0 | 5 | |
| 2 or 3 | 0 | 5 | 2 | 0 | |
| U | 0 | 1 | 0 | 0 | |

Sense 2: [Group] {audience}—a gathering of spectators or listeners at a (usually public) performance

Sense 3: [Group] {audience}—the part of the general public interested in a source of information or entertainment

Sense 4: [State] {hearing, audience}—an opportunity to state your case and be heard

The confusion matrices³ in Table 5.7(a–c) indicate that the knowledge source SS was useful for distinguishing at least some instances of Sense 3 from the easily confusable Sense 2, but not vice versa. Sense 2 is best disambiguated with topical contexts and perhaps it also has an advantage for being the most frequent. Meanwhile, SS is particularly useful for distinguishing Sense 4. The confusability between Sense 2 and Sense 3 is not surprising, as the two senses are so closely related that they both refer to a group of people. Sense 4, on the other hand, is more effectively distinguished with local contexts because most of its usage is in the form of “... the rights of audience ...”. Hence abstract senses are sometimes distinguished in terms of topical context and sometimes better in terms of syntactic patterns. The use of both knowledge sources is needed.

³Some instances in the data were labelled as “U”, with no sense tag given.

Table 5.8 Confusion matrices for the disambiguation of “difficulty”

| | | | | | |
|----------------------|---|---|---|---|---|
| (a) SVM with LC | | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | | 2 | 0 | 2 | 0 |
| 2 | | 1 | 0 | 1 | 0 |
| 3 | | 1 | 0 | 5 | 2 |
| 4 | | 5 | 0 | 0 | 3 |
| U | | 0 | 0 | 0 | 1 |
| (b) SVM with SS | | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | | 3 | 0 | 1 | 0 |
| 2 | | 1 | 0 | 0 | 1 |
| 3 | | 2 | 0 | 4 | 2 |
| 4 | | 3 | 1 | 2 | 2 |
| U | | 0 | 0 | 1 | 0 |
| (c) SVM with LC + SS | | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 | |
| 1 | | 3 | 0 | 1 | 0 |
| 2 | | 0 | 0 | 1 | 1 |
| 3 | | 2 | 0 | 5 | 1 |
| 4 | | 4 | 0 | 1 | 3 |
| U | | 0 | 0 | 0 | 1 |

Meanwhile, the best score for “difficulty” was achieved by SVM with LC + SS. The senses for the word are listed below, all of which are abstract senses:

- Sense 1:** [Act] {trouble, difficulty}—an effort that is inconvenient
- Sense 2:** [Attribute] {difficulty, difficultness}—the quality of being difficult
- Sense 3:** [Communication] {difficulty}—a factor causing trouble in achieving a positive result or tending to produce a negative result
- Sense 4:** [State] {difficulty}—a situation or condition almost beyond one’s ability to deal with and requiring great effort to bear or overcome

It has been noted that one reason for the relatively poor result for this word is the lack of training instances. From the confusion matrices in Table 5.8(a–c), Sense 1 and Sense 3 are apparently better disambiguated than Sense 2 and Sense 4. The difficulty with Sense 2 can partially be attributed to the few training instances it has, while the other three senses have more, and a similar number of, training instances. Sense 4, however, is obviously easy to confuse with the other senses. When local collocations are used, it is often confused with Sense 1. When local syntactic information is used, it has been confused with all other senses. Although the other three senses are more frequent than Sense 4, the errors also suggest that Sense 4 probably does not have much character of its own, to allow it to be effectively distinguished from the rest, at least not so with the knowledge sources under investigation.

Table 5.9 Confusion matrices for the disambiguation of “disc”

| (a) SVM with TC | | | | |
|--------------------|----|----|----|---|
| Expected\Predicted | 1 | 2 | 3 | 4 |
| 1 | 17 | 1 | 1 | 0 |
| 2 | 1 | 37 | 0 | 0 |
| 3 | 0 | 9 | 13 | 0 |
| 4 | 6 | 2 | 0 | 2 |
| 1 or 4 | 6 | 0 | 1 | 0 |
| 1 or U | 1 | 0 | 0 | 0 |
| 4 or U | 0 | 1 | 0 | 0 |
| (b) NB with LC | | | | |
| Expected\Predicted | 1 | 2 | 3 | 4 |
| 1 | 18 | 1 | 0 | 0 |
| 2 | 3 | 34 | 1 | 0 |
| 3 | 0 | 7 | 17 | 0 |
| 4 | 5 | 4 | 0 | 1 |
| 1 or 4 | 7 | 0 | 0 | 0 |
| 1 or U | 1 | 0 | 0 | 0 |
| 4 or U | 0 | 0 | 1 | 0 |

Turning to an example of mixed-sense target word, the best score for “disc” was achieved by SVM with TC and NB with LC simultaneously. The senses for the word are listed below. Sense 1 to Sense 3 are concrete, and Sense 4 is abstract. For this case we will compare the confusability with the two classifiers, respectively, in Table 5.9 (a, b).

- Sense 1:** [Artefact] {disk, disc}—a thin flat circular plate
- Sense 2:** [Artefact] {phonograph record, phonograph recording, record, disk, disc, platter}—sound recording consisting of a disc with continuous grooves; formerly used to reproduce music by rotating while a phonograph needle tracked in the grooves
- Sense 3:** [Artefact] {magnetic disk, magnetic disc, disk, disc}—(computer science) a memory device consisting of a flat disk covered with a magnetic coating on which information is stored
- Sense 4:** [Shape] {disk, disc, saucer}—something with a round shape like a flat circular plate

The impact of availability of training instances can be considered insignificant for the case of “disc”, as all four senses have over 30 instances. The most frequent sense has up to 81 training instances. From Table 5.9, it is obvious that although the two classifiers give the same scores, SVM with TC works better for Sense 2, while NB with LC works better for Sense 1 and Sense 3. It suggests that Sense 2 relies more on topical contexts whereas Sense 1 and Sense 3 are more effectively distinguished with local collocations. Sense 4, however, is not particularly susceptible to either knowledge source, and neither does local syntactic information seem to help, if we consider that adding SS to the classifiers does not improve the score at all.

Context Availability

From the cases illustrated above, we could see that although in general words with only abstract senses tend to be less well disambiguated than those with mixed senses, this concreteness effect may in fact be coupled with how distinct individual senses are from one another in terms of the knowledge sources we use to characterise them.

For the “audience” example, all four senses are abstract, but using both topical contexts and local syntactic information shows improvement on either knowledge source alone, and thus the contribution of both. Ignoring Sense 1 which only had few training examples, the analysis of the sketch difference reveals that Sense 2 has a lot of common patterns with non-Sense 2 examples, alongside the many unique patterns to itself. Similarly, Sense 3 also has a lot of common patterns with non-Sense 3 examples, and some unique patterns for itself. These common patterns and unique patterns are referring to the words collocated with “audience” within the various grammatical relation explicated in Section “Procedures”, but not syntactic patterns. For example, both Sense 2 and Sense 3 of “audience” are found to be followed by similar kinds of prepositional phrases (PP). On the contrary, Sense 4 has almost no common patterns with non-Sense 4 examples. It also has only a few unique patterns, but one of them is very prominent and quite frequent, where “audience” occurs in an of-PP with the lemma “right” preceding the PP. These sketch differences could thus predict the likely confusion between Sense 2 and Sense 3, and the effectiveness of local syntactic patterns to distinguish Sense 4 from the others despite its relatively low frequency.

For the “difficulty” example, although it is handicapped by the lack of training data to start with, the different senses of the word are nevertheless used in relatively uncharacteristic syntactic context. Ignoring Sense 2 which only has few examples, the others do not have many unique local syntactic patterns. Sense 1 and Sense 4 may have some specific adjectival modifiers (e.g. “little”, “great” and “real” for Sense 1, and “personal”, “financial” and “legal” for Sense 4), but this minor uniqueness is not enough to overrule the commonality shared by the senses. For example, all senses tend to be preceded by the determiner “the”, which is obviously not helpful for disambiguation. This lack of distinctive contexts could predict the relative difficulty of disambiguating the word.

For the “disc” example, Sense 1 shares relatively many common patterns with non-Sense 1 data, and so does Sense 4. Sense 2 and Sense 3, on the other hand, share fewer common patterns with others. This possibly predicts the confusability between Sense 1 and Sense 4. In addition, similar to the “audience” case, the unique patterns for individual senses are still restricted to the collocation patterns within particular grammatical relations, instead of any sense enjoying a unique syntactic pattern not found in others. This could explain why features based on words or lemmas are more effective for disambiguating this word, while the addition of local syntactic information does not help at all.

Implications on Lexical Sensitivity

From the above analysis, the following can be observed:

- Nouns with only abstract senses are relatively more difficult to disambiguate than those with a mix of abstract and concrete senses, as seen from the overall scores for the two kinds of words.
- Despite the general superiority of SVM, different combinations of algorithms and knowledge sources were found to work best for individual target words.
- The addition of local syntactic information to topical contexts only marginally improves the overall score, but the actual effect varies across individual target words. Some benefit more from the combined features while others may suffer a drop in the final scores.
- Local syntactic information seems to play a more significant role on the disambiguation of abstract senses than concrete senses.

The fact that different knowledge sources are needed in WSD, as shown in many previous studies, is suggesting that senses are best characterised by different linguistic contextual properties. Past studies have observed that in general adding topical or bag-of-word features is more beneficial for nouns whereas adding local and collocational features works better for verbs and adjectives, but as we have observed in this study, such advantages do not necessarily apply to all words (and their senses) in the whole syntactic category. This means that POS alone may not be adequate to account for the lexical sensitivity of WSD, especially in view of the intra-POS variation with respect to individual knowledge sources. In fact, it may all depend on how distinct the senses are with respect to these knowledge sources, and different words and thus their sense distinctions have different information susceptibility, leading to the differential effectiveness of individual knowledge sources, and hence the lexical sensitivity of the task. The common property shared by instances which can be effectively disambiguated by a certain kind of knowledge source or contextual feature is, simply speaking, context availability and the linguistic properties used by lexicographers for their distinction in the sense inventory in the first place.

The POS effect observed in previous studies could thus be understood this way. There are typical syntactic contexts in which words of different POS are bound to occur. For instance, nouns are often used in the subject and object positions and thus whether we find a verb before or after the target noun or whether its previous word is a determiner may not be a very good contextual feature because the various senses of a given noun may all occur in such similar contexts. On the contrary, if one sense of the noun tends to appear in very specific constructions, such as in very unique prepositional phrases, then in such cases one can expect local collocations and n-gram combinations to be relatively useful for distinguishing this sense from the others. This is perhaps why it has been observed that adding topical features is more beneficial for nouns in general, while adding local collocation and syntactic features helps verb and adjective disambiguation more, since the latter may demonstrate

more variation in syntactic context in terms of subcategorisation patterns, selectional restrictions of the verbs and adjectives, etc. However, some noun senses could still benefit from such local features. Since different nouns may have sense distinctions based on different characteristics, the effectiveness of individual knowledge sources differs across different target words. Thus one problem with previous findings on the relation between knowledge sources and POS is that it may be too crude to look at lexical sensitivity in terms of POS alone and from the overall disambiguation scores. On the one hand, any improvement on the overall results may just be marginal; and on the other hand, the precise effect on individual words could vary considerably. Therefore, to better understand the lexical sensitivity in WSD, one must look into the linguistic features involved in the sense distinction of individual words, which will suggest the differential information susceptibility and thus the usefulness of different knowledge sources in subsequent disambiguation. For example, for the intra-POS variations among nouns, in this study we have observed the concreteness effect. The analysis suggested that concrete senses tend to rely more on topical information or they are more often used in distinctively different domains, while abstract senses are more likely to be characterised by their special local syntactic patterns such as the occurrence in particular PP or being followed by particular PP, in addition to the topic or the domain in which they are often used. The impact of sense concreteness, after all, is coupled with the actual context availability of individual senses, which affects the ease of disambiguation and the effectiveness of various knowledge sources. The model will thus predict that while sense dispersion or granularity will affect the difficulty of disambiguation, but if sufficient characteristic contexts can be associated with the senses and such contexts exist in the data, even closely related senses (such as an originally concrete sense and its abstract and metaphorical extension) could still be effectively disambiguated with the relevant knowledge sources.

If one accepts that senses are the artefacts from lexicography, it makes sense to view WSD as the reverse of the lexicographic process. Feature selection and their combinations or weighting within specific classifiers might then be more appropriately treated as reverse engineering lexicographers' sense distinction. This will enable lexically oriented selection of useful knowledge sources for WSD, which genuinely and positively addresses the lexical sensitivity issue of the task.

Conclusion and Future Directions

In this chapter, we have shown that to face up to the lexical sensitivity issue in word sense disambiguation, one must qualitatively analyse disambiguation results with respect to a certain intrinsic dimension of the target words. While many previous studies have demonstrated the benefits or the disadvantages of using certain knowledge sources for words of particular POS, in the current study we further address the intra-POS variations and discuss lexical sensitivity with respect to sense concreteness. As the context availability model in psycholinguistics predicts,

although concrete words are more easily understood than abstract words, as is evident from the shorter reading time and reaction time observed for concrete words in comprehension and lexical decision experiments, respectively, the concreteness effect will disappear if the stimuli were controlled for the ease to come up with an associative context. Thus it is not entirely a matter of imageability which makes concrete concepts more easily handled, but the readiness to associate a concept with a particular context is a critical factor.

Our analysis of WSD results on the noun samples in the SENSEVAL-3 English lexical sample task has allowed us to make the following observations which may shed further light on the lexical sensitivity issue:

- Words with only abstract senses tend to have lower disambiguation scores than those with a mix of abstract and concrete senses.
- Concrete senses tend to associate better with topical contexts, while abstract senses are more likely to be distinguished with local syntactic patterns in addition to topical contexts and/or collocational patterns.
- The benefit of adding local syntactic information to topical contexts in disambiguation varies across target words, and it depends on the context availability of individual senses and the basis by which lexicographers distinguish and characterise them in the first place.

Hence, in addition to factors like POS, sense granularity, number of senses, availability of training samples, etc., there is something about the intrinsic nature of individual words, such as concreteness, which may affect their susceptibility to different knowledge sources in disambiguation. Some linguistic patterns may be more effective for characterising certain senses, and their presence in the test data would more easily lead to successful disambiguation. It is therefore more appropriate to consider the lexical sensitivity in WSD in terms of information susceptibility, which depends on how the senses of the words were distinguished in the first place and whether their typical contexts are characteristic enough and available in most instances, resulting in the differential effectiveness of individual knowledge sources on different target words. To this end, WSD might be treated as the reverse engineering of lexicography so that the selection of features and their combinations and weighting with specific learning algorithms could be made genuinely sensitive to individual lexical items.

For future work, the following directions will be pursued: First, we have only focused on the noun samples in this study, and it is necessary to extend our testing and analysis to verbs and adjectives as well, to give a fuller picture of lexical sensitivity across different POS. Second, the generalisation of our observations and the portability of systems should be investigated by extending the testing and analysis to other datasets. In particular, since the SENSEVAL-3 noun samples are biased toward abstract senses, we could only make post hoc analysis based on the results. A better approach would be to control the distribution of concrete and abstract senses in the data at the beginning. Third, since many of the bag-of-word features are infrequent, it is worth to match with their synonyms as well, and to use class information if high dimensionality is an issue. More knowledge sources should

also be included in the experiments. Fourth, we could try to leverage dictionaries to provide the critical information on sense distinction, especially in the absence of sufficient sense-tagged training data.

More importantly, so far, as in other previous studies, we have only evaluated WSD and the effectiveness of various knowledge sources from the successful cases, that is, how they contribute positively to disambiguation. Hence if a nice fit between the features collected from the training data and those found in the testing data is present, WSD is relatively straightforward. However, it is obvious that the success rates for some target words are somehow lower than others, and the particular difficulty with such cases could be a result of two scenarios. One is the lack of distinctive and characteristic patterns for some senses. This can be related to the closeness of two senses and low context availability. The other has to do with the conspicuousness of the characteristic patterns. If all senses are distinct and very characteristic with very different linguistic patterns, but such characters are latent or hidden in the test instances, WSD will also become difficult. In either case, one might challenge the lexicographers for an answer, but it remains a problem to be alleviated in automatic WSD. Hence future studies should also address the less characteristic and difficult cases, to boost the disambiguation performance on the relatively more disadvantaged target words.

Acknowledgements The work reported in this chapter was supported by grants from the Department of Chinese, Translation and Linguistics of the City University of Hong Kong.

References

- Agirre E, Edmonds P (2006) Word sense disambiguation: algorithms and applications. Springer, Dordrecht
- Agirre E, Stevenson M (2006) Knowledge sources for WSD. In: Agirre E, Edmonds P (eds) Word sense disambiguation: algorithms and applications. Springer, Dordrecht
- Barsalou LW (1982) Context-independent and context-dependent information in concepts. *Mem Cognit* 11:629–654
- Bleasdale FA (1987) Concreteness dependent associative priming: separate lexical organization for concrete and abstract words. *J Exp Psychol Learn Mem Cogn* 13:582–594
- Chklovski T, Mihalcea R (2002) Building a sense tagged corpus with Open Mind Word Expert. In: Proceedings of the workshop on word sense disambiguation: recent successes and future directions, Philadelphia, pp 116–123
- Chugur I, Gonzalo J, Verdejo F (2002) Polysemy and sense proximity in the Senseval-2 test suite. In: Proceedings of the workshop on word sense disambiguation: recent successes and future directions, Philadelphia, pp 32–39
- Edmonds P, Cotton S (2001) SENSEVAL-2: overview. In: Proceedings of the second international workshop on evaluating word sense disambiguation systems (SENSEVAL-2), Toulouse, France, pp 1–6
- Greenspan SL (1986) Semantic flexibility and referential specificity of concrete nouns. *J Mem Lang* 25:539–557
- Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18

- Ide N, Veronis J (1998) Introduction to the special issue on word sense disambiguation: the state of the art. *Comput Linguist* 24(1):1–40
- Johansson R, Nguens P (2008) Dependency-based syntactic-semantic analysis with PropBank and NomBank. In: *Proceedings of the 12th conference on computational natural language learning (CoNLL 2008)*, Manchester, pp 183–187
- Jorgensen J (1990) The psychological reality of word senses. *J Psycholinguist Res* 19:167–190
- Kilgarriff A (2006) Word senses. In: Agirre E, Edmonds P (eds) *Word sense disambiguation: algorithms and applications*. Springer, Dordrecht
- Kilgarriff A, Rosenzweig J (1999) English SENSEVAL: reports and results. In: *Proceedings of the 5th natural language processing Pacific Rim symposium (NLPRS '99)*, Beijing, China
- Kilgarriff A, Rychly P, Smrz P, Tugwell D (2004) The sketch engine. In: *Proceedings of EURALEX 2004*, Lorient, France, pp 105–116
- Krahmer E (2010) What computational linguists Can learn from psychologists (and vice versa). *Comput Linguist* 36(2):285–294
- Kroll JF, Merves JS (1986) Lexical access for concrete and abstract words. *J Exp Psychol Learn Mem Cogn* 12:92–107
- Kwong OY (2008) A preliminary study on the impact of lexical concreteness on word sense disambiguation. In: *Proceedings of the 22nd Pacific Asia conference on language, information and computation (PACLIC 22)*, Cebu, Philippines, pp 235–244
- Leacock C, Chodorow M, Miller MA (1998) Using corpus statistics and WordNet relations for sense identification. *Comput Linguist* 24(1):147–165
- Lee YK, Ng HT (2002) An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In: *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)*, Philadelphia, pp 41–48
- Lee YK, Ng HT, Chia TK (2004) Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In: *Proceedings of the 3rd international workshop on the evaluation of systems for the semantic analysis of text (SENSEVAL-3)*, Barcelona, Spain, pp 137–140
- Màrquez L, Escudero G, Martínez D, Rigau G (2006) Supervised corpus-based methods for WSD. In: Agirre E, Edmonds P (eds) *Word sense disambiguation: algorithms and applications*. Springer, Dordrecht
- McCarthy D, Koeling R, Weeds J, Carroll J (2007) Unsupervised acquisition of predominant word senses. *Comput Linguist* 33(4):553–590
- McRoy SW (1992) Using multiple knowledge sources for word sense disambiguation. *Comput Linguist* 18(1):1–30
- Mihalcea R, Chklovski T, Kilgarriff A (2004) The SENSEVAL-3 English lexical sample task. In: *Proceedings of the 3rd international workshop on the evaluation of systems for the semantic analysis of text (SENSEVAL-3)*, Barcelona, Spain, pp 25–28
- Miller G (1995) WordNet: a lexical database. *Commun ACM* 38(11):39–41
- Navigli R (2009) Word sense disambiguation: a survey. *ACM Comput Surv* 41(2):1–69
- Paivio A (1986) *Mental representation: a dual coding approach*. Oxford University Press, New York, NY
- Paivio A, Yuille JC, Madigan SA (1968) Concreteness, imagery, and meaningfulness values for 925 nouns. *J Exp Psychol, Monograph Suppl* 76(1,Pt.2):1–25
- Pedersen T (2002) Assessing system agreement and instance difficulty in the lexical sample tasks of SENSEVAL-2. In: *Proceedings of the workshop on word sense disambiguation: recent successes and future directions*, Philadelphia, PA, USA, pp 40–46
- Resnik P, Yarowsky D (1999) Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Nat Lang Eng* 5(2):113–133
- Roth EH, Shoben EJ (1983) The effect of context on the structure of categories. *Cogn Psychol* 15:346–378
- Rundell M (ed) (2002) *Macmillan English dictionary for advanced learners*. Macmillan, Oxford
- Schwanenflugel PJ (1991) Why are abstract concepts hard to understand? In: Schwanenflugel PJ (ed) *The psychology of word meanings*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ

- Schewanenflugel PJ, Harnishfeger KK, Stowe RW (1988) Context availability and lexical decisions for abstract and concrete words. *J Mem Lang* 27:499–520
- Small S, Rieger C (1982) Parsing and comprehending with word experts (A theory and its realization). In: Lehnert WG, Ringle MH (eds) *Strategies for natural language processing*. Lawrence Erlbaum Associates, New Jersey, NJ
- Specia L, Stevenson M, Das Graças Volpe Nunes M (2010) Assessing the contribution of shallow and deep knowledge sources for word sense disambiguation. *Lang Res Eval* 44(4):295–313
- Stevenson M (2003) *Word sense disambiguation: the case for combinations of knowledge sources*. CSLI Publications, Stanford, CA
- Taft M (1991) *Reading and the mental lexicon*. Lawrence Erlbaum Associates, Hove, East Sussex
- Yarowsky D (2000) Hierarchical decision lists for word sense disambiguation. *Comput Humanit* 34(1–2):179–186
- Yarowsky D (2010) Word sense disambiguation. In: Indurkha N, Damerau FJ (eds) *Handbook of natural language processing*. Chapman & Hall, Boca Raton, FL
- Yarowsky D, Florian R (2002) Evaluating sense disambiguation across diverse parameter spaces. *Nat Lang Eng* 8(4):293–310
- Yore LD, Ollila LO (1985) Cognitive development, sex, and abstractness in grade one word recognition. *J Edu Res* 78:242–247
- Yuret D, Yatbaz MA (2010) The noisy channel model for unsupervised word sense disambiguation. *Comput Linguist* 36(1):111–127

Chapter 6

Summarizing Short Texts Through a Discourse-Centered Approach in a Multilingual Context

**Daniel Alexandru Anechitei, Dan Cristea, Ioannidis Dimosthenis,
Eugen Ignat, Diman Karagiozov, Svetla Koeva, Mateusz Kopeć,
and Cristina Vertan**

Abstract The chapter presents the architecture of a system targeting summaries of short texts in six languages. At the core of a summary, which comprises clauses and sentences extracted from the original text, is the structure of the discourse and its relationship with its coreferential links. The approach shows a uniform design for all languages, while language specificity is attributed to the resources that fuel the component modules. The design described here includes a number of feedback loops used to fine-tune the parameters by comparing the output of the modules

D.A. Anechitei (✉) • E. Ignat

Department of Computer Science, “Alexandru Ioan Cuza” University of Iași, Iași, Romania
e-mail: daniel.anechitei@info.uaic.ro; eugen.ignat@info.uaic.ro

D. Cristea

Department of Computer Science, “Alexandru Ioan Cuza” University of Iași, Iași, Romania

Institute for Computer Science, Romanian Academy, Iași Branch, Iași, Romania

e-mail: dcristea@info.uaic.ro

I. Dimosthenis

Atlantis Consulting SA, Thessaloniki, Greece

e-mail: djoannid@atlantisresearch.gr

D. Karagiozov

Tetradcom Interactive Solutions Ltd., Sofia, Bulgaria

e-mail: diman@tetradcom.com

S. Koeva

Institute for Bulgarian Language, Bulgarian Academy of Sciences, Sofia, Bulgaria

e-mail: svetla@dcl.bas.bg

M. Kopeć

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

e-mail: m.kopiec@ipipan.waw.pl

C. Vertan

Department of Linguistics, University of Hamburg, Hamburg, Germany

e-mail: fsha060@uni-hamburg.de

against annotated corpora. “Average” summaries over some human-produced ones are used to evaluate the accuracy of each of the monolingual systems. The study also presents some quantitative data on the corpora used, showing a comparison among languages and results that, mostly, prove to be above the state of the art.

Introduction

The purpose of this chapter is to describe a multilingual summary production line which applies techniques centered on discourse structure. There are a number of features that individuates this approach from other state-of-the-art summarization techniques. The first feature relates to the quality of the summaries produced by the system, which places at its core issues of coherence. That is, a summary is a text by itself and, as such, just like the original, it should preserve the qualities of being cohesive and coherent, even if it is made up of elementary pieces extracted from the original text and reassembled again. The second feature is important from an engineering point of view: we show that a generic architecture can describe summarization systems for more than one language. Part of this architecture includes text processing modules usually used in many natural language processing (NLP) applications, but some are built for the explicit purpose of being integrated into the summarization system and are adapted for one language or another. In all cases, the component modules are designed as standardized input–output black-boxes and added to the summarization system. While the concept of language independence is very much desired and discussed in modern NLP, there are few systems that can truly operate as language independent. Finally, what distinguishes our system from others is that our unique summarization architecture, which can be generically applied to more than one language, opens up some interesting possibilities for forming comparisons among languages. As a result, we have been able to identify some very interesting correlations between quantitative data characterizing the training corpora and the evaluation results obtained from our experiments.

In this chapter, we focus on presenting the individual NLP modules belonging to the proper summarization chain, the linguistic resources necessary for localization of these modules in the specific languages, along with our evaluation and results. The chapter is broken into the following sections: Section “Background” gives a brief background of the approach, Section “Anaphora Resolution” presents the anaphora resolution module, Section “Clause Segmentation”—the clause segmentation module, Section “Discourse Parsing”—the discourse parser, Section “The Summarizer”—the summarization module, Section “Corpora and Results”—the corpora used and the results, and Section “Conclusions” includes a discussion and some concluding remarks.

Background

In our experiments, the languages under consideration are Bulgarian, German, Greek, English, Polish, and Romanian, but, as mentioned earlier, our approach is general enough not to be limited to these specific languages. We targeted only short texts (less than 6 pages) and produced extract-type summaries out of the discourse structure. We describe in this chapter an extract type of summarizer, one in which summaries are made out of sequences of text spans (extracts, body of text) that are copied and pasted from the original input. As it will become obvious later on, the elementary pieces of text out of which we assemble summaries are discourse clauses. Our approach to short text summarization follows the one presented in Cristea et al. (2005), where the summary is generated from a treelike discourse structure of the original text. The discourse structures obtained resemble the Rhetorical Structure Theory (RST) trees (Mann and Thompson 1988), namely, because the constituent nodes evidence rhetorical relations between text spans that are either nuclei or satellites, and the terminal nodes are themselves elementary discourse units (*edus*). However, for our summarization goal in particular, we ignore the relation names and retain only the nuclearity markings from the discourse structure. The final output of the system consists of general summaries, but it is also possible to produce summaries focused on entities, characters, or events. For evaluation, we compare summaries extracted automatically against those indicated by human subjects.

The overall summarization system is truly multilingual in the sense that it first detects the language of the text and subsequently commutes to the specific language processing chain (LPC). Apart from small variations, all language versions have a similar design, as displayed in Figs. 6.1, 6.2, and 6.3.



Fig. 6.1 The block architecture of the summarization process

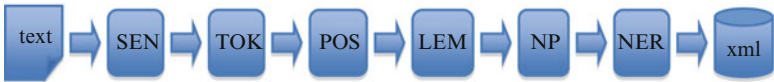


Fig. 6.2 Details of the summarization prerequisites



Fig. 6.3 Details of the proper summarizer chain

The Prerequisite part is a basic LPC, which includes the steps usually needed in many applications. In Fig. 6.2, this is indicated by placing the modules in a pipeline, although slight variations of this chain could be effective in different languages, depending on the proper realization of the envisioned functionalities.

The abbreviations in Figs. 6.2 and 6.3 have the following meaning: SEN = sentence splitter, TOK = tokenizer, POS = part of speech tagger, LEM = lemmatizer, NP = noun phrase chunker, NER = name entity recognizer, AR = anaphora resolver, CS = clause splitter, DP = discourse parser, SUM = summarizer, and SMO = smoothing module.¹

The intermediate format between the modules belonging to the Prerequisite chain and the proper Summarization chain is depicted here as xml, although, in order to cope with the standardization requirements of an international project,² each module has also been encapsulated into an UIMA CAS³ objects' interface. The UIMA modules of the resulted version will be referred to as the *integrated components* in this chapter. With a few exceptions, all modules implement a language-independent vision, in which processing reflects transformations applied to the input in order to obtain an enriched output, and a similar type of processing is performed in all languages. To obtain the specific behavior in one language or the other, the modules are fuelled with language-specific resources. We will not insist in this chapter on the Prerequisite processors, attentively described elsewhere.⁴

Anaphora Resolution

The Model and the Engine

The resolution of pronominal anaphors is important in a summarization task for at least two reasons: (1) we want the pronouns appearing in a summary to have their antecedents included in the summary; and (2) we want the position of the anaphor and the antecedent to be in correlation with the discourse-tree structure (Fox 1987; Cristea et al. 1999; Serețan and Cristea 2002). For these reasons, it is clear that building the discourse structure is a process that works in tandem with one that discovers antecedents of referential expressions. We describe here an approach in

¹Not described in this chapter is a module that simply performs cosmetic changes to the produced summaries, such as placing majuscules at the beginning of sentences, introducing commas between clauses, adhering punctuation signs to the previous words, and, in particular, replacing pronouns with co-referent proper nouns when all of the more informative antecedents happen to be left outside of the summary.

²The ICT-PSP ATLAS project, see Acknowledgements.

³<http://uima.apache.org/>

⁴http://ec.europa.eu/information_society/apps/projects/logos//7/250467/080/deliverables/001_ATLASD41LanguageProcessingChains1012609.pdf

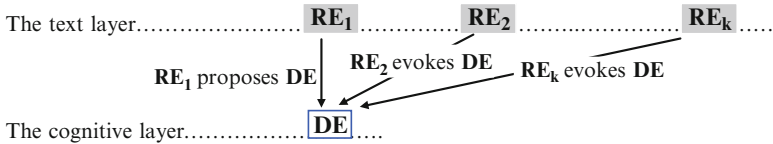


Fig. 6.4 Two-layer representation of a coreferential anaphoric relation

which the discovery of discourse structure comes after the resolution of anaphors and greatly benefits from it.

Anaphora is the phenomenon of reiteration of an entity (called “antecedent”) by a reference (called “anaphor”) that points back to that entity. (However, when there is no need to differentiate between anaphors and antecedents, we will call any one of these participants in anaphoric relations “referential expression” or “REs.”) As such, *anaphora resolution* (AR) is understood as the process of identifying the antecedent of an anaphor. For a proper understanding of a text, it is extremely important that pronouns, common nouns, and even proper nouns correctly recuperate their antecedents. Actually, during reading, it is very likely that an anaphor becomes, in its turn, an antecedent for another coreferential anaphor that appears later on in the text.

We anchor our AR mechanism on a cognitive model that describes the reading of a text as a mental process of developing abstract descriptions of the entities mentioned in the text.⁵ We will call *discourse entity* (DE) a semantic representation (placed on a cognitive layer) of a referential expression (residing on the text layer). In Fig. 6.4, the coreference relation between different participants of a coreferential chain is shown as a series of *propose–evoke–evoke* relations, linking the different textual realizations (i.e., REs), and their unique semantic (cognitive) representation (i.e., a DE).

In our approach the referential expressions are *noun phrases* (NPs), which include different surface forms of pronouns, common nouns, and proper nouns, with their modifiers (except for relative clauses). The notation for an NP should contain an indication for the head noun. NPs could have recursive structures, but in this case the corresponding heads should be distinct. Examples (NPs in brackets, heads—underlined) are the following: <John Smith>, <him>, <<her> hat>, <<the University> building>, <two cats>, and <a wonderful brunette in <a blue car>> .

As the reading progresses, a semantic representation is first born when RE₁ is encountered. Then, at a later moment when RE₂ is read, it evokes the DE already built by RE₁, and any subsequent co-referring REs will in turn evoke the same DE. One way of representing discourse entities in NLP systems is as feature

⁵In fact, the development of these mental structures attached to discourse entities is only one of the many processes that are performed during reading: recognition of events and filling their respective roles, interpretation of metaphors, correlation of time mentions, etc.

structures, which consist of lists of attribute-value pairs. The exact configuration of these attributes as well as their types (range of accepted values) are evidenced by the anaphora resolution model. The DE, thus, becomes a repository of features contributed by the different REs it connects and can be either stable during reading or can evolve from knowledge-poor to knowledge-rich representation.⁶ For instance, the coreferential chain *the professor ... she* implies the proposal of an initial DE, configured as [sem={person, professor}] during the reading of the RE *the professor*, which is then enriched to [sem={person, professor, female}] at the moment of the reading of *she*. Let's note that the reference *she* may include the feature [sem={person, ship}] and thus a partial match helps the resolution process.

The text is processed left to right, and a decision is taken each time a new referential expression is met. The motor leaves behind chains of coreferential expressions. Each chain is characterized by a data structure, recording all features of the REs in the chain. This is what we call a discourse entity—see above. To give an example, suppose John Smith, an ex-professor of computer science, 70 years of age, has been referred to in a text, as *a professor of computer science, John Smith, a 70 year old man, Mr. Smith, he, John, him, he, the old man, the professor, John Smith*, and *he* again. At the end of the text, RARE, ideally, leaves behind a DE which approximately includes the following feature structure [ID=DE009; SEM={person, professor of computer science, 70 years old man}; GEN=male; NAME={John, Smith}], as well as links to all the corresponding REs on the surface string.

The process runs as follows, while the text unfolds left to right. When a new RE, say RE_x, is met, its set of morphological, syntactic, and semantic features is tested against the recently proposed/evoked DEs which have been left behind by the engine. If, among them, there is one, say DE_y, for which the matching score of the pair (RE_x, DE_y) individualizes significantly well, then the actual RE_x is added to the already existing chain of referential expressions attached to DE_y, or else a new DE is generated (*proposed*—in Fig. 6.4), RE_x's features are copied onto it, and RE_x becomes the first referential expression of a new chain.

It is interesting to see that in this model the distinction between anaphora and cataphora is given by the order of distinct surface realizations: proper noun or common noun before the pronoun or vice versa. In fact, cataphora represents only an instance of a whole class of references in which a knowledge-rich reference enriches the semantic representation of an existent knowledge-poor DE, by bringing in new features. Another instance of the same class is coreferential sequences such as *an animal ... the elephant*, which can hardly be accepted as coreferential (compare to the sequence *an elephant ... the animal*).

⁶Sometimes, the representation can change dramatically during the unfolding text, as in the case of coreferential chains of the form *the child ... the young lady ... the woman*. A solution would be to keep more DEs on the cognitive layer, representing distinct instances of the same entity at different moments of time (Cristea and Dima 2001).

Resolution of anaphora in the multilingual summarization enterprise that we describe here has been performed with RARE (*Robust Anaphora Resolution Engine*) (Cristea and Postolache 2005)—a framework for building rule-based anaphora resolution tools.⁷ Its collection of symbolic rules uses weights which are optimized with genetic algorithms. The core of the system is language independent, and its localization to one language or another was assured by specific resources (see Section “Localization of RARE”).

Localization of RARE

The adaptation of the general RARE machinery to different languages was done by localizing a number of resources, the most prominent being the collection of rules incorporating matching conditions between the anaphor (seen as an RE) and the antecedent (seen as a DE). These rules are responsible for deciding whether a referential expression refers (evokes) one discourse entity already mentioned or introduces a new one.

There are three types of rules put to work on a pair (RE_x , DE_y):

- **Certifying rules:** If such a rule is evaluated to TRUE on a pair (RE_x , DE_y), it certifies without ambiguity the DE_y as a referent for the RE_x . For instance, identical proper names usually denote the same person. In the example above, the second RE *John Smith* is deciphered to refer DE_{009} with the help of such a rule (which has included already this name among its features).
- **Demolishing rules:** If such a rule is fired on a pair (RE_x , DE_y), it filters out DE_y as a referent candidate of RE_x . RARE includes a cabled demolishing rule, invalidating the attempt to establish a coreferential link between nested referential expressions. In the example above, this rule invalidates a coreferential link between *computer science* and *a professor of computer science*.
- **Promoting rules:** If such a rule is evaluated to TRUE on a pair (RE_x , DE_y), it increases a resolution score associated with the pair (RE_x , DE_y). A match of the condition expressed in such a rule adds to the overall resolution score associated with the pair a positive value. If RE_x has no certifying rule yet fired with any recorded DE, then that DE_y against which it has the best overall score among those with which RE_x experiences no triggered demolishing rule will be chosen. Supposing the text includes a sentence like *John Smith is a 70 year old man.*, such a rule could yield a coreferential link between the nominal predicate and the subject (more exactly, the DE the subject *John Smith* is referring to). Supposing this DE is DE_{009} , the rule will add to the DE’s set of features {*professor of computer science*, *John Smith*} a new one: *70 year old man*.

⁷Proprietary of UAIC-FII: <http://nlptools.info.uaic.ro>

Clause Segmentation

The Model

A *clause* is a grammatical unit of a sentence that includes, minimally, a predicate and an explicit or implied subject and expresses a proposition (Nguyen et al. 2009), a statement, or an event. Clauses could be continuous or interrupted text spans. The identification of clause boundaries is important for a number of NLP applications, such as machine translation, text-to-speech systems, parallel text alignment, and building the discourse structure and automatic summarization. In a rule-based approach, such as Leffa (1988), the clauses are reduced to a noun, an adjective, or an adverb. Parveen et al. (2011) and Orăsan (2000) describe hybrid methods, in which the results of a machine learning algorithm, trained on an annotated corpus, are processed by a shallow rule-based module intended to improve the accuracy. Pușcașu (2004) transfers the technique that Orăsan describes for English to Romanian sentences with good results. In Șoricuț and Marcu (2003), the discourse segmentation task is formulated as a binary classification problem of deciding whether or not to insert a segment boundary after each word in the sentence. Subba and Di Eugenio (2007) use artificial neural networks to segment sentences into clauses, which are then used as *edus* by a discourse parser. In Hilbert et al. (2006), the list of discourse markers, which indicate possible rhetorical relations, is manually developed.

Many discourse parsing and summarization techniques make use of clauses, such as elementary discourse units of the discourse structure and the building blocks of summaries. Our approach to discourse segmentation starts from the assumption that a clause should be headed by a main verb or a verbal compound. As such, the delimitation of clauses starts from the identification of verbs and verb compounds and then the clause boundaries are looked for in between these pivots. Verb compounds are lexical sequences in which one is the main verb and the others are auxiliaries, infinitives, or conjunctives that complement the main verb such that the semantics of the main verb in the current context obliges taking the whole verbal construction together. An example is “like to swim” (Ex. 1): placing a clause boundary between “like” and “swim” would separate the verb from one of its compulsory arguments.

Ex. 1 <When I go to the river,> <I like to swim with friends.>

The exact place of a clause boundary between verbal phrases is, in many cases, indicated by discourse markers (key words or expressions) like in Ex. 2.

Ex. 2 <Markers are good> <because they can give information on the discourse structure.>

Often, a discourse marker signals a rhetorical relation that glues together two text spans. When they are missing, such as between the 1st and the 2nd clause in

Ex. 3, boundaries can still be indicated by punctuation marks or other clues which, presumably, may be identified by statistical methods.

Ex. 3 <Although the snow was falling uninterruptedly,> <the slope was still in pretty good condition.>

The clause segmenter is trained on explicit annotations given in manually built files for all the languages under scrutiny. During the training of the segmenter, a window of n POS tags to the left of the candidate marker and m POS tags to the right defines the context. For the cases in which we do not have any marker at the boundary between clauses, a symmetrical window of l POS tags is used. The values of the three parameters m , n , and l are set at the calibration time.

Discourse markers may have one or several rhetorical functions. We have already mentioned that our discourse trees put in evidence only the nuclearity of arguments, while the name of relations is ignored. As such, to characterize markers, only the features relevant to their nuclearity patterns have to be retained and collected from the annotated corpus. In (Carlson et. al, 2001) the nucleus is defined as being more salient in the discourse structure, while the satellite represents supporting information. The notations N_N, N_S, and S_N represent the nuclearity (N = nucleus, S = satellite) of the two arguments around a marker. For instance, the “and” marker occurs in the English corpus 205 times with the N_N pattern and the “which” marker occurs 35 times with the N_S pattern and in only one case with the N_N pattern. The model is built using the *MaxEnt*⁸ library.

The *Training module* generates the markers’ model: For each marker, an attribute (TYPE) may take one of the two values (FRONT and BACK), representing the position of the marker relative to the boundaries found in the corpus. If a marker is annotated both ways, the value of the attribute TYPE will be decided based on the highest frequency. For example, if the manually annotated corpus displays more cases of clause boundaries found in front of the “and” marker than after it, then the value of the attribute TYPE will be FRONT. An example of such a segmentation is given in Ex. 4:

Ex. 4 <Verbs and verb compounds are considered pivots > <and clause boundaries are looked for in between them.>

The *Segmenter module* consists of two steps: First, it applies a machine learning algorithm to recognize if pairs of verbs can be taken as compound verbs and, second, it applies rules and heuristics based on pattern matching and machine learning algorithms to identify clause boundaries.

The training of the segmentation model is targeted at putting in evidence patterns of the markers’ uses upon which segmentation boundaries are decoded. Negative examples, optimally equal in number with the positive examples, are also searched for in the corpus in all cases of literals which can function as markers in some contexts and non-markers in others. Positive and negative examples are

⁸The maximum Entropy Framework: <http://maxent.sourceforge.net/about.html>

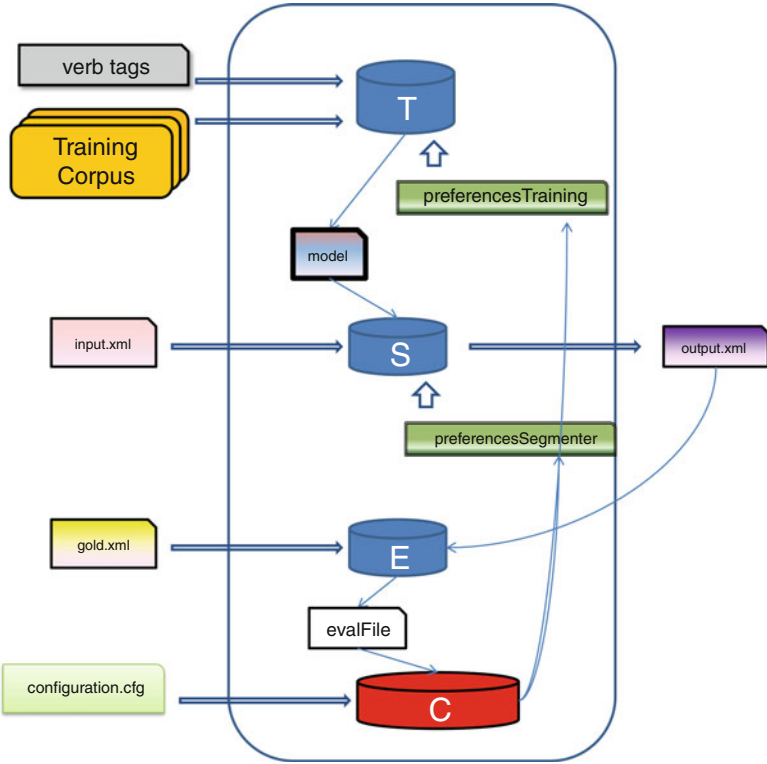


Fig. 6.5 The calibration chain for clause segmentation

also collected for clause boundaries which are not explicitly announced by markers. Ex. 5 gives a couple of negative examples for the cue words *and* and *that*—marked with strikethrough. These are cue phrases that could play the role of discourse markers only in some cases. It also shows a case of clause boundary where a marker is missing (between units ^[2] and ^[3]).

Ex. 5 <In times past there lived a king ~~and~~ queen,^[1] > <who said to each other every day of their lives,^[2] > <“Would ~~that~~ we had a child”!^[3]>

If, in the sequence of tokens between two verbs, the system will detect neither a marker nor examples in the corpus of clause boundaries without markers, the text will not be segmented.

Finally, an *Evaluation module* (E in Fig. 6.5) is used to compare a test file (the output from the Segmenter module) against a gold file (manually annotated at clause boundaries). Two metrics have been considered in doing this comparison. The first one calculates Precision, Recall, and F-Measure by comparing the number of boundaries in the test and gold files. The second metric, called *Accuracy* (A in the formulas below), is less restrictive and computes the inclusion of words in the proper segments:

$$A = \frac{S}{N}, \quad \text{where } S = \sum_{i=1}^N s_i;$$

$$s_i \min \left(w_i, \frac{1}{w_i} \right), \quad \text{where } w_i = \frac{L_T^{w_i}}{L_G^{w_i}}.$$

$L_T^{w_i}$ —represents the length of the clause the word w_i belongs to, in the test file.

$L_G^{w_i}$ —represents the length of the clause the word w_i belongs to, in the gold file.

s_i —represents a score attached to the word w_i , the same for all words belonging to the same test clause.

N —represents the total number of words in the test file.

S —represents the sum of scores of all words.

In a multilingual system, an important issue is the calibration of the system for each of the targeted languages. The multitude of parameters make manual calibration a procedure that is very delicate, time consuming, and prone to errors. In order to avoid this, we have developed an automatic calibration procedure intended to find that configuration of the system's parameters that would achieve the best segmentation results for each language. This makes the quality of a specific clause segmenter rely only on the quantity and quality of the manually annotated corpus and less on contextual data as reflected in the set of parameters. The Calibration module needs a configuration file, an input file, a corresponding gold file, and a training corpus. It iterates the sequence of modules < Training, Segmenter, Evaluation > on the whole scope of preference parameters until the best possible results are obtained. The specific values of parameters associated with the best run will be frozen and considered in the life-long routine of the system, as shown in Fig. 6.5.

Discourse Parsing

Incremental Parsing at the Discourse, Paragraph, and Sentence Level

Discourse parsing is the process of inferring the structure of a discourse from its basic elements (sentences or clauses), the same as one would build a parse of a sentence from its words (Bangalore and Stent 2009).

RST (Mann and Thompson 1988) is one of the most popular discourse theories. In RST the discourse segments (*edus*) are plain text units; their aggregation in larger segments configures an understanding about the meaning of their combination. This theory puts in evidence a whole class of relationships between segments of text that details the coherence of a text. A text segment assumes one of the two roles

in a relationship: nuclear or satellite. If a nuclear unit is deleted, the discourse may become incoherent, while if a satellite is lost, the discourse only loses some details. As many things at the level of discourse interpretation, where we talk about comprehensibility and degree of coherence, the distinction between nuclei and satellites is often subjective. The size of a text unit is arbitrary, but each should include a self-contained predication (De-Silva and Henderson 2005). As in many other approaches (Taboada and Mann 2006), in our model also the *edus* are clauses. Rhetorical relations (for simplification, binary), holding between nonoverlapping text spans, are of two kinds: *hypotactic*, and *paratactic*. Hypotactic relations connect satellites to nuclei, while paratactic relations hold between text segments of equal importance; they are considered nuclear.

Discourse structures have a central role in several computational tasks, such as question-answering, dialogue generation, summarization, information extraction, etc. The HILDA discourse parser (Hernault et al. 2010) is a text-level discourse parser with state-of-the-art performance. The system was trained on a variety of lexical and syntactic features extracted from a manually annotated corpus. Some of HILDA's features are borrowed from Şoricuţ and Marcu (2003), where the discourse tree is built with the help of two classifiers in a cascade—a binary structure classifier to determine whether two adjacent text units should be merged to form a new sub-tree and a multi-class classifier to determine which discourse relation label should be assigned to the new sub-tree (Feng and Hirst 2012).

Our *Discourse Parser* produces discourse trees that include nuclearity markings but lack rhetorical relation names. The terminal nodes of the discourse tree represent clauses (*edus*), while the intermediate nodes represent spans of text larger than an *edu*. It adopts an incremental policy in developing the trees, on three levels: the sentence level, the paragraph level, and the discourse level (representing the whole text). At each level, the parser goes on with a forest of developing trees in parallel, ranked by a global score that takes into consideration a number of heuristics (detailed in Section *Heuristics*). At each step in the process, the system retains only the best scored trees of the previous step. The aim of this pruning process is to master the exponential explosion of the developing structure.

This section gives a description of the basic incremental parsing approach. The input to the parser is the text augmented with information about SEN (sentence boundaries), TOK + POS + LEMMA (tokens with specification of their part-of-speeches and lemmas), NP (noun phrases, acting as referential expressions), DE (discourse entities, acting as coreference chains), and CLAUSE (clause boundaries, acting as *edus*). As already seen, this complex annotation is a result of the prerequisite modules, the RARE module, and the clause splitter module (see Figs. 6.2 and 6.3).

All generated trees observe the principle of sequentiality (Marcu 2000): *A left-to-right reading of the terminal frontier of the tree associated with a discourse must correspond to the span of text it analyzes in the same left-to-right order.*

Our incremental discourse parsing approach borrows the two operations used in (L)TAG (*lexicalized tree-adjoining grammar*) (Joshi and Schabes 1997): *adjunction* and *substitution*. The adjunction operation takes an initial tree or a developing tree

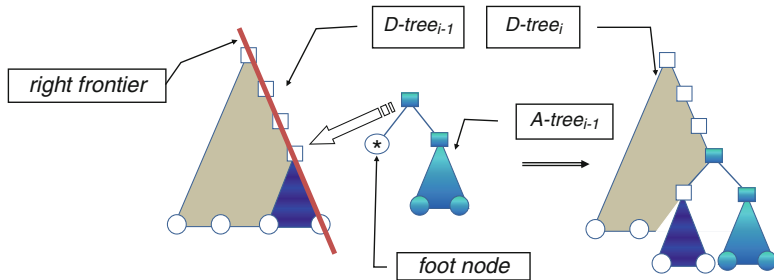


Fig. 6.6 The adjunction operation

(D-tree) and creates a new developing tree by combining it with an auxiliary tree (A-tree). The auxiliary tree includes a special node, called *foot node* (denoted by the * sign), which is placed on its terminal frontier. The adjunction operation temporarily dismounts the sub-tree headed by an adjunction node placed on the right frontier⁹ of a D-tree, attaches it to the A-tree by replacing the foot node, and finally replaces the adjunction node of the D-tree with the augmented A-tree. Figure 6.6 depicts this operation.

An auxiliary tree whose foot node is placed as a left child of its parent node is called a *left-footed auxiliary tree*. As proved in Cristea (2005), only left-footed auxiliary trees participating in adjunction operations on the right frontiers of D-trees maintain the correctness of intermediary D-trees at each step.

We start from the assumption that a discourse consists of several paragraphs, each paragraph has one or more sentences, and each sentence, in turn, has one or more clauses. At the paragraph and sentence levels parsing goes on incrementally by consuming, recursively, one entire structure of an inferior level. Treating these spans separately is possible based on the assumption that for each span there is a corresponding sub-tree in the overall discourse tree. What this means, for instance, is that a clause belonging to the sentence S_i cannot, by itself, complement the sentence S_{i-1} or a part of it (Şoricuţ and Marcu 2003).

Figure 6.7 displays all possible types of auxiliary trees. As it can be seen, the *alpha* and *beta* trees are left-footed A-trees, and therefore are appropriate for adjunction, and the *gamma* and *delta*, lacking foot nodes, are appropriate for substitution. On the one hand, when there are discourse clues suggesting an expectation, only the *beta* and *gamma* types can be used, because only these types include substitution nodes. On the other hand, the root nodes of any of these trees can have any of the nuclearity types: N_N, N_S, and S_N.

⁹The right frontier represents the path from the root of the tree to its rightmost leaf node.

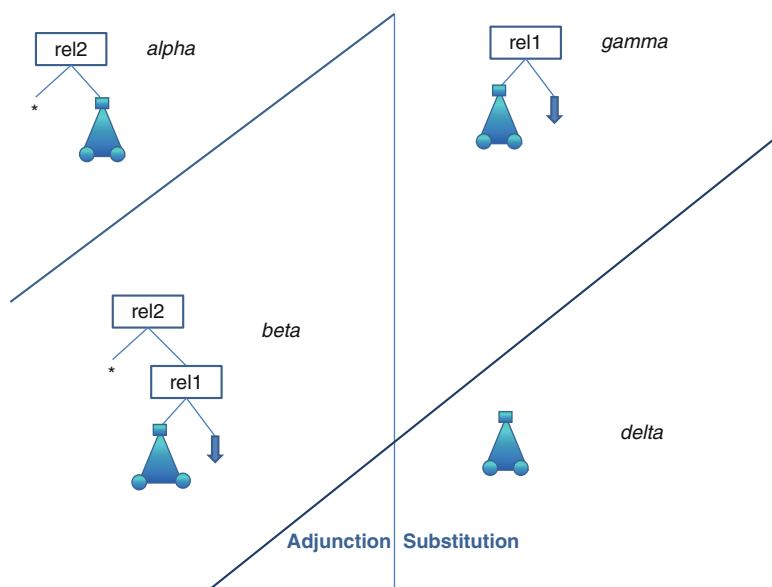


Fig. 6.7 Types of auxiliary trees

Combinatorics at the Sentence Level

For short sentences we apply a different approach (Cristea et al. 2003) that exploits the markers' patterns of arguments and uses combinatorics to explore the search space of solutions. A corpus was used to extract possible patterns of arguments of discourse markers. Following the identification of markers in the sentence, a list of all possible combinations of arguments is computed, by taking into account all patterns of all markers. The patterns are sensible to the position of markers within the clauses in the corpus. For example, the marker *because* may have both arguments to the right (in which case the first is satellite and the second nuclear, as in *Because it was raining, John took his umbrella.*), or one to the left and one to the right, in which case the first is nuclear and the second—satellite (*John took his umbrella, because it was raining.*).

After the lists are computed, that combination of arguments should be determined, which gives rise to a consistent, well-formed, tree structure. The well-formedness is checked against a set of rules (Cristea et al. 2003), as for instance it is impossible to have two distinct markers which cover the same sequence of *edus* and it is impossible to have nested arguments on both sides of the two markers.

The rules filter out the majority of combinations and what remains should be a list of valid trees. In the best cases, supposing a sentence contains ten clauses and nine markers and each of them has only one unique pattern, the system would generate

minimum $9^9 = 387,420,489$ combinations (because each marker has at least 9 lists), all having to be validated for well-formedness. The number of possibilities in a sentence with n clauses is 3^{n-1} (because a sequence of two *edus* can give rise to 3 structures: N_N, N_S, and S_N).

Constraints dictated by the necessity to have a response in real time have obliged us to apply the combinatorics method only on sentences shorter than 8 clauses. All long sentences are parsed using the incremental parsing approach described in Section “Incremental Parsing at the Discourse, Paragraph and Sentence Level.”

Heuristics

As mentioned already, the exponential explosion of partial trees is mastered by a ranking and pruning policy. Only the best ranked trees are retained in the process at each step. We describe in this section a number of heuristics used to assign scores to the developing trees, as ways to guide the elaboration of the final shape of the tree. For each tree t , a global score (GS^t) is computed by summing up weighted scores of each individual heuristic using the formula

$$GS^t = \sum_i^N s_i \times w_i,$$

where s_i and w_i are the scores of one heuristic and its corresponding weight, and N is the total number of heuristics applied. The score of each of the heuristics is normalized in the range 0 to 1. The weights themselves are established during a calibration process that resembles the one presented in Section “The Model.” The only difference is that, given the lack of discourse gold files (extremely costly to produce), we have calibrated the discourse parser by directly comparing summaries.

Centering on veins: Centering Theory (CT) (Grosz et al. 1995; Brennan et al 1987) is known as a theory of local discourse structure which models the interaction of *cohesion* and *salience* in the internal organization of a text. The four Centering transitions between subsequent utterances of a discourse segment (*continuation*, *retaining*, *smooth shifting*, *abrupt shifting*, to which the *no Cb* transition can also be added) can be taken as measures of discourse coherence (from the smoothest, easier to interpret one—to the most discontinuous, difficult to decipher).

There are known approaches that suggest that the granularity of Centering utterances can go down to clauses (Kameyama 1997). By seeing in an utterance an *edu*, a still fragile bridge is opened towards considering Centering transitions as a criterion to appreciate the coherence of a given discourse. Thus, there remains one big barrier: the limitation of locality. Moreover, Veins Theory (VT) (Cristea et al. 1998) puts in evidence a relationship between referentiality and discourse structure that helps to identify coherent subsequences in the original discourse, called veins.

These are exactly the segments looked for by CT. VT thus offers a way to extend the local conclusions of CT to the global discourse and, in so doing, a way to associate a measure of global coherence. However, our goal is to discover the best discourse structure characterizing a text. It seems natural to make the supposition that *among all possible tree structures that can be associated with a discourse, the true one displays the best global coherence score*.¹⁰ We have expressed this in the form of a heuristic that guides the elaboration of the structure: the parser favors adjunction positions that maximize the scores of CT transitions on veins. Applied persistently, this heuristic is expected to produce that tree structure that reflects the overall smoothest understanding of the discourse.

Lower adjunction levels: The heuristic favors adjunctions operated on the lower part of the right frontier (or innermost right frontier). The trees developed when this heuristic is persistently applied will be predominantly balanced to the right and downward. This corresponds to a discourse which most of the time adds details on the lastly mentioned issue. On the contrary, a tree developing to the right and upward corresponds to a discourse that always comes back to the original idea, completing it with new details.

Opening minimum referentiality domains: If the material node m contains a reference that can be satisfied by antecedents belonging to the VT domains D_1, \dots, D_k , give better scores to domains having fewer referents. The heuristic favors adjunctions on the upper levels of the right frontier. Indeed, supposing a predominantly left-balanced D-tree (in which most of the hypotactic relations have the nuclear daughters on the left side, actually very common), an A-tree also left balanced opens for the new material node, a domain of referentiality which is longer if the adjunction node is lower on the right frontier. In other words, if I go on adding details to the most recent topic, I have access to the largest part of what has been said until now. On the contrary, if I go on adding details to an old topic, I have access only to the old discourse.¹¹ Now look at this property the other way round: if an entity belonging to the new material node is bound to refer to a mention (which will become its antecedent) belonging to the old discourse, the material node can be attached anywhere on the RF, but if the reference link is directed towards a new mention (antecedent), the adjunction cannot be made except on the lower part of the RF. So, by favoring tree structures having minimal domains of referentiality, we force adjunctions to the upper levels of the RF. This heuristic will therefore counterbalance the tendency incurred by the previous one.

Maximum referentiality score: The heuristic favors adjunction positions where most referents of the material node find antecedents on the referentiality domains given by veins. In relation with the occurrence of the antecedent on the vein, the

¹⁰This supposition approximates empirical results on measuring the coherence of human-produced discourses by Centering scores, as of Cristea and Ifene (2011): *On average, human discourses have a degree of coherence which is slightly less than the highest possible*.

¹¹This is also conformant with the stack referentiality of the Attentional State Theory (Grosz and Sidner 1986).

syntactic category of the anaphor also counts, because not all referential expressions are equal in evoking power. We have started from the original experimental findings of Ide and Cristea (2000) where if the anaphor is (a) a zero pronoun, then it is compulsory that the vein contains an antecedent; (b) a clitic, then it is extremely desirable that an antecedent be on the vein; (c) an overt pronoun, then it is desirable that an antecedent be on vein; (d) a common noun, then it is good if an antecedent is on the vein; and (e) a proper noun, then it is not necessary to have an antecedent on the vein.

Subsequently, we have defined three cases for placing an antecedent: (a) the antecedent belongs to the previous unit (clause), where *previous* is considered with respect to the vein of the current unit; (b) the antecedent belongs to one unit of the vein of the current unit which is not the previous one; and (c) the antecedent does not belong to any unit of the vein of the current unit.

Conforming to a combination of these criteria, a score is computed for each anaphor of the current unit, and these scores are summed up. The heuristic favors adjunction positions that maximize this score.

Consume substitution nodes first! The heuristic instructs, in case the D-tree includes an open substitution node, to consume this substitution node first (by using a gamma or a delta A-tree) before proposing an alpha or a beta A-tree (see Fig. 6.7).

Invalidate unclosed expectations! The heuristic strongly discourages those structure development directions that leave unachieved trees when the whole text is consumed. It gives extremely low scores to trees which still have open expectation nodes.

The Summarizer

In this chapter we call a *short text* a text spanning between $\frac{1}{2}$ a page up to 6 pages. Our summaries belong to the category usually known as *excerpt-type summaries*,¹² which are summaries that copy contiguous sequences of tokens from the original text. Actually, in our case, such a summary should contain elementary discourse units that are copied and pasted from the original text.

In truth, the structure of a discourse as a complete tree gives more information than properly needed for summarization purposes. However, by exploiting the discourse structure, we expect to add cohesion and coherence to our summaries. Also, three types of summaries can be extracted from the discourse structure:

1. A general summary—which tells, in short, what is the whole text about
2. An entity-focused summary—showing what the text does say about a certain entity

¹²Contrary to an excerpt-type summary is a *rephrase-type summary*, which contains a reduced, freely produced, verbalization of the original text.

3. An *edu*-focused summary—the minimum text that is required to understand an elementary discourse unit in the context of the whole discourse

The simplest way to obtain a general summary is to take the vein expression of the root node.¹³ Similarly, an *edu*-focused summary is given by the vein expression of that *edu*. In short, because both the general and the *edu*-focused summaries are by themselves vein expressions, they inherit the coherence properties of veins.

The summaries focused on entities need some reflection. Suppose one discourse entity is traced and a summary focused on that entity is wanted. If there is only one *edu* in which the entity is mentioned, the vein expression of that *edu* gives a well-focused summary of the entity. A problem appears if the entity is mentioned in more than just one *edu*. Because there is no a priori reason to prefer one clause to any of the others, among those in which the entity is mentioned, it is clear that a combination of the vein expressions of each *edu* in which the entity is mentioned should be considered. We proposed more methods of building a final summary in this case.

The first method takes the vein expression of the lowest node of the tree that covers all units in which the entity is mentioned.¹⁴ Since the length of a vein expression of a node is dependent on the deepness of the node in the tree structure,¹⁵ this method results in shorter summaries. The second method considers that particular summary (vein expression) which includes most of the mentions of the entity. The third method simply takes the union of all vein expressions of the units that mention the central entity. Finally, the fourth method builds a histogram out of all vein expressions of the units mentioning the central entity and selects all units above a certain threshold. The last two methods do not produce vein expressions and therefore are more prone to incoherent summaries than the first two methods, the last one being the most exposed.

In general, the commander of a summary also suggests a desired length (in terms of a percentage of the initial length of the short text¹⁶). But, as it was evident from above, the lengths of our summaries are dictated by the veins or the combination of veins they include, and, as such, there is not an obvious way in which they could be controlled. Moreover, we can make the observation that by pruning all satellite nodes of a tree and collapsing parent nodes with daughter nodes having only one descendent, a tree is obtained whose head/vein expression of the root is equal to that of the original tree. This tree obviously contains only

¹³This is identical to Marcu's (1997) method for text summarization based on nuclearity and selective retention of hierarchical fragments, because his *salient units* correspond to *heads* in VT, and the vein expression of the root is its head expression.

¹⁴Let's note that this method could still produce a summary that ignores mentions of the chased entity. In this case one of the other methods should be used.

¹⁵Consistent with the discussions in Section "Heuristics," but formal proof of this is not our concern here.

¹⁶Let's note that indication of the length of the summary as a percentage of the original length is an option only in the case of short texts. A summary of a book, for instance, should be drafted in totally different terms.

nuclear nodes. This shows that a general summary (as the one given by the vein expression of the root) cannot itself be further summarized using veins. To cope with the necessity to control the length of a summary down the original length which resulted from vein expressions, heuristics could be applied. However, in all cases we enter an arena in which we are no more protected by the coherence properties of veins. Such heuristics could include the elimination of clauses that do not contain referential expressions participating in coreference chains, or clauses whose deletion is not harmful (although may contain REs that are part of coreferential chains) simply because other coreferential REs with good evoking power still remain in the summary.

The Summary Evaluation System

It is notoriously true that a gold corpus to be used in the evaluation of a discourse parser is very difficult to obtain due to at least three factors: First, the determination of the discourse structure of a text involves choices which do not always have only one solution because of subjectivity factors. That is, even very-well-trained human annotators could arrive at totally different structures for the same text. Secondly, the annotation process is extremely time-consuming. And, thirdly, the cost of such a process is generally high because of the complexity of the task and the high skills that are needed.

Considering all these factors, we have decided to use an indirect method for evaluating the DP module, which skips a direct confrontation of a discourse parse tree against a discourse gold tree and concentrates instead on the evaluation of a discourse structure and the evaluation of a summary. The idea is that a good summary cannot be due to anything but the result of a good discourse structure. Conversely, a poor summary reflects defects in the discourse structure.

But evaluation of summaries is in itself a tricky thing; and not surprisingly there is quite a bit of literature dealing with this topic. Since the summarization process is subjective, when it comes to building the summarization gold corpora we propose having more than one annotator for the same text. However, because our human annotators were instructed to produce only extract-type summaries and a summary was a sequence of clause IDs, a very good automatically produced summary is perhaps much closer to the one thought by a human than in cases when no such constraints would have been imposed. This is because the automatic summary will include entire clauses, the same as the gold summary does. Thus, if a clause is decided by both the program and the human to belong to the summary, then both summaries will include the whole sequence of tokens belonging to that clause. Unfortunately, this good news should be tempered to a certain extent by the possible errors of the segmentation module. In order to also cope with the errors introduced by the Clause Segmenter, the Summary Evaluation module computes Precision, Recall, and F-measure by comparing tokens (words) in the test against the ones in the gold summarization files.

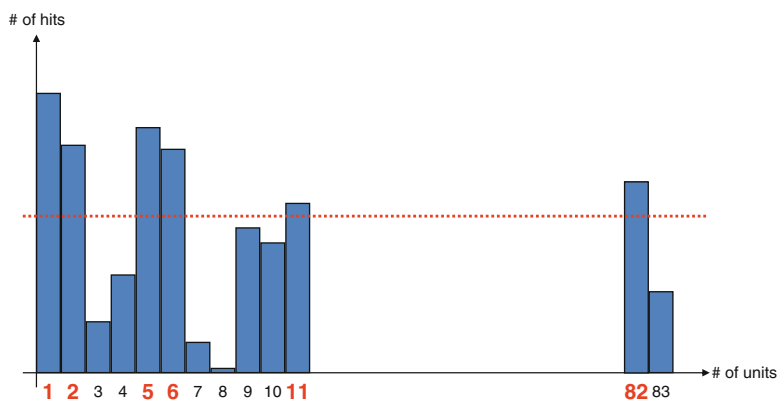


Fig. 6.8 Approximating a gold summary out of a number of human-produced summaries

To evaluate an artificial summary when there is more than one human-produced summary that counts as being gold, we adopted a measure based on a histogram, as proposed in Cristea et al. (2005). The human subjects received texts in which *edus* were already marked and numbered, and they were asked to indicate a 20% summary (word reduction rates). Then, a histogram was drawn by counting the number of times each *edu* from the original text was mentioned by the subjects as belonging to their summaries. In these histograms the sequence of *edu* numbers is placed on the x-axis and the frequency of mentioning on the y-axis. A sliding horizontal line (threshold) is fixed in this histogram at a position such that the number of units above the line approximates the 20% reduction rate. The respective golden summary is given by all units whose corresponding frequencies were above the threshold (see Fig. 6.8).

Corpora and Results

Two different types of corpora have been used in our experiments: containing clause boundaries and annotation of markers, and containing summaries.

The corpora of all languages included short texts of 2–4 pages each, from different domains: fairy tales, financial news, political articles, geographical descriptions, etc. The preprocessing chain was launched on each of these texts, producing XML markers, added to the original text, to put in evidence: sentence, clause and token boundaries (these including POS and LEMMA), and markers. The markers' attributes are NUC having the values "N_N", "N_S", "S_N", "NN", "_NS", "_SN", "NN_", "NS_", "SN_"; TYPE—with one of the values "int" or "ext", where TYPE="int" means that the marker is internal to a sentence, and TYPE="ext" means that the marker relates to a sentence from another span of text; for TYPE="ext", a CONNECT attribute was also filled in, with one of the

values: "expect", "fulfill", or "relate": CONNECT="expect" would have the meaning that the span unit the marker belongs to introduces an expectation, like the marker *on the one hand* in Ex. 6.

Ex. 6. from Cristea and Webber (1997): <On the one hand, John is very generous.>^[1] | <For example, suppose you needed some money.>^[2] | <Then, you would just have to ask him for it.>^[3] | <On the other hand, he is very difficult to find.>^[4]

CONNECT="fulfill" means that the span of text the marker belongs to fulfills (satisfies) an open expectation. This is the case with the *on the other hand* marker of Ex. 6, which indicates span [4] as fulfilling the expectation opened in [1]. CONNECT="relate" means that the span of text the marker belongs to relates in some way to the previous discourse, but neither raising nor fulfilling an expectation.

At the end of this process a program collected all markings from the corpus and abstracted the information related to them in a file. Subsequently, this file could be edited manually by the annotator with new markers that are defined manually. In fact, it was this kind of file that was used by the discourse parser in the process of building the discourse trees of the texts.

Summaries produced manually were used twofold in our experiments: to calibrate the parameters of the discourse parser and to finally evaluate the whole summarization chain. Each text in the summarization corpus was manually annotated by minimum of four subjects and all texts used in the evaluation had a compression rate of 20%. As mentioned above, the summaries displayed a list of clause IDs, indicating the clauses considered by the human subjects to be part of the summary. Table 6.1 shows the dimension of these corpora and the clause segmentation evaluation results (achieved by comparing the number of boundaries) for each of the languages under experiment.¹⁷ In the last column, the evaluation data represent averages over all languages.

When comparing the quantitative data with the evaluation results, there seems to be evidence of a number of correlations. For instance, it is clear that the dimension of the corpus (#tokens, #clauses, #markers, etc.) influences the quality of the segmenter. If we plot on the same graph the F-measures of all languages, in correlation with the number of markers of their corresponding corpora, a diagram, like the one in Fig. 6.9, results, on which a certain monotonicity tendency is observed. However, it can also be noticed that languages like GR and BG (whose F-measures are lower than the interpolation over all languages, which is marked by a thin line of the figure) seem to need more data for equivalent segmentation quality. We consider that the amount of data we have acquired are yet insufficient to risk any general, language-independent, statement regarding a stiff correlation between the dimension of the corpus and the performance of the clause segmentation module for

¹⁷The results are comparable with the state of the art. For instance, Pușcașu (2004) reports $P = 93.37$, $R = 91.43$, and $F = 92.38$ for English and $P = 95.59$, $R = 95.03$, and $F = 95.30$ for Romanian. Her method also uses rules to correct some boundaries.

Table 6.1 Segmentation corpora and evaluation

| Language | | BG | DE | EN | GR | PL | RO | Total/Avg |
|---------------|-----------|--------|--------|--------|--------|--------|--------|-----------|
| # sentences | | 2,749 | 1,375 | 2,246 | 1,055 | 1,096 | 1,571 | 10,092 |
| # tokens | | 51,116 | 31,839 | 53,504 | 30,207 | 21,377 | 47,016 | 235,059 |
| # clauses | | 6,468 | 2,726 | 4,880 | 2,778 | 2,574 | 3,720 | 23,146 |
| # markers | | 2,507 | 396 | 1,832 | 1,493 | 698 | 947 | 7,873 |
| # int markers | | 2,507 | 264 | 1,383 | 1,320 | 643 | 745 | 6,862 |
| # ext markers | “expect” | 0 | 16 | 117 | 1 | 1 | 183 | 183 |
| | “fulfill” | 0 | 83 | 61 | 30 | 2 | 204 | 204 |
| | “relate” | 0 | 16 | 270 | 133 | 0 | 544 | 544 |
| Evaluation | P | 0.97 | 0.93 | 0.98 | 0.90 | 0.89 | 0.91 | 0.93 |
| | R | 0.77 | 0.66 | 0.94 | 0.84 | 0.97 | 0.88 | 0.84 |
| | F | 0.86 | 0.77 | 0.96 | 0.87 | 0.82 | 0.89 | 0.86 |

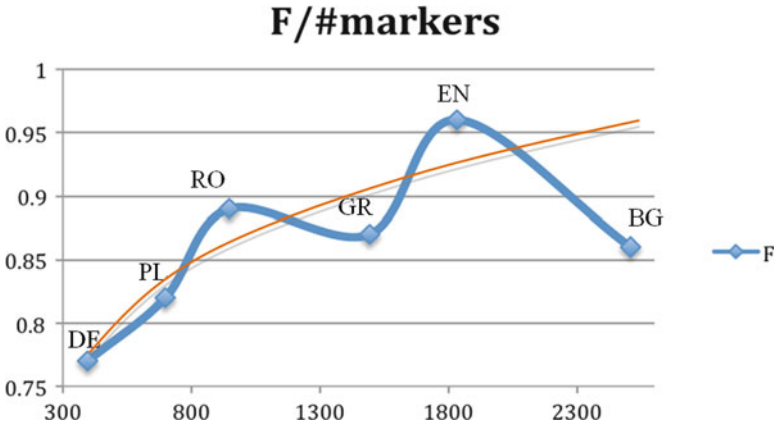


Fig. 6.9 The correlation between the #markers and F-measure

Table 6.2 The summary corpora

| Language | BG | DE | EN | GR | PL | RO | Total |
|----------------|-------|-------|-------|-------|-------|-------|--------|
| # of sentences | 1,168 | 781 | 489 | 692 | 541 | 526 | 4,197 |
| # of clauses | 2,955 | 1,815 | 1,499 | 1,742 | 1,303 | 1,317 | 10,631 |

a new language. But such laws, if revealed, could guide the design of the corpora when a certain quality is envisioned.

Table 6.2 shows the dimension of the main parameters characterizing the summary corpora in the 6 languages.

Finally, Table 6.3 contains a comparison of the summarizer results against two other known approaches. OTS¹⁸ is seldom used as a benchmark for other

¹⁸<http://libots.sourceforge.net/bench.html>

Table 6.3 Summary evaluation and baselines

| Language | | BG | DE | EN | GR | PL | RO | Avg |
|-------------------------------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| UAIC Veins Theory approach | P (H) | 0.19 | 0.23 | 0.27 | 0.23 | 0.17 | 0.22 | 0.22 |
| | R (H) | 0.29 | 0.44 | 0.41 | 0.41 | 0.36 | 0.32 | 0.37 |
| | F (H) | 0.23 | 0.30 | 0.32 | 0.29 | 0.23 | 0.25 | 0.27 |
| Open Text Summarizer approach | P (H) | 0.16 | 0.19 | 0.24 | 0.27 | 0.19 | 0.29 | 0.22 |
| | R (H) | 0.25 | 0.20 | 0.22 | 0.33 | 0.21 | 0.06 | 0.21 |
| | F (H) | 0.19 | 0.20 | 0.23 | 0.27 | 0.20 | 0.10 | 0.20 |
| LexRank approach | P (H) | 0.15 | 0.23 | 0.27 | 0.24 | 0.24 | 0.21 | 0.21 |
| | R (H) | 0.18 | 0.25 | 0.25 | 0.22 | 0.24 | 0.22 | 0.18 |
| | F (H) | 0.16 | 0.24 | 0.26 | 0.23 | 0.22 | 0.21 | 0.19 |

summarization systems. LexRank (Erkan and Radev 2004) computes the relative importance of textual units and sentences based on the concept of eigenvector centrality in a graph representation of sentences. Instead of passing words to the summarizer, we were passing sequences of numbers—token IDs, NP IDs, and NE IDs. In this way we made the input to the LexRank summarizer language independent. The figures in Table 6.3 are computed by comparing occurrences of IDs of words in the test against those in the gold summaries. As such, this metric is similar to the unigram-based ROUGE score (ROUGE-1), which is known to display the best agreement with human judgments (Lin and Hovy 2003) among all other higher gram ROUGE scores. The H’s appearing in parenthesis after the three evaluation measures (precision, recall, and F-measure) signify that the gold data used for comparison have been approximated out of the ones indicated by humans, by using the histogram method described in Section “The Summary Evaluation System.” As can be noticed (the best values are marked in bold), our summarizer behaves better globally (in terms of F-scores) than both of the other methods.

Conclusions

Some of the most well-known summarization systems today work on the assumption that similarity of sentences also indicates their importance. As such, to get ranked highly and placed in a summary, a sentence must be similar to other sentences that are in turn also similar to many other sentences. But a text which has a low degree of repetition may mislead the summarization system, which finds few elements to hinge on. Also, frequency-based summarizers practically disregard any concern about the coherence properties of the obtained summaries.

Our method of extracting the summary places coherence criteria at its core base. By evidencing coreferential links and aligning them with the discourse structure, the most plausible discourse tree can be built. The summary is then a direct product of this tree. We expect, therefore, that the resulted summaries show a higher coherence than those produced by frequency methods. One of the most important properties of

these summaries is the low occurrences of dangling pronouns. In the original text, a coreference chain is given by the list of REs attached to a DE. In principle, an antecedent of an RE could be considered any element of this list that is positioned to the left of the RE in the text. If the RE is a pronoun (a referential expression with a weak evoking power), most of the time the list of antecedents should also include elements with a higher evoking power (as proper nouns, for instance). In the process of text interpretation, the reader is able to recuperate the proper antecedents while also linking the referents in the proper chain. On the other hand, an excerpt type of summary includes only part of the clauses of the original text and, as such, the DE lists are shorter, some of them actually disappearing completely. If the process of summarization chaotically deletes units or is not driven by coherence principles, it could evidently trigger the disappearance of all high-evoking power referential expressions in the antecedents' chain of a pronoun. These are the dangling pronouns often mentioned in literature as negative side effects of summarization systems. The reading of our summaries reveals that our system is much more robust when it comes to this danger.¹⁹ This is due to the high scoring of discourse trees whose units include in their referentiality domains proper chosen antecedents. When this is not possible and a dangling pronoun escapes in the summary, it could be replaced, in a postprocessing phase, by a high-evoking power expression picked up from the pronoun's original coreference chain.

Still, the architecture that we describe has some drawbacks, because the rather complex processing chain may induce errors. We have identified different causes of these errors: defects in the prerequisite chain, technical defects in a component module of the proper summarization chain, quantity and quality of the corpora used for training different modules, and improper fixing of the parameters of different modules. For instance, a malfunctioning of the POS-tagger, which may tag a token as an adjective instead of a verb, might induce a clause segmentation error (because a verb is a pivot in the segmentation; see Section "The Model"); this triggers a discourse parsing error, which, in turn, rebounds during the summarization phase. As another example, an anaphora resolution error that is dropped behind by RARE may trigger a low scoring of a discourse tree, which, although correct, would be mistakenly rejected. So, aiming at a high quality for all component modules is compulsory. In most cases, the quality of a module is a direct result of two elements: the corpora used to train the attached model and the set of parameters. In principle, the higher and more accurately annotated the corpus is and the richer its set of parameters, the finer the calibration process can be designed and, consequently, the better its accuracy will be.

Our tests, operating in a practical setting²⁰ (Karagiozov et al. 2012), have shown that the system produces rather useful summaries. And a number of enhancements, mentioned below, can be readily foreseen. For instance, the Clause Segmentation

¹⁹A thorough evaluation on these grounds will constitute the basis of a future study.

²⁰The ATLAS system i-Librarian (<http://www.ATLASproject.eu>) and the ATLAS service EUDocLib (<http://eudoclib.ATLASproject.eu/>).

module could be placed before the Anaphora Resolution module in the processing chain. This way, it will become possible for RARE to also exploit the clause boundaries' information (one such example is the pronouns in different persons which cannot co-refer if they are in the same clause). The Discourse Parser module evaluates thousands of trees, before establishing which one is the best candidate for a discourse structure of a text. This process is extremely time-consuming and the DP module uses a multi-threading launch of the evaluation procedure in order to speed up the computations. A great part of this process involves the computation of veins. However vein expressions could be inferred without being computed exhaustively, by applying a set of formulas which use the head and vein expressions already existent in the developing tree and take into consideration the place of the adjunction/substitution of the RF (or inner RF) and other parameters.

Acknowledgements The work described in this chapter was supported by ATLAS (*Applied Technology for Language-Aided CMS*)—a project funded by the European Commission under the ICT Policy Support Programme, Grant Agreement 250467, and, partly, by the METANET4U ICT-PSP project, Grant Agreement 270893. Our thanks go to the following people: Anelia Belogay—for the impeccable leadership of the ATLAS project; Angel Genov—for mastering the Bulgarian chain and preparing data for Bulgarian; Walther von Hahn—for providing the linguistic support for German; Maciej Ogrodniczuk and Adam Przepiorkowski—for mastering the Polish chain; Polivios Raxis—for coordinating the evaluation processes; and Sabina Deliu—for preparing corpora and organizing the evaluation activity for Romanian.

References

- Bangalore S, Stent A (2009) Incremental parsing models for dialog task structure. In: Proceedings of the meeting of the European chapter of the association for computational linguistics (EACL), Athens, Greece
- Brennan SE, Walker Friedman M, Pollard, CJ (1987) A centering approach to pronouns. In: Proceedings of the 25th annual meeting of ACL, Stanford, pp 155–162
- Carlson L, Marcu D, Okurowski M (2001) Building a discourse-tagged corpus in the framework of rhetorical structure theory. In: Proceedings of the 2nd SIGDIAL workshop on discourse and dialogue, Denmark
- Cristea D, Dima GE (2001) [Anintegratingframeworkforanaphoraresolution](#). In: Information science and technology, vol 4, no. 3–4. Romanian Academy Publishing House, Bucharest, pp 273–291
- Cristea D, Ide N, Romary L (1998). Veins theory: a model of global discourse cohesion and coherence. In: Proceedings of the 17th international conference on computational linguistics ACL'98, Montreal, August, pp 281–285
- Cristea D, Ide N, Marcu D, Tablan V (1999) Discourse Structure and Co-Reference: An Empirical Study, In: Proceedings of the workshop on the relation between discourse structure and reference, Maryland, June 1999, workshop in conjunction with The 37th Annual Meeting of the Association for Computational Linguistics—ACL'99, Maryland, June, pg 48–57
- Cristea D, Iftene A (2011) Grounding coherence properties of discourse. In: ALEAR Final Report, vol II. Embodied Cognitive Semantics, Berlin, April
- Cristea D, Postolache O, Pușcașu G, Ghetu L (2003) Summarizing documents based on cue-phrases and references. In: Proceedings of the International Symposium on Reference Resolution and its Applications to Question Answering and Summarization, Venice, Italy, June

- Cristea D, Postolache O (2005) How to deal with wicked anaphora. In: Antonio Branco, Tony McEnery, Ruslan Mitkov (eds) *Anaphora processing: linguistic, cognitive and computational modelling*. Benjamin Publishing Books, Amsterdam, ISBN 90-272-4777-3
- Cristea D, Postolache O, Pistol I (2005) Summarization through discourse structure. In: Alexander Gelbukh (ed) *Computational linguistics and intelligent text processing*, 6th international conference CICLing 2005, Mexico City, Mexico, February 2005, Proceedings, Springer LNSC, vol 3406, ISBN 3-540-24523-5, pp 632–644
- Cristea D, Webber BL (1997) Expectations in incremental discourse processing. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid
- De Silva N, Henderson P (2005) Narrative support for technical documents: formalising rhetorical structure theory. At the International Conference on Enterprise Information Systems (ICEIS), Miami, FL, USA, 24–28 May 2005
- Erkan G, Radev D (2004) LexRank: graph-based lexical centrality as salience in text summarization. *J Artif Int Res* 22(1), ISSN: 1076-9757, AI Access Foundation, USA, pp 457–479
- Feng VW, Hirst G (2012) Text-level discourse parsing with rich linguistic features. In: Proceedings of ACL-2012
- Fox B (1987) *Discourse structure and anaphora: written and conversational English*. Cambridge Studies in Linguistics—48. Cambridge University Press, ISBN: 9780521330824
- Grosz BJ, Joshi AK, Weinstein S (1995) Centering: a framework for modeling the local coherence of discourse. *Comput Linguist* 12(2):203–225
- Grosz BJ, Sidner C (1986) Attention, intention and the structure of discourse. *Comput Linguist* 12:175–204
- Hernault H, Prendinger H, duVerle DA, Ishizuka M (2010) HILDA: a discourse parser using support vector machine classification. *Dialogue Discourse* 1(3):1–33
- Hilbert M, Lobin H, Bärenfänger M, Lungen H, Puskas C (2006) A text-technological approach to automatic discourse analysis of complex texts. In: Proceedings of KONVENS 2006, Konstanz
- Ide N, Cristea, D (2000) A hierarchical account of referential accessibility. In: Proceedings of the 38th Annual Meeting of the association for computational linguistics, ACL'2000, Hong Kong, China
- Joshi A, Schabes Y (1997) Tree-adjointing grammars. In: Rozenberg G, Salomaa A (eds) *Handbook of formal languages*, Springer, Berlin, ISBN: 978-3-642-63859-6 pp 69–123
- Kameyama M (1997) Intrasentential centering: a case study. In: *Centering theory in discourse*. Clarendon Press, ISBN: 978-0-19-823687-0
- Karagiozov D, Belogay A, Cristea D, Koeva S, Ogrodniczuk M, Raxis P, Stoyanov E, Vertan C (2012) I-Librarian—Free Online Library For European Citizens, in INFOtheca. *J Librarianship Inform* XIII(1):27–43, BS Print. Belgrade, ISSN: 1450-9687
- Leffa V. (1988). Clause processing in complex sentences. In: Proceedings of the first international conference on language resource and evaluation, vol 1, pp 937–943, May 1998
- Lin C-Y., Hovy EH (2003) Automatic evaluation of summaries using N-gram Co-occurrence statistics. In: Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics (HLT-NAACL), Edmonton, Canada
- Mann WC, Thompson SA (1988) Rhetorical structure theory: a theory of text organization. *Text* 8(3):243–281
- Marcu D (1997) The rhetorical parsing, summarization and generation of natural language texts, Ph.D. thesis, Department of Computer Science, University of Toronto
- Marcu D (2000) *The theory and practice of discourse parsing and summarization*. The MIT Press, Cambridge, MA
- Nguyen VV, Nguyen ML, Shimazu A (2009) Clause splitting with conditional random fields. *Inform Media Technol* 4(1):57–75, reprinted from: *Journal of Natural Language Processing* 16(1): 47–65 (2009) © The Association for Natural Language Processing
- Orăsan C (2000) A hybrid method for clause splitting in unrestricted English texts. In: Proceedings of ACIDCA'2000, Monastir, Tunisia
- Parveen D, Sanyal R, Ansari A (2011) Clause boundary identification using classifier and clause markers in urdu language. *Polibits Res J Comput Sci* 43:61–65

- Puşcaşu G (2004) A multilingual method for clause splitting. In: Proceedings of the 7th annual colloquium for the UK Special interest group for computational linguistics (CLUK 2004), Birmingham, UK
- Sereţan V, Cristea D (2002) The use of referential constrains in structuring discourse. In: Proceedings of The 3rd international conference on language resources and evaluation, LREC-2002, Las Palmas, Spain
- Subba R, Di Eugenio B (2007) Automatic discourse segmentation using neural networks. In: Proceedings of 11th workshop on the semantics and pragmatics of dialogue, Trento, Italy
- Taboada M, Mann WC (2006) Rhetorical structure theory: looking back and moving ahead. In: Discourse studies, vol 8, Nr. 3 (2006), pp 423–459
- Şoricuţ R, Marcu D (2003) Sentence level discourse parsing using syntactic and lexical information. In: Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology (HLT/NAACL), vol 1. Edmonton, Canada, pp 149–156

Chapter 7

Handling Two Difficult Challenges for Text-to-Speech Synthesis Systems: Out-of-Vocabulary Words and Prosody: A Case Study in Romanian

Tiberiu Boroș, Dan Ștefănescu, and Radu Ion

Abstract Given the unrestricted context for text-to-speech (TTS) synthesis and the current multilingual environment, TTS is often hampered by the presence of out-of-vocabulary (OOV) words. There are many precipitating factors for OOV words, from the use of technical terms, proper nouns, rare words that were not covered by the lexicon, and foreign words partially morphologically adapted; the latter, in fact, is a problem often confronted by non-English TTS synthesis systems. Furthermore, in order to derive natural speech from arbitrary text, all words that make up an utterance must undergo a series of complex processes such as: diacritic restoration; part-of-speech tagging; expansion to pronounceable form; syllabification; lexical stress prediction; and letter-to-sound conversion. OOV words require both automatic and trainable methods that can perform such tasks, which are usually based on a limited lexical context. The exception to this rule are those cases where part of speech and surrounding words are used as discriminative features such as in homograph disambiguation and abbreviation expansion. In this chapter we introduce the basic architecture of a generic natural language processing module in TTS synthesis, proposing data-driven solutions to various tasks, comparing our results concerning OOV words and prosody modeling with the current state-of-the-art TTS synthesis systems.

T. Boroș (✉) • D. Ștefănescu

Research Institute for Artificial Intelligence, Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

Institute for Intelligent Systems, The University of Memphis, Memphis, TN, USA

e-mail: tibi@racai.ro; dstfnsu@memphis.edu

R. Ion

Research Institute for Artificial Intelligence, Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

e-mail: radu@racai.ro

Introduction

Text-To-Speech (TTS) synthesis has been the main focus of a large number of studies for quite some time now, starting with the first generation of TTS systems which used rule-based speech synthesis (Allen et al. 1987). These systems were later followed by corpora-based methods, such as concatenative unit selection speech synthesis or Hidden Markov Models (HMM) speech synthesis (Tokuda et al. 2000). Regardless of the approach, much effort has been invested in trying to improve the naturalness of the synthesized voice and to increase the level of acceptance of TTS systems among its users. Still, the main difference between TTS systems and other systems that allow computer–human interaction using spoken language (e.g., interactive voice response (IVR) systems) is that a TTS system must be able to synthesize voice starting from arbitrary text. Some speech synthesis systems, however, currently achieve a remarkable level of quality and naturalness of the synthetic voice, either working with narrow domains or using large-scale speech corpora. Yet, there are number of problems that still plague text-to-speech synthesis today.

In this chapter we focus on solving the crucial problem of out-of-vocabulary (OOV) words in order to achieve better quality speech synthesis in the Romanian language. This is not an easy problem to solve given the fact that unlike digital signal processing (DSP), where there are many well-established methods and theories, things are not as clear for TTS systems which include natural language processing (NLP) and prosody generation. In fact, there are many different schools of thought in prosody modeling as well as a wide range of studies. Yet, there is still no “best” way to capture the essence of what makes human voices “better” than synthetic ones. Generating correct prosody is a delicate and complex area of study which, at this point, only relies on surface clues to what the voice based on the text should sound like. Obviously, this is not sufficient because in real-life settings the correct way of “saying” things (e.g., proper intonation, pauses, relative phoneme durations and word contrast) requires a deeper understanding of the message. And although some commercial systems for TTS do obtain surprisingly good results when dealing with arbitrary text, this is more likely due to hand-written rules combined with the effect of using a large-scale recorded speech corpus, while performing very little digital signal processing.¹

One of the main impairments of Romanian TTS synthesis is the lack of freely available resources. It is only in recent years that the Romanian Speech Synthesis (RSS) database (Stan et al. 2011) has been freely released for research purposes. This has undoubtedly aided a lot of research in this field. One of our recent projects was to build a prosody annotated corpus, based upon a section of the RSS database. The corpus is composed of a mixture of data obtained from all the NLP methods that will be presented in this chapter, with an additional prosody layer that was manually

¹For example, one of the leading TTS systems, IVONA, uses a unit-selection algorithm with limited time-scale modifications (Kaszcuk and Osowski 2009) producing remarkably good results in terms of naturalness and intelligibility of the synthetic voice.

created. Our prosody layer uses the Tone and Break Indices (ToBI) (Silverman et al. 1992) standard for annotation with the adjustments introduced by Jitcă et al. (2012) to suite the Romanian prosody phenomena. The corpus is available through the META-SHARE platform.²

Background

The typical set of NLP steps involved in any TTS task consists of preprocessing (text segmentation, part-of-speech (POS) tagging, shallow parsing, diacritic restoration, etc.), text-normalization (converting abbreviations, acronyms, numeric entities to their spoken form), phonetic transcription, syllabification and lexical-stress assignment. Any real-world TTS system uses lexicons for these tasks, but regardless of the coverage they might have, the system should be able to handle unknown OOV words. To integrate OOV words into the NLP flow of the TTS system, machine-learning (MT) techniques are devised in order to process them. Since there are significant differences in the quality of speech synthesis among various systems resulting from their ability to correctly handle OOV words, it is critically important to study this issue. Though a lot of research has been carried out for the above-mentioned NLP steps and current state-of-the-art methods do yield good results, most of the existing open-source implementations, however, rely either on methods whose results may be poorly documented or on outdated methods that are not competitive with current state-of-the-art proposed approaches. We enumerate below several such TTS open-source systems which are mainly intended for English but, with proper lexicons and training data, they can also work for other languages:

- DFKI MARY (*Modular Architecture for Research on speech sYnthesis*) is a result of collaboration between DFKI and the Institute of Phonetics at Saarland University. It was initially designed for German (De) but it includes a Voice Creation Toolkit and it currently provides a TTS interface for English (En), Russian (Ru), Italian (It), Turkish (Tr), and Telugu (Te). Some of the modules for English use tools provided with Festival. MARY TTS uses a HMM POS Tagger implemented after TnT (Brants 2000) and it implements a custom method for letter-to-sound (LTS) conversion of OOV words.
- Festival is a TTS system developed at Carnegie Mellon University (CMU). The NLP processing module of Festival implements CLAWS for POS tagging (DeRose 1988) and it uses CART (Black et al. 1998) method for building letter-to-sound rules on OOV words. Typical pronunciation errors in Festival have the following reasons³: (1) letter-to-sound rules fail on OOV words, (2) foreign proper names often fail, (3) wrong POS identified (newspaper headlines are particularly difficult), (4) POS is right but it is not in the lexicon and (5) POS is

²<http://ws.racai.ro:9191>.

³http://festvox.org/festtut/notes/festtut_toc.html#TOC42.

not enough to differentiate pronunciation (and not yet dealt with by homograph disambiguation CART).

- Flite is a derivative of Festival and uses similar methods and techniques for the task of phonetic transcription of OOV words.
- FreeTTS, which is based on Flite, is likewise a derivative of Festival and uses similar methods and techniques for the task of phonetic transcription of OOV words.

To our knowledge (see Table 7.1), little information is available about the individual performance of each module embedded in the above-mentioned TTS systems. Only letter-to-sound and POS tagging modules are reasonably documented, as they implement well established methods for performing subtasks in the text preprocessing step of TTS synthesis. Although some of these methods were considered to be optimal about a decade ago, they are now far behind the current state-of-the-art for designing well-functioning TTS systems. For example, CART method (Black et al. 1998) for building letter-to-sound rules, obtains an accuracy of 57.8% on OOV words when used on the English CMUDICT, while the current state-of-the-art letter-to-sound methods (e.g., MIRA) have an accuracy of about 70% using the same data. Far less information is provided for other modules such as syllabification and lexical stress prediction. Most systems implement custom, undocumented methods for such tasks, and computing the OOV accuracy for them is complicated, thus requiring substantial effort.

RACAI Text-Processing for Romanian TTS

In the sections below, we describe a tool chain, named Bermuda, for TTS synthesis. This system has been entirely developed at the Research Institute for Artificial Intelligence (RACAI) of the Romanian Academy. Bermuda, as well as the trained Romanian models, were developed within the Metanet4U European project and are available on the MetaShare distribution platform.⁴ The tools provide easy access to a low-level annotation schema instance that is the result of the text preprocessing step of our TTS system. The output is encoded as an XML that contains information such as morpho-syntactic descriptors, phonetic transcriptions, syllabification, lexical stress, prosodic modifiers for pitch and/or duration, pitch accents, boundary tones, break indices and custom attributes that can be added (e.g., word prominence and emphasis).

Although our experiments have been centered on Romanian, most of the tools are data-driven and can be trained for other languages as well. We provide detailed comparisons between the methods implemented by the above-mentioned open-source systems and those developed at RACAI.

⁴<http://ws.racai.ro:9191>.

Table 7.1 Details on open-source TTS systems NLP modules. These are accuracy figures on all/only-OOV words. For phonetic transcription and lexical stress assignment, the statistics are computed on CMUDICT

| System | POS-tagging | Phonetic transcription | Syllabification | Lexical stress assignment |
|---------------------------|--|---|-----------------------------|---------------------------|
| DFKI MARY | TnT | ML/under-documented | Rule-based/under-documented | Rule-based/not documented |
| Festival | 96.7%/85.5% similar to CLAWS 97%/– | –/ CART with EM-derived L2P alignments–/57.8% | –/ Not documented–/– | –/ CART–/62.79% |
| Flite (based on festival) | Not documented–/– | CART–/57.8% | Not documented–/– | Not documented–/– |
| FreeTTS (based on flite) | N/A–/– | CART–/57.8% | Not documented–/– | Not documented–/– |

Performance of Our System on OOV Words

The basic lexical stock of any TTS system consists of words with phonetic transcription, syllabification, and lexical stress. However, regardless of the size of such lexicons, there are always OOV words that need to be taken care of. Most statistics report performances considering all the tokens of a text, irrespective of their presence or absence within the lexicon backing up the processing strategy. While such evaluations are informative to the final user, indicating the overall quality of the system (besides making the numbers more appealing), from the developer's point of view, the evaluation on the OOV words is more productive, for it is the way systems handle OOV words which differentiates them. Thus, when we introduce our tools and calculate their performance rate, we factor in OOV words, though the overall performance of the system upon testing on an arbitrary text is usually higher since not all words are OOV.

What this means in practical terms is that, for instance, when we obtain a result of 60% accuracy on OOV words, if there are, say, only 10% OOV words in a 100-word text, the overall accuracy rate is 96% because there should be only four words that are incorrectly processed. If we talk about 90% accuracy on OOV words, the overall accuracy measured in this case should be 99%. Also, some of the lexicons we use are small (~1,500 words for Romanian phonetic transcription) and increasing their size has the following effects: the overall accuracy increases (meaning there are fewer OOV words); the OOV accuracy also increases when there is more training data to learn from), which again leads to a higher overall accuracy. Another aspect is that sometimes incorrectly processing a word has little or no impact on the quality of the synthesized speech. For example, if the phonetic transcription and lexical stress are correct, there will be no impact on the synthesized voice. Incorrect phonetic transcription, however, may be either unobservable or perhaps even nonchalantly accepted by listeners. For example, the Romanian word “este” (is) is correctly pronounced as “j e s t e,” but sometimes even native speakers start neglecting the initial “j” and just pronounce the word as “e s t e,” rendering this version of the pronunciation of “este” one that can be readily accepted by listeners.

Text Preprocessing

Sentence Splitter

Our Sentence Splitter is inspired by Reynar and Ratnaparkhi (1997) for identifying sentence boundaries using the Maximum Entropy (MaxEnt) framework.⁵ Our goal was to construct a domain-independent system, which can be used for any Roman

⁵MXTERMINATOR at: <http://web.mit.edu/course/6/6.863/tools/jmx/MXTERMINATOR.html>.

alphabet language. In this respect, RACAI's Sentence Splitter is similar to the second system proposed by Reynar and Ratnaparkhi in their paper. However, our approach has some significant differences: the most notable difference is that the system developed by Reynar and Ratnaparkhi is trained on the already tokenized Wall Street Journal corpus and some of its features are strongly dependent on this tokenization. Therefore, in order to detect sentence boundaries for a new text, this text must be tokenized in the same manner, which might be impossible in case the tokenizer is not available. In order to avoid this issue, we decided to use features which are not sensible to tokenization. For a sentence boundary candidate, our system looks at 12 features: the previous (and next) nonspace character is or is not: letter, uppercased, digit, symbol or punctuation (ten features), plus the lengths of the previous and next space-delimited character strings. We avoid using the previous token as a feature or adding additional lists of abbreviations, since these are language dependent. For the Maximum Entropy classifier, we used SharpEntropy, which is a C# port of the well-known MaxEnt toolkit written in Java.⁶ We conducted experiments on both Romanian and English text and we estimate that this new sentence splitter outperforms our current sentence splitter,⁷ which splits a text into sentences according to language-dependent rules. A thorough evaluation is currently under study. Our Maximum Entropy Sentence Splitter can be freely downloaded as a C# library.⁸

Diacritic Restoration (Romanian Version)

This processing step is particularly relevant for Romanian. Diacritic restoration is one type of spelling correction in which the correct diacritical mark of a letter is inserted in a word which would otherwise be incorrect, have a different (unintended) meaning or violate different syntactic constraints for the language in question. Thus, the decision to insert the diacritic is based on the context of the word and, for Romanian, we differentiate among the following cases:

- The word is incorrect according to a predefined (large) lexicon but a diacritic version of it exists in the lexicon, e.g., “maşina” is correct, “masina” is not.
- The word does not possess the correct diacritic form to agree with its syntactic constraints, e.g., the indefinite noun in “o mamă” (“mother”) is correct but its definite form is not “o mama” (“the a mother”).
- The word does not have the intended meaning in context, e.g., the word “fata” means “the girl” but word “faţa” means “the face.”

The main idea of our diacritic restoration program called DIAC+ (Tufiş and Ceauşu 2008) is that the morpho-syntactic information obtained by POS tagging the diacritic-free text is, for the vast majority of cases, sufficient to solve the ambiguities

⁶<http://www.codeproject.com/Articles/11090/Maximum-Entropy-Modeling-Using-SharpEntropy>.

⁷This splitter is integrated into the TTL platform. See Section on “POS tagger”.

⁸<http://nlptools.racai.ro/nlptools/index.php?page=ssplit>.

that occur when deciding whether to introduce a diacritic or not. For instance, the sequence “o mama” is tagged with an indefinite article and an indefinite noun but the only correct form for “mama” when it is an indefinite noun is “mamă”.

In Romanian (cf. Tufiş and Ceaşu 2008), on average, every third word of an arbitrary text contains at least one diacritical character. In terms of characters, more than 8.2% have diacritical signs.

In line with the discussion at the beginning of Sect. “Background,” if we consider all characters of a text, the diacritic restoration accuracy of the DIAC+ is 99.4%. However, if one takes into account only the characters that require diacritics, the accuracy figure is around 95.1%.

Irrespective of how one measures the diacritic restoration accuracy, most errors occur in words for which different diacritical variants have the same morpho-syntactic description (type 3 above) but different meanings (fata-faşa-fâşa).

POS Tagger

Part-of-Speech (POS) tagging is a technique used in many applications such as parsing, information extraction, word sense disambiguation, and machine translation, which applies a unique label (tag) to every token (words and punctuation) inside a sentence. The labels are called POS tags and the entire inventory of POS tags is called a tagset. A typical, though not the only, usage of POS tagging in TTS systems is to distinguish pronunciations for heteronyms if their phonetic transcription and/or lexical stress depend on part of speech. A number of statistical methods for POS tagging have been proposed over the years such as Hidden Markov Models (HMM) (Brants 2000), Maximum Entropy Classifiers (Berger et al. 1996; Ratnaparkhi 1996), Bayesian Networks (Samuelsson 1993), Neural Networks (Marques and Lopes 1996) and Conditional Random Fields (CRF) (Lafferty et al. 2001).

Our POS tagger is called TTL (Ion 2007; Tufiş et al. 2008) and it uses the tiered tagging technique (Tufiş 1999), which was developed for dealing with large tagsets. In order to cope with data sparseness, it uses a reduced tagset composed of CTAGs. A CTAG is a generalization of a richer POS tag called a Morpho-Syntactic Descriptor (MSD) from which context-irrelevant (and dictionary retrievable) attributes have been removed. For instance, the attribute for gender (masculine “m” or feminine “f”) from MSDs “Ncfsrn” and “Ncmsrn” is deleted to obtain the CTAG “NSRN” because the gender information can usually be inferred from the word form’s inflection analysis. Thus, the tiered tagging process uses CTAGs to obtain very good performances due to the fact that fewer CTAGs mean better statistical confidence in MLE parameter estimates. Then, using dictionaries, linguistic rules and ML techniques in the case of unknown words (Ceaşu 2006) the tiered tagging method has to automatically map from CTAGs to MSDs.

TTL is a Perl module, also exposed as a Web service,⁹ which performs the following text preprocessing operations: tokenization, POS tagging using the tiered

⁹<http://ws.racai.ro/ttlws.wsdl>.

Fig. 7.1 TTL MSD
ambiguity resolution rule

```
choose ^Di..#$ if
+1 Nc.# or
+1 ^Np or
+1 ^[ARM] and +2 Nc.# or
+1 ^[ARM] and +2 ^Np or
+1 ^[ARM] and +2 ^[ARM] and +3 Nc.#
end
```

tagging methodology, lemmatization, and shallow parsing (detecting the boundaries of noun, verbal, adjectival/adverbial, and prepositional phrases). The POS tagging module of TTL is a reimplementaion of the TnT HMM POS tagger of Thorsten Brants (2000) extended with several heuristics designed to improve the accuracy of the POS tagging process:

- All the named entities that are recognized when the text is tokenized (e.g., real and integer numbers, units of measurement, dates, times, abbreviations, and multiword expressions) are assigned fixed POS tags (when no ambiguity is present) such that the Viterbi output of the best POS tag sequence assignment will always contain these tags.
- In the case of lowercase unknown words, the guesser of the POS tag will take into consideration only the open-class categories: nouns, verbs, adverbs and adjectives.
- When encountering uppercase variants of otherwise unknown words (e.g., still unknown after conversion to lowercase or title case), TTL will always assign the “proper noun” (Np) MSD to these words.

As mentioned before, TTL implements the tiered tagging methodology. Thus, the HMM tagging described above is performed using CTAGs and then, a recovery process is initiated that will determine the correct MSD for every tagged word. This process goes through the following steps:

1. If the word with its currently assigned CTAG is found in the tagger’s dictionary, and the word/CTAG pair unambiguously maps to a MSD, this MSD is assigned.
2. If the word/CTAG pair maps to several MSDs, then language-dependent and context-aware rules will decide which of the mapped MSDs is applicable in the context. For example, in Fig. 7.1 we present the rule to disambiguate between the indefinite pronoun and indefinite determiners MSDs which map to the same CTAG “PI”. The rule says that the determiner reading is to be selected if the word is followed (position +1) by a common noun with which it agrees in number (the attribute value for “#” is the same) or if the next word is a proper noun or if the next word is an adjective, adverb or numeral (^[ARM]) and the word after that (position +2) is a common noun which agrees with the determiner in number, etc.
3. If the word is not found in the tagger’s dictionary, the word is assumed to belong to the open-class category (nouns, verbs, adjectives or adverbs). Then, by a suffix analysis which is dependent on the POS, following the TnT suffix analysis approach, the appropriate MSD is selected.

The usage of tiered tagging differentiates our tagger from the taggers in other systems, due to the fact that it is able to cope with the data sparseness specific to highly inflectional languages, which require large tagsets (Romanian, Hungarian, Czech, Estonian, Slovenian, etc.).

Text Normalization

This step is responsible for expanding certain expressions into words, before speech synthesis can happen. Arbitrary texts contain numbers, dates, abbreviations, acronyms, numbers, etc., which are not suitable for a direct conversion into a phonetic representation.

Text normalization raises a series of challenges mainly because of the ambiguities that may occur in selecting the optimal expansion course for each type of expression (number, abbreviation, acronym, etc.).

The input text is normalized using a set of handwritten rules and a list of well-known abbreviations. Every DOT character is removed from the text (after the expansion of abbreviations and expressions) except for sentence boundaries. Known abbreviations and some expression types are expanded and other (unknown) abbreviations or acronyms are converted to spoken form based on a letter-by-letter rule (e.g., “LRC” → “lerece”¹⁰). The accuracy of this module cannot be measured on OOV words, because there is no possible way of determining how to expand an abbreviation which is not found within the lexicon.

Phonetic Transcription

Phonetic transcription (also referred to as grapheme-to-phoneme (G2P) or letter-to-sound (L2S)) can be formalized as finding a relation between letters and corresponding phonemes, which is not a straight forward task and may pose some challenges for languages such as English. For Romanian, phonetic transcription rules are relatively simple compared to English or French (Burileanu et al. 1999), but there are several exceptions that need to be managed. For the purpose of language independence, data-driven methods are preferable as they only require words and their phonetic transcription equivalents for training, which are easier to obtain than wide coverage set of phonetic transcription rules. Several Machine Learning (ML) methods have been proposed for the PT task: Black et al. (1998), Jiampojamarn et al. (2008), Pagel et al. (1998), Bisani and Ney (2002), Marchand and Dampier (2000) and Demberg et al. (2007).

¹⁰This example corresponds to a conversion for Romanian.

The novelty of our approach consists in the two-layer architecture we used:

1. *The first layer* deals with the phonetic transcription of words by two possible options: a MaxEnt classifier and a custom algorithm called *Dictionary-Lookup or Probability Smoothing* (DLOPS).
2. *The second layer* contains an automatic error correction method, which is trained to relabel the output of the first-layer methods, based on the already predicted phonetic transcriptions. This significantly improves the accuracy of the system from 1 to 7% on OOV words.

All the data-driven methods for phonetic transcription require alignments between letters and phonemes. For so-called phonetic (or pseudo-phonetic) languages (e.g., Romanian), the task of grapheme to phoneme conversion is significantly easier and more accurate than for many other languages (such as English). However, there are several issues, common to several languages. The simplest example is that not all words have the same number of phonemes and letters and even if this condition is satisfied, it still does not imply an one-to-one alignment (e.g., experience—IH K S P IH R IY AH N S, where the letter *x* spawns two phonemes “K” + “S” and the ending “e” is silent; a similar phenomenon happens when we phonetically transcribe the word Romanian “experiență” (experience) into *e k s p e r i e n t s @*, where again *x* spawns “k”+“s”). Expectation-Maximization (EM) can be used to find one-to-one or many-to-many alignments between letters and phonemes (Black et al. 1998; Jiampojamarn et al. 2008; Pagel et al. 1998).

Letter-to-Phoneme Alignment

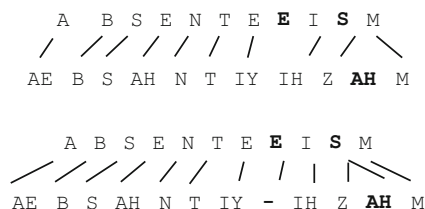
Bermuda uses the services of GIZA++ (Och and Ney 2003) in order to find alignments between the input word segmented at letter level and its corresponding phonemes. GIZA++ is a free toolkit for aligning items in a parallel corpus. In the past, it has mainly been used to create training data for machine translation (MT) systems but it can also be used to preprocess training data for L2P conversion systems as shown by Rama et al. (2009).

For each training lexicon we run GIZA++ for a primary letter-to-phoneme alignment with default parameters (ten iterations of IBM-1, HMM, IBM-3, and IBM-4 models). The dictionary is split into two files: the first file contains one word per line with its letters separated by spaces, so that GIZA++ will treat them as words in the source language. The second file contains phonetic symbols that “translate” the corresponding word on line number *n*, also separated by spaces (regarded as words in the target language). Each line in the dictionary used by our training algorithm has the following format:

$$L_1 < \text{space} > L_2 < \text{space} > \dots L_n < \text{TAB} > P_1 < \text{space} > P_2 < \text{space} > \dots P_n,$$

where L_i is the *i*th letter of the word and P_j is the *j*th phoneme in its phonetic transcription. Notice that the two sequences have the same number of elements but

Fig. 7.2 Alignment correction



not all phonetic transcriptions have this property. To obtain this property, we used an encoding where, if a letter does not emit any phoneme, we assign it the empty phoneme “-” (the dash character) and if a single letter emits more than one phonetic symbol, we separate the two symbols with the character “.” (the dot character). When GIZA++ does not have a match for a certain phoneme in the transcription, it aligns it with NULL (a “missing” word in the source language).

An unaligned phoneme must be generated by one of the close-by letters. When GIZA++ aligns one phoneme to NULL, we use a simple correction algorithm that looks at the letters that emitted the previous and the next phonemes and links the unaligned phoneme to the letter with which it was most frequently aligned. If there is a tie, the algorithm chooses the previous letter. As shown by others (Bisani and Ney 2002; Marchand and Dampier 2000; Jiampojamarn et al. 2008; Demberg et al. 2007), a phonetic transcription (for a single letter) more often than not depends on close letters. As an example, let us illustrate the alignment correction for the word *absenteeism* (Fig. 7.2). Between the phonetic symbols emitted by S “Z” and M “M,” we have the unaligned (or NULL aligned) symbol “AH”. In this case, we correct the alignment by assigning the phoneme “AH” to the letter “s,” as “AH” between “Z” and “M” was most frequently aligned with “s” (next to “m”).

The correction algorithm also inserts the empty phoneme “-” for every NULL aligned letter. In Fig. 7.2, the letter at position 8 (bolded E) does not emit any symbol and so, we insert the character “-” in the phonetic transcription at the appropriate position.

The DLOPS Algorithm

DLOPS is a recursive “divide and conquer” algorithm. It works by obtaining a transcription of a group of letters through table lookup or by approximating the transcription from smaller contained units.

DLOPS uses n-grams up to order 5 from the phonetic transcription symbols. This is done by sliding a context window and counting occurrences of symbol sequences. Next, we build a transcription model (a lookup-table) consisting of a set of letters and their possible phonetic transcriptions with corresponding probabilities. We create transcription probabilities for 1, 2, 3 and 4 letters (7.1).

$$P_{i,k} = \frac{C(\text{LS}_i, \text{PT}_{i,k})}{C(\text{LS}_i)} \quad (7.1)$$

$C(\text{LS}_i)$ Number of occurrences of letter sequence i

$C(\text{LS}_i, \text{PT}_{i,k})$ Of occurrences of letter sequence i with phonetic transcription k

$P_{i,k}$ Transcription probability of phonetic transcription k given the letter sequence i

The pseudo code for our method is:

Input:

- $w[]$ —vector containing letters of the word
- n —size of vector (number of letters)
- table —hash table containing groups of letters and phonetic transcriptions with probabilities

Output:

- $t[]$ —vector of phonetic transcriptions and their scores

```

1. DLOPS(  $w[]$  ){
2.   if ( exists( $\text{table}[w]$ ) ) then
3.     return transcriptions from  $\text{table}[w]$ ;
4.   else
5.      $\text{idx} \leftarrow \text{findPivot}(w)$ ;
6.     return
       MergeResults(
         DLOPS( $w[1 \dots \text{idx}]$ ),
         DLOPS( $w[\text{idx} \dots n]$ )
       );
7.   endif
8. }
```

The algorithm tries a dictionary lookup (line 2) and if the letter sequence has a corresponding set of phonemes it returns all possible phonetic transcriptions with their associated probabilities (line 3). If the lookup procedure fails, the algorithm searches for an optimal split position in the letter sequence (line 5). Once we have this location, we approximate the phonetic transcription of the given letter sequence using phonetic transcriptions of the two overlapped substrings. Given that the two substrings overlap on the middle character located at the juncture point given by the pivot location, we expect the candidate phonetic transcriptions for the two substrings to also overlap.

The score S of a transcription candidate, composed of two adjoined phoneme sequences S_1 and S_2 , is computed using the original transcription scores (P_1 and P_2 given letter sequences $w[1 \dots \text{idx}]$, $w[\text{idx} \dots n]$) of these phoneme sequences and a *fusion probability*. This probability is a smoothing function applied over a five symbols phoneme sequence that is composed of the last two symbols of S_1 before the fusion index and the first three symbols of S_2 (7.2).

$$S = P_1 P_2 \prod_{j=t-k}^{t+k} F_j \quad (7.2)$$

| | |
|------------|---|
| P_1, P_2 | Emission probability of phoneme sequences S_1 and S_2 |
| t | The fusion index |
| F_j | n -gram interpolation model for position j using a smoothing function |
| k | Half of the length of the fusion window ($k = 2$) |

findPivot (FP—at line 5 in the above output) function tries to maximize the transcription probability of the first ranking transcription candidate for one or both of the letter substrings. This is implemented as a back-off procedure containing three functions with different behaviors: FP1, FP2, and FP3. FP3 tries to maximize the transcription probability for both substrings. If it fails, it reverts to FP2, which is a function that tries to maximize either the left or the right substring. Again, if FP2 fails it backs off to the simplest function FP1 which selects the pivot by splitting the string in the middle.

Phonetic Transcription with Maximum Entropy

The second method used for the task of PT is based on a MaxEnt classifier. In order to do this, we need to reframe the phonetic transcription task as a label prediction process applied to every letter inside the word. Each letter is now described by an object characterized by a set of n features (corresponding to a point inside the n -dimensional features space). In our method, we use features extracted from a limited context window divided into lexical features (based on letters of the word) and phonetic features (based on previously predicted labels). After testing several feature sets, we chose the one yielding the best results in our experiments, which is given below. For a given letter L , we considered the following features:

- $l_{-i} \dots L \dots l_{+i}$, for $i = \overline{1, 2}$: features 1 and 2 in Table 7.2,
- $l_{-i} \dots L$, for $i = \overline{1, 3}$: features 3–5,
- $L \dots l_{+j}$, for $j = \overline{1, 3}$: features 6–8,
- p_{-1} : feature 9,

where l_{-i} is the previous i -th letter and l_{+i} is the next i -th letter, with respect to the position of letter l ; p_{-1} is the previous predicted phoneme, with respect to p .

We arrived to similar conclusions as those presented in Jiampojarn et al. (2008). We have tested some other features based on word length or the position of the letter in the word or whether the letter is a vowel or not, etc., but their usage did not improve the model.

There are cases when we exclude certain features (see Table 7.2). For example, we do not use $L_{-3} \dots L$ feature (the 4-gram ending with the given letter) for the first letter of a word mainly because the information it encodes is already encoded by the

Table 7.2 The features corresponding to every letter of the word *abolish*

| Letters of <i>abolish</i> | Features | Label |
|---------------------------|--|-------|
| <i>abolish</i> | 1:#ab 2:##abo 3:#a 6:ab 7:abo 8:abol 9:# | AH |
| <i>abolish</i> | 1:abo 2:#abol 4:#ab 3:ab 6:bo 7:bol 8:boli 9:AH | B |
| <i>abolish</i> | 1:bol 2:aboli 5:#abo 4:abo 3:bo 6:ol 7:oli 8:olis 9:B | AA |
| <i>abolish</i> | 1:oli 2:bolis 5:abol 4:bol 3:ol 6:li 7:lis 8:lish 9:AA | L |
| <i>abolish</i> | 1:lis 2:olish 5:boli 4:oli 3:li 6:is 7:ish 8:ish# 9:L | IS |
| <i>abolish</i> | 1:ish 2:lish# 5:olis 4:lis 3:is 6:sh 7:sh# 9:IS | SH |
| <i>abolish</i> | 1:sh# 2:ish## 5:lish 4:ish 3:sh 6:h# 9:SH | – |

l_{-1} L feature. Moreover, this feature would be completely indiscriminative in these cases because its value would be identical for all the beginning *letter* sequences.

Automatic Error Correction

Noticing systematic errors in the predictions for both the DLOPS and the MaxEnt PT methods, we added a second-layer method for correction. This task is also performed by a MaxEnt classifier. Here, we add other features beside those used by the first MaxEnt classifier. We used the already predicted labels for all the letters in a word to add supplementary (phonetic-based) features. The system is then trained to relabel all letters inside the word based on the initial prediction and the correct label (according to the training data). This is done to assure cohesion at the phonetic level, correcting certain predictions that would be unpronounceable.

Thus, when doing error correction we use the following features for a given letter L , having the phonetic transcription P :

- $l_{-i} \dots L \dots l_{+i}$, for $i = \overline{1, 2}$: features 1 and 2 in Table 7.3,
- $l_{-i} \dots L$, for $i = \overline{1, 3}$: features 3–5,
- $L \dots l_{+j}$, for $i = \overline{1, 3}$: features 6–8,
- $p_{-i} \dots P \dots p_{+i}$, for $i = \overline{1, 2}$: features 9 and 10,
- $p_{-i} \dots P$, for $i = \overline{1, 3}$: features 11–13,
- $P \dots p_{+j}$, for $i = \overline{1, 3}$: features 14–16,

where l_{-i} is the previous i -th letter and l_{+i} is the next i -th letter; p_{-i} is the previous i -th predicted phoneme and p_{+i} is the next i -th predicted phoneme (with respect to the current position) (see Table 7.3 for correction features extracted for the word *abolish*).

Since we have two methods for the first layer, the error correction classifier must be trained for each of these methods. Thus, we end up with two error correction models, each trained to correct the systematic errors of its correspondent prediction method. In the end, we end up with four ways to do phonetic transcription: DLOPS, PTC, DLOPS + ERC, and PTC + ERC.

Our tests indicate that these methods perform differently depending on the training set. BermudaPT has an automatic tenfold testing method that divides the

Table 7.3 The features for correcting phonetic transcriptions for the word *abolish*

| Letters and PT of <i>abolish</i> | Features | Label |
|--|---|-------|
| <i>abolish</i> AHBAA L ISSH- | 1:#ab 2:##abo 3:#a 6:ab 7:abo 8:abol 9:#_AH_B 10:#_#_AH_B_AA 13:#AH 14:AH_B 15:AH_B_AA 16:AH_B_AA_L | AH |
| <i>abolish</i> AHBAA L ISSH- | 1:abo 2:#abol 4:#ab 3:ab 6:bo 7:bol 8:boli 9:AH_B_AA 10:#_AH_B_AA_L 12:#AH_B 13:AH_B 14:B_AA 15:B_AA_L 16:B_AA_L_IH | B |
| <i>abolish</i> AHBAA L ISSH- | 1:bol 2:aboli 5:#abo 4:abo 3:bo 6:ol 7:oli 8:olis 9:B_AA_L 10:AH_B_AA_L_IH 11:#AH_B_AA 12:AH_B_AA 13:B_AA 14:AA_L 15:AA_L_IH 16:AA_L_IH_SH | AA |
| <i>abo</i> lishAHBAA L ISSH- | 1:oli 2:bolis 5:abol 4:bol 3:ol 6:li 7:lis 8:lish 9:AA_L_IH 10:B_AA_L_IH_SH 11:AH_B_AA_L 12:B_AA_L 13:AA_L 14:L_IH 15:L_IH_SH 16:L_IH_SH_- | L |
| <i>abol</i> ishAHBAA L ISSH- | 1:lis 2:olish 5:bolis 4:oli 3:li 6:is 7:ish 8:ish# 9:L_IH_SH 10:AA_L_IH_SH_- 11:B_AA_L_IH 12:AA_L_IH 13:L_IH 14:IH_SH 15:IH_SH_- 16:IH_SH_-# | IS |
| <i>abolish</i> AHBAA L ISSH- | 1:ish# 2:ish# 5:olis 4:lis 3:is 6:sh 7:sh# 9:IH_SH_- 10:L_IH_SH_-# 11:AA_L_IH_SH 12:L_IH_SH 13:IH_SH 14:SH_- 15:SH_-# | SH |
| <i>abolish</i> hAHBAA L ISSH- | 1:sh# 2:ish## 5:lish 4:ish 3:sh 6:h# 9:SH_-# 10:IH_SH_-# 11:L_IH_SH_- 12:IH_SH_- 13:SH_-14:-# | - |

training data into ten subsets and calculates the word accuracy rate (WAR) by mediating the results obtained by training on each tuple of nine subsets and testing on the tenth.

The CMUDICT (English) (CMU, 2011) is the most problematic dataset in literature because it contains a large number of foreign words and proper nouns. The top ranking result of testing BermudaPT on this dataset is 68% for the PTC + ERC method which is close to the results obtained by state-of-the-art methods, being surpassed only by Perceptron (70%) and MIRA (71%) (Jiampojamarn et al. 2008). For the Romanian lexicon PTC + ERC obtained an average accuracy of 93%. The only method that was tested on the same lexicon is CART (Black et al. 1998; Stan et al. 2011) and the authors reported an accuracy of about 87%.

Syllabification

Syllabification is also a core component in spoken language processing, as breaking the words into their phonological units (or “building blocks”) is an essential requirement in modern approaches to both speech synthesis and speech recognition. TTS lexicons contain syllabification information for each word (along with phonetic,

transcription, etc.), but for OOV words, the system is faced with the added problem of having to generate this information. Though all languages have some phonetic rules for syllabification, it is often the case that their etymological principles may contradict such rules, making this task surprisingly difficult. A syllable is typically composed of a “nucleus” vowel with or without surrounding consonants which are divided into the “onset” (the consonants preceding the vowel) and the “coda” (the consonants succeeding the vowel).

An interesting aspect mentioned in Ungurean et al. (2011) is that most TTS systems perform the task of phonetic transcription before syllabification because of structural reasons such as the fact that every syllable has a nucleus vowel. Performing phonetic transcription after syllabification should theoretically improve the performance of phonetic transcription. This is not the case for English, where rules for both phonetic transcription and syllabification are complex and the accuracy of syllabification is not high enough to improve the performance of phonetic transcription (Marchand and Damper 2007).

Several rule-based and data-driven methods have been proposed over the years for automatically determining syllable boundaries within words. While rule-based methods are centered on theoretical aspects of the syllabification problem, data-driven methods are generally preferable, since they are language independent and only require, as training, previously constructed lexicons of already syllabified words.

Among the algorithms designed for this task, some have to be mentioned. We use the term *juncture point* to denote the places where hyphen marks are placed within a word.

The IB1 (Daelemans et al. 1997) algorithm creates n-grams (of predetermined size) from word juncture points and stores them into a database. When a new word has to be split into syllables, every n-gram from the new word’s possible junctures is matched against the n-grams already available from the training step. n-grams are compared using a distance measure to determine how similar two n-grams are to one another.

The algorithm of Marchand and Damper (2007) is called Syllabification by Analogy (SbA) and follows the principles of the Pronunciation by Analogy (PbA) algorithm. It works by applying a “full pattern match” on the input string using entries in a dictionary compiled from the training corpora. Marchand and Damper also investigate the possibility of using syllabification to improve G2P performance on English words.

Bartlett et al. (2008) use structured SVMs to predict tags for letters in a given word and compare results obtained using different tagging strategies. Their method outperforms the results of the SbA method.

The look-up procedure (Weijters 1991) constructs a table of n-grams from the training corpus and uses it to predict hyphen marks. Each n-gram contains the “focus” character with its left and right context, including hyphen marks. When syllabification is performed on a new word, the algorithm determines if a “focus” character should be followed by a hyphen, using the majority of “similar” n-grams.

Kahn (1976) argued that the decision to insert a juncture point at a given index can be determined using three categories of consonant substrings: possible syllable-initial, possible syllable-final and universally bad syllable-initial.

Our system, *BermudaSyllabifier*, treats syllabification as a classification problem, predicting if a juncture should or should not occur after every letter of the word. As our training corpus for syllabification was significantly larger than the corpus for phonetic transcription, we decided to use only lexical features for syllabification. Our approach is similar to the above described methods in the sense that it uses the principle that some combinations of letters are more likely than others to spawn a syllable inside a word. As an effect of the context window size (we used a window of nine characters), not only patterns based on letters in single syllable participate in the decision but also supplementary relations between the current syllable and the succeeding and preceding ones are formed. This methodology yields good results and the only two methods that obtain higher accuracy (to our knowledge) are the SbA and the structured SVMs. The tests using a tenfold validation method on Webster's Pocket Dictionary (Amsler 1980) indicate that our method has an accuracy of 84% (for OOV words), 4% lower than SbA. No direct comparison was made to the second method (SVMs), as the authors used CELEX and NETTalk as their reference datasets. However, according to their experiments, Structured SVMs perform better than SbA on the task of syllabification. Nevertheless, for the Romanian language we obtained a much better accuracy of 98% for OOV words on a tenfold validation evaluation conducted on a syllabification lexicon compiled from the Romanian Academy Explanatory Dictionary (DEX 2009).

Although our lexicon for syllabification contains most common words, it does not contain any proper nouns and there are always possibilities of finding other unknown words, so we intend to extend the *BermudaSyllabifier* tool to include more methods for automatic syllabification.

Our training data for syllabification consists of a lexicon of words split into syllables. The features used for juncture prediction are letter n -grams that fit within the current context window. Once a juncture is predicted, a hyphen mark is inserted after the current letter and it becomes part of future n -grams used to predict next juncture points. We used a window size of nine letters (centered on the current letter) because in this way we surely capture the syllable that contains the current letter plus additional context information (before and after this syllable). Increasing the window size did not provide significant improvements on the accuracy of the prediction and smaller windows show a large decrease in performance.

To thoroughly define our set of features we used the following notation, where l_i refers to the current letter that we must predict a label for, l_{i+k} represent succeeding letters and l_{i-k} refers to either previous letters or previously inserted hyphen marks:

- $l_{-i} \dots L$, for $i = \overline{2, 4}$: features 1–3.
- $L \dots l_{+i}$, for $i = \overline{2, 4}$: features 4–6.
- $l_{-i} \dots L \dots l_{+i}$, for $i = \overline{1, 2}$: features 7 and 8 in Table 7.4.

Table 7.4 shows the features extracted from the Romanian word “şu-şo-tea” (eng. “to whisper”).

Table 7.4 Hyphenation features used by our classifier with the associated output

| F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | O |
|-------|---------|-----------|-------|---------|-----------|-------|-----------|---|
| #.#_ş | #.#_#_ş | #.#_#_#_ş | ş_u_ş | ş_u_ş_o | ş_u_ş_o_t | #_ş_u | #.#_ş_u_ş | F |
| #_ş_u | #.#_ş_u | #.#_#_ş_u | u_ş_o | u_ş_o_t | u_ş_o_t_e | ş_u_ş | #_ş_u_ş_o | T |
| u_ _ş | ş_u_ _ş | #_ş_u_ _ş | ş_o_t | ş_o_t_e | ş_o_t_e_a | _ş_o | u_ _ş_o_t | F |
| _ş_o | u_ _ş_o | ş_u_ _ş_o | o_t_e | o_t_e_a | o_t_e_a_# | ş_o_t | _ş_o_t_e | T |
| o_ _t | ş_o_ _t | _ş_o_ _t | t_e_a | t_e_a_# | t_e_a_#_# | _t_e | o_ _t_e_a | F |
| _t_e | o_ _t_e | ş_o_ _t_e | e_a_# | e_a_#_# | e_a_#_#_# | t_e_a | _t_e_a_# | F |
| t_e_a | _t_e_a | o_ _t_e_a | a_#_# | a_#_#_# | a_#_#_#_# | e_a_# | t_e_a_#_# | F |

Lexical Stress Assignment

In natural speech certain syllables inside a word have a higher prominence compared to the neighboring syllables of the same word. When this phenomenon occurs, it is said that the syllable is carrying lexical stress. Lexical stress prediction is critical in prosody generation for TTS systems as it governs the correct pronunciation of diverse words and it is used to discriminate between homographs. Also, high-level prosodic events manifest themselves at the level of stressed syllables.

Black et al. (1998) measured the accuracy of the lexical stress assignment on the basis of the total number of correctly classified syllables (94%) from the test-set and not in terms of WER, using their CART classifier for phonetic transcription. Webster (2004) introduces a method called Automatic Morphological Stress Prediction (AMSP) that determines the lexical stress position using information provided by automatically determined affixes. The output consists of a “stressed orthography” that is input for a decision tree phonetic transcription model. The model then outputs the final phonetic transcription of the word with marked stress position. The final stress position determined by the decision tree can be different from the initial position in the “stressed orthography.” Webster views this process as the decision tree trying to “correct errors made by AMSP.” He states that the AMSP increases the final predicted lexical stress accuracy from 81 to 87%. In 2003, Oancea and Bădulescu introduced their rule-based method for lexical stress prediction on Romanian. They trained and tested their method on the same lexicon (4,500 words) achieving a 94% accuracy. Ungurean et al. (2009) used Katz back-off for lexical stress assignment based on letter n-grams. Their algorithm works by calculating the probability of every possible combination of stress pattern on an input string. According to their evaluation, this method achieves an accuracy of over 99% for OOV words.

Although it is acknowledged that there are several degrees of lexical stress, our tool is designed to detect the location of the primary stress only, the largest perceived contrast of a syllable, compared to the other syllables in the word.

Bermuda Stress Predicting (BermudaSP) tool uses BermudaSyllabifier to split the words into syllables. A MaxEnt classifier is trained to label each syllable as carrying lexical stress (T—true) or not (F—false). When predicting the location

Table 7.5 Lexical stress prediction accuracy in terms of OOV

| Category | Word accuracy rate (%) |
|--|------------------------|
| Nouns (110,592 word forms) | 91 |
| Verbs (114,023 word forms) | 99 |
| Adjectives (111,796 word forms) | 94 |
| Adverbs (518 word forms) | 88 |
| <i>Average of the above (336,929 word forms)</i> | 95 |
| <i>Ignoring POS (336,929 word forms)</i> | 90 |

of the lexical stress inside a word, the probability for every compound syllable of fitting one of the two classes (T or F) is computed, but instead of directly using the classifier labels, the entire set of syllables is iterated and the syllable with the highest probability assigned to class T as carrying primary lexical stress is selected (only one syllable in a polysyllabic word can carry primary lexical stress).

Franzén and Horne (1997) conducted a study on stress patterns in Romanian. They showed that stress is rather influenced by derivational affixes than by inflectional ones, especially for nouns and verbs. Since the vast majority of derivational affixes change the grammatical category of a word, we were motivated to split our training data into five categories: nouns (N), verbs (V), adjectives (A), adverbs (R) and mixed (M). This is where the main difference between our approach and other methods can be seen: By splitting the training data based on the part of speech increases the overall accuracy by 5% (see Table 7.5). When predicting the primary lexical stress position for a given word, a model is chosen based on the POS tag of the given word. If the POS is different from the first four categories or if it is unknown (if there is no context and the goal is to detect stress on a word), BermudaSP uses the mixed model, which is a model created by training on the entire lexicon regardless of the POS.

The system is based on the following features:

- Current syllable.
- Next syllable.
- Second next syllable.
- Third next syllable.
- Previous syllable.
- Penultimate syllable.
- Third previous syllable.
- Number of syllables in the word.
- Distance from the end of the word in syllables.

Results for each category are shown in Table 7.5 using a tenfold evaluation methodology. Figures show that splitting the data into grammatical categories improves the overall accuracy of the system by 5%.

Foreign Words in TTS Synthesis

Beside unseen words which were not present in the training data, the OOV class also contains foreign words. Improper handling of such words has a negative impact on the TTS synthesis, as applying the same native phonetic transcription or syllabification rules on them produces undesirable results. There are two ways to handle such words, each requiring them to be firstly identified as foreign. The first strategy is to attain the above-mentioned subtasks of syllabification, phonetic transcription and lexical stress prediction using a custom set of rules adapted to the foreign language from which the word originates. However, having different sets of rule packages for languages other than the native language is considered challenging. The second strategy is to use transliteration on these foreign words and to convert them to *pseudo-native* words. This facilitates the usage of a single package of native rules/learned models for the tasks of phonetic transcription, syllabification and lexical stress.

The difference between the two proposed methods is that the first one applies phonetic transcription with syllabification and lexical stress rules for the foreign language(s) followed by an adaptation at phonetic level between the two languages while the second method uses transliteration to produce a pseudo-native word and then uses its native rule sets for attaining the final goal (thus not requiring training additional methods for syllabification and lexical stress).

Transliteration between two languages is the process in which the letters of a word in the first (source) language are transformed or mapped into letters that would correspond to a word in the second (target) language. Transliteration was initially introduced in Machine Translation for the task of converting words without a corresponding direct translation (e.g., proper names) between languages that are highly incompatible at phonetic or orthographic levels. For example, a Japanese native speaker cannot distinguish between the English sounds “L” or “R.”

Over the years, several methods for transliterating were introduced, mainly focused on languages such as Chinese, Japanese, Korean or Arab. Knight and Graehl (1997) presented a method for transliteration between Japanese and English, using finite state transducers. This method was later adapted in Stalls and Knight (1998) for bidirectional transliteration between English and Arab. Other methods for transliteration are presented in Jung et al. (2000), Meng et al. (2001), and Virga and Khudanpur (2003). In their work, Haizhou et al. (2004) classifies the above-mentioned methods as phonetic approaches to transliteration. They propose a new technique that focuses on direct orthographic mapping (DOM). Their method is also referred to as n-gram-based transliteration.

For our Romanian TTS system, we trained the BermudaPT to perform transliteration instead of phonetic transcription from English to Romanian because many foreign words found in Romanian written texts originate from English. Using tenfold validation on a subset of 40K transliterated words from the CMUDICT, we obtained an accuracy of 78%. This accuracy refers to English words that are not found in the transliteration lexicon (thus being OOV) and therefore must be automatically transliterated to Romanian. To our knowledge, there is no similar study on transliteration between English and Romanian with which to compare our results.

Detecting Which Words Require Transliteration in TTS

One common problem with both approaches to foreign word adaptation for TTS synthesis is detecting when an OOV word is a foreign word and also the source language of that word. One partial solution to this problem is to use a lookup table of word-forms for each foreign language for which the system has transliteration rules. Such a list is easier to obtain than a list of fully processed words and it can be done by crawling through documents written in specific languages. Any OOV word found by the TTS system has to be checked against these precompiled lists and once the word occurs in the lexicon of some language it can be transliterated to a native pseudo-word using a specific rule set. It is also important to keep a separate word-form list for the native language as well and to check if the word is not inside that list, as words in different languages may have identical orthographies (e.g., “merge” in Romanian means “walk” but it is also a valid English word). This list is important for determining when not to apply transliteration.

There are however cases where a word or a group of words does not appear in any lexicon, as in the case of uncommon proper nouns. Based on the fact that some orthographic symbols (especially those that have diacritics) or groups of symbols are uncommon in certain languages, the assumption that a word should be transliterated can arise from testing for such occurrences. For example, characters such as “y” or groups like “ck” are highly uncommon for Romanian.

The reasons for using the transliteration method instead of custom-designed methods and lexicons are straight forward:

- The resources involved in constructing lexicons and building methods for custom syllabification and phonetic transcription are far more challenging than just building transliteration lexicons.
- The statistical methods used for generating prosody are trained on native words of the language for which the system was designed. Using custom syllabification lexicons for foreign words, usually unseen syllables would show up and impede the overall performance.
- One might argue that applying native syllabification and lexical stress rules on pseudo-words does not generate the same pronunciations as for the word’s native language. However, this is not an understanding issue, since a nonnative speaker could pronounce such foreign words similarly, misplacing the lexical stress and making adaptations at the phonetic level.

Conclusions

This chapter presents a detailed analysis of the performance of our TTS synthesis system for Romanian compared with state-of-the-art TTS systems. We explored solutions for many of the important challenges for TTS systems, conducting all of our evaluations on OOV words, which is the true indicator of performance

when employing statistical approaches. In examining performance rates we looked at text preprocessing, phonetic transcription, syllabification, and lexical stress prediction. Within such systems, which chain a considerable number of modules (and are therefore subject to error propagation), it is important that each module is properly tuned and tested. Moreover, by determining which module performs less efficiently we were able to decide which component(s) of the system would require further research so as to improve the overall quality of the TTS system. The TTS application¹¹ described in this chapter is freely available for testing at the RACAI NLPTools Web site.¹² Our current efforts are on developing the ToBI corpus based on the Romanian Speech Synthesis (RSS) database (Stan et al. 2011).

Acknowledgements We thank Acad. Dan Tufiş for his valuable contributions and support of this work as both PhD coordinator and colleague. We also thank Dr. Adrian Zafiu for providing us with the English–Romanian transliteration corpus.

References

- Allen J, Hunnicutt MS, Klatt D (1987) From text to speech: The MITalk system. Cambridge University, Cambridge, p 216
- Amsler RA (1980) The structure of the Merriam. Webster Pocket Dictionary. Doctoral dissertation, TR-164, University of Texas, Austin
- Bartlett S, Kondrak G, Cherry C (2008) Automatic syllabification with structured SVMs for letter-to-phoneme conversion. In: Proceedings of ACL-08: HLT, pp 568–576
- Berger AL, Pietra VJD, Pietra SAD (1996) A maximum entropy approach to natural language processing. *Comput linguist* 22(1):39–71
- Bisani M, Ney H (2002) Investigations on joint-multigram models for grapheme-to-phoneme conversion. In: Proceedings ICSLP, vol 2, pp 105–108
- Black AW, Lenzo K, Pagel V (1998) Issues in building general letter to sound rules. In: The third ESCA/COCOSDA workshop (ETRW) on speech synthesis
- Brants T (2000) TnT: a statistical part-of-speech tagger. In: Proceedings of the sixth conference on applied natural language processing. Association for Computational Linguistics, pp 224–231
- Burileanu D, Sima M, Neagu A (1999) A phonetic converter for speech synthesis in Romanian. In: Proceedings of the XIVth international congress on phonetic sciences ICPH'S'99, pp 503–506
- Ceausu A (2006) Maximum entropy tiered tagging. In: Proceedings of the 11th ESSLLI student session, pp 173–179
- CMU (2011) Carnegie Mellon pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>
- Daelemans W, Van Den Bosch A, Weijters T (1997) IGTREE: using trees for compression and classification in lazy learning algorithms. *Artif Intell Rev* 11(1):407–423
- Demberg V, Schmid H, Mohler G (2007) Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. In: Annual meeting-association for computational linguistics, vol 45, no. 1, p 96, June 2007

¹¹Much of the work presented here has been done during a PhD on text-to-speech synthesis for Romanian at RACAI.

¹²<http://nlptools.racai.ro>.

- DeRose SJ (1988) Grammatical category disambiguation by statistical optimization. *Comput Linguist* 14(1):31–39
- Franzén V, Horne M (1997) Word stress in Romanian. *Lund Working Papers in Linguistics* 46:75–91
- Haizhou L, Min Z, Jian S (2004) A joint source-channel model for machine transliteration. In: *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, p 159, July 2004
- Ion R (2007) Word sense disambiguation methods applied to English and Romanian. PhD Thesis (in Romanian). Romanian Academy, Bucharest
- Jiampojarn S, Cherry C, Kondrak G (2008) Joint processing and discriminative training for letter-to-phoneme conversion. In: *Proceedings of ACL-08: HLT*, pp 905–913
- Jitcă D, Apoei V, Păduraru O (2012) Transcription of Romanian intonation-RoToBI. http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/RoToBi/RoToBi_System.html
- Jung SY, Hong S, Paek E (2000) An English to Korean transliteration model of extended Markov window. In: *Proceedings of the 18th conference on computational linguistics*, vol 1. Association for Computational Linguistics, pp 383–389
- Kahn D (1976) Syllable-based generalizations in English phonology, vol 156. Indiana University Linguistics Club, Bloomington
- Kaszcuk M, Osowski L (2009) The IVO software blizzard challenge 2009 entry: improving IVONA text-to-speech. In: *Blizzard Challenge Workshop*, Edinburgh, Scotland
- Knight K, Graehl J (1997) Machine transliteration. In: *Proceedings of the thirty-fifth annual meeting of the association for computational linguistics and eighth conference of the European chapter of the Association for Computational Linguistics*, Somerset, NJ, pp 128–135
- Lafferty J, McCallum A, Pereira FC (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data
- Marchand Y, Dampier RI (2000) A multistrategy approach to improving pronunciation by analogy. *Comput Linguist* 26(2):195–219
- Marchand Y, Dampier RI (2007) Can syllabification improve pronunciation by analogy of English? *Nat Lang Eng* 13(1):1–24
- Marques NC, Lopes GP (1996) A neural network approach to part-of-speech tagging. In: *Proceedings of the 2nd meeting for computational processing of spoken and written Portuguese*, pp 21–22
- Meng HM, Lo WK, Chen B, Tang K (2001) Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In: *Automatic speech recognition and understanding, 2001. ASRU'01. IEEE Workshop on IEEE*. pp 311–314.
- Och FJ, Ney H (2003) A systematic comparison of various statistical alignment models. *Comput Linguist* 29(1):19–51
- Pagel V, Lenzo K, Black A (1998) Letter to sound rules for accented lexicon compression. In: *Proceedings of the international conference on spoken language processing*, Sydney, Australia
- Rama T, Singh AK, Kolachina S (2009) Modeling letter-to-phoneme conversion as a phrase based statistical machine translation problem with minimum error rate training. In: *Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the Association for Computational Linguistics*, Companion Volume: Student Research Workshop and Doctoral Consortium. Association for Computational Linguistics, pp 90–95.
- Ratnaparkhi A (1996) A maximum entropy model for part-of-speech tagging. In: *Proceedings of the conference on empirical methods in natural language processing*, vol 1. pp 133–142, May 1996. Philadelphia, PA, USA
- Reynar JC, Ratnaparkhi A (1997) A maximum entropy approach to identifying sentence boundaries. In: *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, pp 16–19.
- Romanian Academy. DEX (2009) The explanatory dictionary of the Romanian language
- Samuelsson C (1993). Morphological tagging based entirely on Bayesian inference. In: *9th Nordic conference on computational linguistics*, June 1993

- Silverman K, Beckman M, Pitrelli J, Ostendorf M, Wightman C, Price P, Pierrehumbert J, Hirschberg J (1992) ToBI: a standard for labeling English prosody. In: Proceedings of ICSLP, vol 2, pp 867–870, October 1992
- Stalls BG, Knight K (1998) Translating names and technical terms in Arabic text. In: Proceedings of the COLING/ACL workshop on computational approaches to semitic languages, pp 34–41, August 1998,
- Stan A, Yamagishi J, King S, Aylett M (2011) The Romanian speech synthesis (RSS) corpus: building a high quality HMM-based speech synthesis system using a high sampling rate. *Speech Commun* 53(3):442–450
- Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech parameter generation algorithms for HMM-based speech synthesis. In: IEEE international conference acoustics, speech, and signal processing, 2000, vol 3. IEEE, pp 1315–1318
- Tuşiş D (1999) Tiered tagging and combined language models classifiers. In: Text, speech and dialogue. Springer Berlin/Heidelberg, pp 843–843
- Tuşiş D, Ceaşu A (2008) DIAC+: a professional diacritics recovering system. In: Proceedings of LREC, Marrakech, Morocco
- Tuşiş D, Ion R, Ceaşu A, Ştefănescu D (2008) RACAI's linguistic Web services. In: Proceedings of the 6th language resources and evaluation conference-LREC
- Ungurean C, Burileanu D, Popescu V, Derviş A (2011) Hybrid syllabification and letter-to-phone conversion for TTS synthesis. In: U.P.B. Sci. Bull., Series C, vol 73, Iss. 3, 2011, ISSN 1454-234x
- Ungurean C, Burileanu D, Derviş A (2009) A statistical approach to lexical stress assignment for TTS synthesis. *Int J Speech Technol* 12(2–3):63–73.
- Virga P, Khudanpur S (2003) Transliteration of proper names in cross-lingual information retrieval. In: Proceedings of the ACL 2003 workshop on multilingual and mixed-language named entity recognition, vol 15. Association for Computational Linguistics, pp 57–64, July 2003
- Weijters A (1991) A simple look-up procedure superior to NETtalk? In: Proceedings of the international conference on artificial neural networks—ICANN-91, Espoo, Finland
- Webster G (2004) Improving letter-to-pronunciation accuracy with automatic morphologically-based stress prediction. In: Proceedings of INTERSPEECH 2004, pp 2573–2576

Chapter 8

MAP: An Abstraction-Based Metaphor Analysis Program for Overcoming Cross-Modal Challenges

Sylvia Weber Russell

Abstract Much of the research on computational metaphor interpretation is based on metaphors that are prevalent in the language being processed. Moreover, the more detailed work in this area tends to focus primarily on specific domains of discourse. Such approaches, while of interest and value, are not adequate to handle novel metaphoric expressions that occur in a context unrestricted by domain, such as an ordinary conversation or a digression from a domain-specific context. This chapter describes MAP (Metaphor Analysis Program), a computer program that processes both novel and conventional cross-modal metaphor without restriction to a particular domain of discourse. MAP relies on the kind of semantic analysis that models what humans are hypothesized to do when they extend a literal meaning to a metaphoric one, especially when there is no clear indication of a discursive context. To do so, the model depends on an analysis of a metaphorically used word in its literal sense and the role of this sense in structuring the topic of the metaphor. Problems of multiple metaphoric interpretations and the uncertainty of constraints on metaphoric coherence are addressed by a number of vivid illustrations.

Introduction

Metaphor, a subject of wide interdisciplinary interest, can be defined as a figure of speech in which a word or phrase denoting one concept is substituted for or used to describe another by way of a highlighted analogy. A word or phrase used in

S.W. Russell (✉)

Department of Computer Science, University of New Hampshire

Durham, NH 03824, USA

e-mail: swr@cs.unh.edu

a metaphoric sense is thus *extended* from a more literal,¹ usual sense, presenting challenges for a natural-language program oriented only toward nonmetaphoric usages. Computational treatments of metaphor have focused mainly on conventional metaphors and/or have been restricted to metaphoric language in particular contexts (disciplines, fields, or domains). However, *novel* (unconventional, new, original) metaphors, including those that involve a broad range of contexts, also occur in various kinds of text and, while less prevalent in speech, they figure as a linguistic phenomenon that must eventually be dealt with in speech applications as well as in text. Such usages are not always found in the lexicon or interpretable through statistical processing.

The ability to interpret metaphoric usages helps to address some of the challenges particular to speech processing. For example, in speech-to-text applications, recognition of a coherent metaphoric usage can support the correct choice of a homophone. In the case of noun ambiguity, for instance, “pain” as an object of the verb “tamp down,” “squeeze out,” or “siphon off” would be metaphorically coherent and therefore preferable to “pane” as an object, since the given verbs require a nonsolid object. In the case of verb ambiguity, “ceding the view” is preferable to “seeding the view,” since given the appropriate circumstances, anything can be “ceded,” but “seeding” imposes constraints on its object. In addition to homophones, speech liaisons can be disambiguated by recognition of metaphoric usages. For instance, whether an input pair is recognized by a speech application as “break codes” vs. “break odes” may depend on the metaphoric acceptability of the former and relative incoherence of the latter.

As another application of metaphor processing in text or speech, in machine translation, a literal translation of a novel metaphor is often not in the program’s lexicon and a word-for-word translation may not work for a variety of reasons. A program must then find an interpretation which, while lacking the metaphoric nuances of the input expression, roughly conveys the intended meaning.

In electronic communication, there might seem to be little expectation of frequent metaphors. However, Tagg (2009) notes that texting often involves a desire for brevity, intensity, evaluation, and intimate/insider communication, all of which can be provided by metaphor-based expressions. Tagg has indicated the prevalence of *extended metaphor*—unconventional adaptations of common metaphors. Expressions she has identified in texting include the novel verb usage “bash out a flat plan,” the somewhat novel phrase including an oft-used metaphoric verb, i.e., “plough through (a pile of) ironing,” the novel noun-based metaphor “shower of (social) misfits,” and the extended metaphoric idiom “I have some salt, if you want to rub it in my open wound.”² Phrases of this kind occur in speech, though to a lesser extent than in text.

¹The question of what is truly literal is a subject of theoretical debate.

²The latter type, i.e., a decomposable metaphoric idiom, which can be quite original (“The cat is already peeking out of the bag”), has been treated in Russell et al. (2007).

This chapter describes a computer program that processes both novel and conventional metaphor without restriction to a particular domain of discourse. The program uses semantic analysis that models what humans are hypothesized to do when they extend a literal meaning to a metaphoric one, especially when there is no clear indication of a discursive context. The types of metaphor processed by the program are *cross-modal/cross-domain*. These metaphors extend or cross from one *conceptual domain/mode*—the domain of the *source* or *vehicle* concept—to another conceptual domain—that of the domain of the *target* concept or *topic*. The source concept is represented by the metaphorically used word(s); the target concept by a word or words signifying what is being described through the metaphor. (By contrast, *within-domain* metaphors remain within one conceptual domain, usually the physical.) Some examples of cross-modal metaphors are “to quarry ideas,” which involves extension from a physical domain (the literal domain of “quarry”) to a mental domain; Ortony’s (1979) example, “Encyclopedias are gold mines,” which involves extension from a domain of control (possession), i.e., having gold, to a mental domain, i.e., reading; and “the cadenza of his gymnastic performance,” which involves extension from a sensory domain (sound, i.e., music) to a physical domain. Cross-modal metaphors may or may not be novel; the focus in this chapter will be on cross-modal metaphor, whether novel or not.

The discussion begins with a summary of work by other researchers, followed by an indication of the role of abstraction in cross-modal metaphor. A program called MAP (Metaphor Analysis Program), which uses abstract representations to address cross-modal verb-based metaphor, is then outlined. This description is followed by a discussion of factors involved in MAP’s interpretation of verb-based and noun-based metaphor, with special attention to metaphoric coherence and some ambiguities of interpretation. The chapter concludes with a description of the paraphrase process together with sample paraphrases.

Approaches to Metaphor

Lakoff and Johnson (1980) have demonstrated the ubiquity of metaphor through an extensive list of what they call *conceptual metaphors*, such as “THE MIND IS A CONTAINER” (“Keep it in mind, Be open-minded”) and “TIME IS MONEY” (“Don’t spend your time on this”). Since through metaphor different source concepts may be used to structure the target in different ways, they recognize both “LIFE IS A JOURNEY” and “LIFE IS A GAME,” and perhaps other “life” metaphors. The reliance on conceptual metaphors focuses on conventional rather than novel metaphor, but most of the conceptual metaphors appear to be cross-modal, and this model provides a starting point for some of the following computational work.

The early approach of Carbonell (1980, 1982) is based on the stored conceptual metaphors of Lakoff and Johnson. For example, for “The gold market soared,” they match the “upward movement” of “soaring” with “UP” in the “MORE IS UP” metaphor. Since “MORE” is an “increase in value or volume,” the “MORE IS UP”

metaphor can then provide a match with “gold market,” succeeding in inferring that “the gold market is increasing in value or volume.” However, systems that rely only on stored conceptual metaphors cannot interpret linguistic metaphors that are new to the system, i.e., that do not fit any stored conceptual metaphor. For example, interpretation of “The discussion scintillated” would require something like a “DISCUSSIONS ARE STARS” or perhaps a “DISCUSSIONS ARE SPARK PRODUCERS” conceptual metaphor, which may not exist.

Hobbs (1992) addresses metaphor without recourse to stored metaphors by expressing linguistic relationships. He illustrates this with the metaphorical sentence “We insist on serving up these veto pitches that come over the plate the size of a pumpkin,” from a Democratic congressman complaining about President Ford’s vetoes (Salmons et al. 1975). Here we focus on only a part of the sentence. First, a “veto” cues relevance to a relatively complex function of Congress, described formally as a *schema*. The noun “pitches” belongs to a baseball schema, the source domain. The decomposition of the complex baseball schema into basic concepts reveals that the role of “pitch (a ball)” in the baseball schema matches the role of “send (a bill)” in the Congress schema, as the verbal form of “pitch” is defined as “send.” Thus the physical baseball domain is used to express the transfer of (mental) information. No indication of implementation is given.

Martin’s (1990) model (MIDAS) of verbal metaphor is similar in concept to Carbonell’s earlier work (above), but has some different components. Martin uses *abstraction hierarchies*, for example, to accept a new metaphor on the basis of a known (to the system) metaphor. Thus if “Computer-Process” and “Conversation” are both subordinate to “Process” in the abstraction hierarchy, then “Kill-Computer-Process” can be accepted as a metaphor on the basis of the known metaphor “Kill-Conversation.” While this approach is an advance over sole recourse to stored conceptual metaphors, the use of hierarchies has the well-known disadvantage that it introduces organization complexities, since a concept may belong to more than one, even many categories.

Carbonell and Minton (1983) specify their method for interpretation of noun-based metaphors in terms of transfer of portions of a graph consisting of concepts linked by relations. Thus for “X is a puppet of Y,” the CONTROL relation between the object “puppet” and the actor “puppeteer” is transferred to the node between X and Y. A representation system is not specified.

In contrast to the above researchers, Suwa and Motoda (1991) describe a system of representation of verb structures in terms of what they call *abstract primitives*, through which they match source and target verbs—a method which, as in the cases above, apparently succeeds in an interpretation only if such a match exists.

In the work of Barnden et al. (2003, 2004) and Agerri et al. (2007), it is acknowledged that many metaphoric usages are not adequately covered by Lakoff’s conceptual metaphors. They present *view-neutral mapping adjuncts* (VNMA’s), which “transfer those aspects that are not part of any specific metaphorical view” or conceptual metaphor (Agerri et al. 2007). VNMA’s are applied as *default rules*; they appear to specify semantic components that are generally invariant in metaphoric extension.

The metaphor theory and attendant hypotheses underlying the system of Narayanan (1999), discussed in greater detail previously (Russell 2009), have some similarities to the work of Barnden et al. and Agerri et al. Narayanan focuses on just two target domains, economic policy and politics, but implements these with fine granularity, working out specific mappings to his target domains. Entities (nominals) and actions are projected directly onto the target through pre-established conceptual metaphors in the sense of Lakoff and Johnson, such as “MOVERS ARE ACTORS” or “OBSTACLES ARE DIFFICULTIES,” which must be stored.

Cross-Modal Metaphor and Abstraction

In a metaphoric usage, there are both similarities and differences between the source and target concepts. In cross-modal metaphor, a concept signifying the similarity is abstract. In abstraction, one representation is converted to another representation in which some details of the source representation are dropped, but the underlying structure, or part of the structure, is left unaltered. As Bouchard (1995) states, “only semantic representations based on sufficiently abstract notions are able to account for language use [and I would add in particular for metaphoric usage] adequately” (p. 15).

Abstraction is sometimes thought of in terms of separation; in metaphor there is separation of elements common to source and target from details which differ. Thus for the verb-based metaphor “The news torpedoed his hope,” the relatively abstract concept of causing to not exist is common to both the literal and the metaphoric senses of the verb “torpedo” and can be applied to its object. For the noun-based metaphor “Encyclopedias are gold mines,” the relatively abstract concept of acquiring something is common to the uses of both encyclopedias and gold mines. The abstracted representation can be thought of as an interlingua between the metaphor and its literal translation.

Abstraction also enters into verbal cross-modal metaphor in the form of the object of the verb. In a linguistic metaphoric extension from a physical-domain verb, a concept which is not a physical object may be treated as an abstract object, allowing a verbal concept to act on it. The object (e.g., the noun “hope”) is abstract in the sense that it is separated from an actor or possessor of the verb “hope”—and can now be “torpedoed.” *Reification* (or *nominalization*)—treating an attribute, action, relation or situation as an abstract object in the form of a noun—is thus a first step in the creation of this kind of metaphor. Expressed syntactically, reification is an instance of the general metaphor which treats an abstract concept as a spatial object (Russell 1989). Much of metaphoric language (as well as mathematical language) (Russell 2009) can thus be viewed as sharing spatial grounding, suggesting that nonphysical verbal concepts can be analyzed in terms of spatial structures. (See also Lakoff and Nuñez (2000) on spatial ideas as a grounding for metaphor and mathematics.)

MAP: An Abstraction-Based Metaphor Analysis Program

MAP is a computer program (Russell 1986, 1992) which interprets a syntactically simple cross-modal metaphoric text in terms of a roughly equivalent paraphrase conventionally considered as literal. MAP processes sentences in which either the verb usage or (in a partially developed module) a certain type of noun usage is metaphoric. Assimilated metaphors and novel metaphors are treated in the same way, though assimilated metaphors alternatively could be defined directly in the lexicon. While many and perhaps most verbs and nouns have multiple literal senses, MAP assumes for now that the input corresponds with the one literal sense given in its lexicon.

For verb-based metaphor, the expected input format is either a 3-word predicative sentence³ consisting of a syntactic subject, a verb, and a syntactic object (4 words if the object noun is preceded by a possessive adjective) or a 2-word sentence consisting of an abstract subject followed by the verb (3 words if the subject is preceded by a possessive adjective). Verbs are expected together with any prepositions or adverbs with which they are used (e.g., “plow-through, siphon-off”), since prepositions are significant to the meaning of the verb phrase. Abstract components representing the verb in its literal sense in the lexicon are interpreted in the target domain as indicated by the noun with which the verbal concept is used. Thus for “She chased away the thought,” a mental domain is indicated by “thought,” and the components underlying “chased away” in its physical sense lead to a paraphrase including a phrase corresponding to the mental-domain phrase “voluntarily stopped thinking about.” For nominal (noun-based)⁴ metaphor, expected as a 3-word *phrase* (<noun1> is/are <noun2>), sometimes called a *category metaphor*, the components underlying stereotypical *salient predicates* (Russell 1986) of the source nominal are transferred to the target representation. Thus for “Political movements are glaciers,” the potentially extended predicates include components representing slow change.

Determining Abstract Components

The most critical aspect of the program resides in the lexicon, where verbs and nominals are represented by abstract components that are *extensible*, i.e., potentially extended from one domain to another in metaphor. These components are the basic concepts that language users are hypothesized to recognize in a literal meaning of a verb that allows them to understand a metaphoric use of that verb, even if they have never heard it before. The determination of a set of components of verbal concepts

³Articles, determiners and verb suffixes are omitted in the program.

⁴For purposes of this discussion, the words “nominal” and “noun” are used interchangeably.

thought to be extensible is thus a task that is central to the semantic analysis of metaphor. In this sense the MAP model differs from that of [Lakoff and Johnson \(1980\)](#) as described earlier, which identifies metaphor themes after they have entered into common usage.

It is important that the ontology including these components be relatively small and transparent, for purposes of evaluation and revision. An assumption that natural language and mathematics share spatial structure, e.g., structure in terms of objects and relations, suggests that a small number of abstract or primitive descriptors that overlap with those of mathematics in being spatially based reflect some intuitive consensus of speakers of the language. Structural descriptions in turn provide a framework for qualifying primitive *features*, some of which also correspond with mathematical concepts ([Russell 2009](#)), e.g., in arithmetic, calculus, or logic. Concepts of elementary physics that apply to a wide range of everyday physical situations can also be expected to play a representational role. The following structures and features either have a math-physical counterpart and/or have a broad linguistic consensus. However, there is no claim here that the components of the ontology are universal, uniquely correct, or language-free. The remainder of this subsection describes components that are abstract and therefore extensible, thus representing the potential similarities between source and target, followed by a specification of domains, which represent the differences.

Verb Structures

All verb structures are based on a STATE. The STATE may take the form of either the existence or functioning of a nominal (OBJECT BE, OBJECT FUNCTION); the expression of an attribute of a nominal (OBJECT BE <attribute>)⁵; or the identity or transition relation of a nominal to other concepts (OBJECT AT/THRU LOCATION). These abstract STATE structures can be thought of logically as one- or two-argument predicates or linguistically as unary or binary abstract *case structures*.⁶ These forms are illustrated by verbs as follows:

- OBJECT BE “exist”
- OBJECT FUNCTION “live, run, flourish”
- OBJECT BE <attribute> “glow, blossom”
- OBJECT AT LOCATION “inhabit”
- OBJECT THRU LOCATION “plow-through”

⁵Attributes, such as “red,” “asleep,” or “hopeful,” some of which have verb forms, are specific properties of the world and are not necessarily considered primitive.

⁶Abstract case structures are simpler than traditional case structures. For example, [Fillmore’s \(1968\)](#) grammatical dative and locative cases, as well as [Schank’s \(1975\)](#) *conceptual* recipient and directive cases are combined in MAP’s LOCATION.

Components modifying these STATEs represent negation, change of state (start, end)⁷ orientation toward or away from. Some illustrations are:

- NOT (OBJECT FUNCTION) “(ideas) sleep” (pace Chomsky)
- END (OBJECT BE) “(hope) disintegrate”
- ORIENT-NEG (OBJECT AT LOCATION) “lean-away-from (greed)”
- MORE (OBJECT BE (MAGNITUDE HIGH)) “(suspicion) grow”

Any STATE structure can be a result caused by an AGENT (event or animate being), as for the verb “torpedo,” which has the abstract structure AGENT NOT (OBJECT FUNCTION). In MAP, an AGENT is external to the STATE structure, i.e., it operates on a STATE not including itself. For example, in “He marketed the proposal,” “he” is the AGENT and “proposal” is the OBJECT, with the targets of the “marketing” serving as the implicit LOCATION(s). However, in “He plowed through the discussion,” “he” is an OBJECT moving to a LOCATION, and is not an external AGENT. For AGENTive verbs, which involve causation, it is the result portion of the abstract structure that receives a domain specification. Verbs requiring representation that includes other higher-level components, such as purpose, conjunction, and temporal sequence, have not yet been included in MAP’s lexicon.

MAP’s abstract structures are consistent with the criterion of Jackendoff (1983) that builds on the hypothesis of Gruber (1965) in generalizing across modalities/domains (Jackendoff’s *semantic fields*). Also, as in Jackendoff’s hypothesis, in MAP the conceptual domain of a simple sentence is determined by the domain of the nominal it includes. In fact, conceptual domains differ only in the nature of the OBJECT nominals that appear in the conceptual structures of events and states. MAP also shows an analogy with the syntax-based linguistic model of Bouchard (1995), though Bouchard’s purpose—to unite semantics with grammatical constraints—differs from that of MAP. As in Bouchard’s theory (in contrast to Jackendoff’s cases), MAP has only two case-like entities—OBJECTs and LOCATIONs—in addition to AGENT (which as an “external” AGENT, apart from the actor in a sentence, is arguably not a true case).

Verb Features

While a verb’s abstract structure is the basis of its abstract definition, it is qualifiers and connotations that are generally the point of a metaphor. These are represented in the lexicon as abstract *conceptual features* of the verbal concept. They have polarity (+, −, or sometimes both) and/or magnitude specifications, as appropriate. As qualifiers of actions, some of these features are basic spatiotemporal descriptors. Some examples are:

⁷Martin (1990) has incorporated a version of such components as an extension of his system.

- CONTINUITY (–) “lurch (to prosperity)”
- REPETITION (+) “hammer (idea)”
- SPEED (HIGH) “leap (to fame)”

VOLITION is a feature describing an actor who intends the STATE representing the verb. This feature, however, is not necessarily extended in a metaphor, though it always accompanies the presence of an AGENT. For example, while the verb “harvest” requires volition in its physical sense, it does not for the metaphoric use “He harvested anger.” However, VOLITION can be a useful feature for a recipient in a verb structure in indicating acceptance vs. passivity. Thus “He marketed democracy” implies acceptance by a potential recipient as the goal of the marketer.⁸

Simple connotations are represented as FORCE (which may be HIGH or LOW) and EVALUATION (POSITIVE or NEGATIVE, which may further have HIGH or LOW qualifiers for extreme values). EVALUATION and FORCE correspond to Osgood’s (1980) evaluation and potency factors—two of the three nonstructural factors (the other being activity, implicit in the motion-oriented features above) he empirically determined to be extended in metaphoric usage (see also Aarts and Calbert (1979)). FORCE may express the effect on an actor or the effect on the recipient or affected object, an effect which may extend more or less subtly to the reader or hearer. Illustrated FORCE types are:

- (FORCE-on-actor LOW) “(he) sail-through (discussion)”
- (FORCE-on-recipient HIGH) “(group) guillotine (proposal)”

Both verbs and nouns may have EVALUATION feature values assigned to them, though most are neutral:

- (EVALUATION POSITIVE)
 - nouns: “peace,” “privilege”
 - verbs: “hope,” “decorate”
- (EVALUATION NEGATIVE)
 - verbs: “torpedo,” “usurp”
 - nouns: “injury,” “expense”

Connotative components are missing from most of the research discussed in section “Approaches to Metaphor,” except for that of Narayanan (1999), who uses some of these factors, such as EVALUATION, as *parameters* along with his extensible spatiotemporal structures.

Various emotions are also in the design of MAP, though not yet implemented. Emotional states as real-world concepts are not abstract in the sense that MAGNITUDE or REPETITION is. However, they are clearly extensible, though with

⁸Acceptance thus distinguishes belief from mere reception, corresponding to Schank’s (1975) use of the psychological terms *long term memory* and *conscious processor*.

varying intensity (the fear experienced when one's hope is torpedoed may not have the same intensity as that when one's boat is literally torpedoed).

Conceptual Domains

As indicated earlier, metaphoric extensions in MAP are represented as operating between conceptual domains or modes, which can be thought of as corresponding to human faculties; domains are thus orthogonal to the extensible portion of the ontology. Conceptual domains apply to both verbs and nominals. Domains and subdomains (with examples of nominals) are:

- MENTAL

INTELLECT ("idea")

ATTITUDE ("interest," "fear")

WILL ("intention")

- SENSORY

SIGHT ("view")

HEARING ("sound")

TASTE ("flavor")

SMELL ("odor")

FEELING ("pain")

- CONTROL

INTRINSIC ("talent")

EXTRINSIC

(control of) PHYSICAL concepts ("house")

WEALTH ("stocks")

ACTs deriving from authority ("rights, duties")

- PHYSICAL

ANIMATE

INANIMATE

Added for nominals, though not yet implemented:

- TIME ("year")

- SPACE ("room")

This taxonomy within the ontology is obviously breadth- rather than depth-oriented.

In MAP it is the inconsistency between the conceptual domains of a verb and the OBJECT of the structure that triggers potential cross-modal metaphor recognition

as opposed to literal meaning.⁹ To this end, every verb and nominal in the lexicon is assigned the conceptual domain in which it is thought to be literal. MAP accepts metaphors in which a verb in any conceptual domain is a source concept of the metaphor, though the source is most often PHYSICAL.

Since the conceptual domains represent different faculties, it will come as no surprise that the nature of the domain may have implications for the determination of metaphoric vs. literal phrases and for structure-related inferences. For example, SENSORY-domain verbs represent connections between a bodily mental sense and the physical world. Thus it is not only phrases in which SENSORY verbs have SENSORY objects (e.g., “He heard a sound”) that are literally consistent; rather, a SENSORY verb with a PHYSICAL object might *also* be considered a literal usage, in spite of the domain difference between the verb and the object (“He heard a violin,” “He heard drops of rain”). This apparent deviation can be considered a case of ellipsis of the implicitly understood word “sound (of)”; such cases are discussed further with respect to constraints in section “Coherence Complications”. This point, however, is of only theoretical interest; while MAP currently tests for literal usages in all domains, in an operational context literal and assimilated usages would likely not be subject to a metaphor analysis procedure.

Implications of particular domains concerning inferences can be illustrated by an example occasionally cited in the literature. In the MENTAL-INTELLECT domain, communication between people can be expressed metaphorically (according to the *conduit metaphor*, as in “He tossed her an idea”). However, while the expression implies that the target of the transfer has the concept (represented as a resulting state in MAP’s interpretation), there is no implication that the concept leaves the source, as it would in a literal, PHYSICAL domain. That is, the communicated concepts are (metaphorically) copied rather than literally transferred, as would be implied by the literal sense of “toss.”

The small size and transparent organization of the set of extensible and nonextensible components allow the management of the ontology and the task of representation in terms of that ontology to be feasible. Also, by defining words through the abstract components, we can note which components, when imposed on the target domain, positively or negatively affect the quality of the interpretation of the metaphors.

Interpretation

In its interpretations, MAP does not compare a source representation with a target, as its process does not assume an existing similarity. Rather, the abstract source representation is imposed, i.e., directly projected onto the target, changing its

⁹The dependence of metaphor recognition on nonliteralness does not necessarily imply that literal meanings are always accessed before metaphoric ones by humans.

representation. This process is similar to the one (Indurkha 1992) posits for metaphors that are *similarity-creating*, which apply to some metaphors that are within the same conceptual domain¹⁰ and arguably to all cross-modal metaphors, since the various modalities by definition differ conceptually from one another. As Indurkha states,

In instantiating the source concept network in the target realm, parts of the realm are “grouped” together and made to correspond to the concepts of the concept network. In this process, the target realm is given a new ontology, and its structure, as seen from the more abstract concept network layer, is changed. (p. 254)

However, source representations cannot be imposed arbitrarily. This section discusses various kinds of ambiguities and their possible resolutions, and is followed by examples of paraphrases.

Determination of Coherence

Coherence constraints on interpretations assure (as far as possible) that the expression makes sense metaphorically, i.e., would not be considered semantically anomalous by most people. Of course, with a little effort and imagination, an interpretation can perhaps be found for any potential metaphor. However, a phrase which fails to satisfy certain constraints may indicate a mis-parse by the natural language processing program hosting the metaphor program. An extreme example of such a text is given by garden path sentences¹¹ including extended noun compounds that are syntactically ambiguous, such as “He disassembled the computer privilege revocation prevented him from using.” Here the two possibilities of the intended meaning are “disassembled the computer” and “disassembled the privilege.” The verb “disassemble” constrains its object to be complex, i.e., to have parts. As “privilege” would not be defined as having parts, it is excluded as an object in favor of “computer.” Another type of ambiguity is given by conjunctions, as in “Daniel heard Erika break into the discussion and the decision of the participants.” Here the conjunction “and” joins the objects of “heard” rather than potential objects of “break into,” because a decision (as a simple action rather than as a written or spoken judicial decision) does not contain anything and therefore cannot be broken into.

¹⁰Indurkha cites parts of Stephen Spender’s poem, *Seascapes*, that image a swath of flowers as a “downhill rush [of water].”

¹¹A garden path sentence is one for which understanding does not proceed in a straight line.

Types of Constraints on Nominals

To allow such determinations of coherence in MAP, the lexicon includes some abstract constraints that the OBJECT of the verb structure must satisfy, analogous to *selectional restrictions* on nonmetaphoric language (e.g., restrictions on what kind of nouns can be direct objects of a particular verb). MAP recognizes two types of descriptors of nominals. As constraints imposed by verbs on their objects, they can be combined in various ways through conjunction and disjunction. The first type of descriptor consists of conceptual features (cf. features of this type for verbs in section “Verb Features”). For cross-modal metaphor interpretation, these features are fewer in number than literal semantic features of nominals, since many details of the nominal concept drop out of the picture through abstraction. For example, PART (of), CONTAINED (in), and FIXED (to) features that posited for a nominal which is a spatial object merge in nonspatial language, since certain topographical features of spatial objects do not apply. MAP’s feature set and its application are discussed in detail in [Russell \(1992\)](#). Current features for nominals, understood in an abstract sense, with values “+,” “–”, or “variable” are:

- SHAPE (discrete vs. amorphous)
- 1-DIMENSIONAL or CONTINUOUS (linear-like)
- PART
- CONTAIN (can contain)
- COMPLEX (vs. elementary)
- FLUID
- PERMANENT (normally inherent)
- ANIMATE (dynamic).

In terms of feature values, “privilege” in the garden path example given earlier is not the object of “disassemble,” because it is –COMPLEX, while “disassemble” requires a +COMPLEX object. In the second example, “decision” is not the object of “break into,” because it is –CONTAIN, while “break into” requires a +CONTAIN object.

The second type of nominal descriptor is given by the nominal’s *conceptual category* in the sense of [Schank \(1975\)](#). For example, “eviction” is conceptually a reification of a conceptual act; “health” is a reification of a conceptual attribute. Along with features, the conceptual category (act or attribute) of the object may serve as a constraining criterion; “He plowed-through health” would be considered incoherent, since constraints imposed by “plow-through” preclude the object from being a conceptual attribute such as “health.”

Coherence Complications

Inevitably the process of specifying constraints on objects for coherence determinations leads to recurrent questions regarding where to draw the line. That the determination of constraints and the strictness of their application may not be clear

is not surprising, since humans also often differ on whether a putative metaphor makes sense or not. Further ambiguity is presented by the distinction between conceptual and pragmatic (pertaining to world knowledge) criteria. The feature PERMANENT demonstrates the fuzziness of the line between these two types of assessment. For example, the MENTAL-INTELLECT attribute “intelligence” and the CONTROL-INTRINSIC attribute “talent” are thought of as +PERMANENT, while emotions are –PERMANENT. It is possible, however, that intelligence could lose its “permanence” through accident or be created in a lab or (metaphorically) in a robot. While the +PERMANENT feature aids in dubbing as (probably) incoherent those phrases based on verbs representing the start of a state of intelligence, such as “He planted intelligence” (vs., e.g., “He planted hope”), constraints to allow coherence in exceptional cases are needed. Thus pragmatic considerations tend to creep in when the abstract nature of the above features precludes fine-grained judgments of metaphoric coherence. In an operational natural-language system, the acceptability of “plant intelligence” could be supported, if, e.g., followed by “into the computer program.”

The determination of constraints imposed by a verb is especially complicated by the possibility of ellipsis, or the omission of a word necessary for coherence with the verb but inferred by language-users by virtue of the nominal which is present. The assumption by MAP of ellipsis vs. no ellipsis would lead to different assessments of whether the phrase is a coherent cross-modal metaphor or, rather, an incoherent metaphor or a literal expression.

For example, the phrase “He marketed the idea” is a coherent cross-modal metaphor if “idea” is conceptually the OBJECT of “market,” but is a literal phrase if the intended OBJECT is, e.g., “rights (to the idea),” which could be bought with money, the CONTROL-of-WEALTH domain of “market.” Different dual assessments pertain to the phrase “He sold the war.” This phrase is incoherent if “war” is the conceptual OBJECT of “sell,” but a coherent cross-modal metaphor if the elided OBJECT “idea (of war)” or “plan (for war)” is the true OBJECT, i.e., if “He sold the (idea of, plan for) war” is intended.

Many cases of ellipsis (such as “sell war”) are so common as to be easily understood by language-users without explicit awareness of the ellipsis; a natural language processing program needs no special (metaphor) processing for an interpretation. In any case, any phrase which is passed to MAP is generally treated “as is;” any judgments involving ellipses is left to the host program. A sole exception in the current implementation concerns sentences in which the object is an act or event that has a physical location (e.g., “discussion, concert”). If a PHYSICAL-domain verb in such a sentence expresses going to or from, the elided location is assumed as the OBJECT and the phrase is not considered to be a cross-modal metaphor.

Multiple Interpretations

Beside excluding interpretations on the basis of constraints, a metaphor program must sometimes deal with more than one coherent metaphoric meaning proceeding

from one literal verb sense, indicating different interpretations. For example, “anger” differs from many other emotions in that it is directed at someone or something. Thus a phrase indicating transfer of anger, such as “His anger made-the-rounds,” may be interpreted as directed toward different objects of the anger rather than as adopted by different persons. In MAP, attitudes that can be directed outwards are indicated as such in the lexicon. Thus TARGET is a feature for “anger,” enabling the relative probability that “His anger made-the-rounds” means that he directed his anger to successive targets, rather than that successive people had his anger. The discursive context would play a determining role in the selection of an interpretation here.

As another example, the verb “swallow” with the object “idea” would be paraphrased as accepting the idea and therefore having the idea, while the same verb with the object “pride” would result in a paraphrase with an opposite meaning, i.e., eliminating the pride and therefore *not* having the pride. In this case the choice of meaning can be guided by the fact that “pride” is an attribute, which is something already belonging to the possessor, making acquisition redundant, while “idea,” being alienable, may not be.

A more difficult case is presented by the verb “squeeze out” and other verbs relating to both a source, e.g., object of the preposition “from,” and an object of the verb. While the abstract structure of this verb in its literal sense minimally means to move something out of something else, there are at least two metaphoric meanings, depending on whether the intention is to rid someone or something of the object or to bring the object forth from someone or something, thereby making it available to other animate beings. The provisional procedure of MAP is that if the object is an attribute (e.g., “He squeezed-out anger”), the result is that the object is gone, while if the object is not an attribute (e.g., “He squeezed-out the idea”), the result is that the object is brought forth. Articles also help to disambiguate a verb: While “squeeze-out” the “agreement” may lead to the interpretation “to cause an existing agreement to end,” “squeeze-out an agreement” probably means “to cause the existence of the agreement.” Other cases are even less clear, such as “squeeze-out a view” (a little weird, but the sense could either be as in “clouds squeezed out a view of the sun” or as in “he squeezed out a view between the two attendees in front of him”). In such cases, MAP will not always get it right. A possibility is to pass multiple metaphoric interpretations to the host program.

Disambiguation and Possessives

The presence of a possessive pronominal adjective, such as “her,” can assist in disambiguation of phrases that potentially have multiple interpretations, such as “he squeezed-out her ambition.” Here “squeeze out” can be treated similarly to “swallow,” in that an attribute (“ambition”) is possessed inalienably (by “her”) and therefore receives the interpretation of elimination rather than extraction. A possessive adjective may also have a role in distinguishing two different though related senses of the verb, because the possessive form indicates that an animate being

other than the actor is involved. For example, “break-into her discussion/anger” (meaning “interrupt her discussion/anger”) is quite different from “break-into discussion/anger,” meaning “start to discuss/be angry.”

Possessives do not always disambiguate verb structures, however, especially when the object nominal is a conceptual act, such as the nominal “defense.” For “her defense,” for example, “defense” can be either the “defense” that “she” makes or the “defense” made of “her.” While MAP does not have the role of making this choice, which is left to the host program, it is able to produce either interpretation. For the latter sense, MAP uses the verb form of “defense” in its lexicon definition to paraphrase “he leaned-away-from her defense” in terms of “not defending her.”

In a final example, the presence of a possessive adjective can help to specify what is unspecified in the input phrase. For “he plowed through the research-proposal” it is unknown whether the relationship between the subject and the proposal is passive (reading) or active (writing). In such cases the interpretation settles on one of a few relational abstract (i.e., very general) verbs, depending on the verb structure and the conceptual category and domain of the OBJECT. Here the output verb is INTERACT-WITH. A possessive adjective preceding the object that is distinct from the subject, however, as in “he plowed through her research-proposal,” can disambiguate the interpretation, selecting the general, passive meaning ATTEND-TO.

Paraphrase

The output of MAP for each input sentence is a pseudo-English paraphrase consisting of lexicalizations of the verb components described earlier, along with pronouns and nouns (or syntactic variations thereof) that occur in the input. Articles, determiners, and grammatical suffixes are omitted.¹²

Verbal Metaphor

To produce a paraphrase of verbal metaphor, the program maps to English words all abstract components identified in the analysis of the verb and integrates the object. For example, the abstract definition for the verb “torpedo” includes the components

Action: (CONTINUOUS—) (SPEED HIGH)

Effect: (FORCE-ON-RECIPIENT HIGH)

These are mapped to “suddenly,” “quickly,” and “with intense effect” respectively.

EVALUATION components are POSITIVE or NEGATIVE, as described in section “Verb Features”. The determination of an EVALUATIVE connotation for

¹²Alternatively, components could be mapped to another primitive-component-based system, such as Schank’s (1975).

the phrase as a whole requires some simple combination logic, since the concept underlying either the verb or the object or both may be described as POSITIVE or NEGATIVE. Thus if both the verb and OBJECT noun have a POSITIVE value, as in “Her hope blossomed,” or both have NEGATIVE value, as in “He torpedoed her anger,” the combination is mapped to the phrase “with positive connotation” in the interpretation. If the verb is NEGATIVE and the OBJECT noun POSITIVE, as in “The news torpedoed his hope,” the lexicalization “with negative connotation” is included. A case in which the verb is POSITIVE and the OBJECT noun is NEGATIVE, as in “He embroidered his anger” or “Her anger blossomed” might be considered to convey malice or irony. This topic is of interest in itself, but such sentences are not currently assigned a connotation by MAP.

An alternative kind of paraphrase implicit in the model is translation into a specific verb.¹³ If there is a verb in the target domain that has an abstract representation in common with the source concept or part of the source (at least the structure), and nonstructural components do not conflict, then that verb can be included in the paraphrase. For the example “The news torpedoed his hope,” that verb could be “disappoint,” which has the same structure as the verb “torpedo,” i.e., AGENT (or event) cause NOT STATE: (OBJECT BE), but is in the MENTAL-ATTITUDE domain, the domain of “hope,” rather than in the PHYSICAL domain of “torpedo.” Remaining components (here, this would be HIGH FORCE, HIGH SPEED, NEGATIVE EVALUATION) would be lexicalized directly.

Two possible paraphrases of “The news torpedoed his hope” are then:

from structure:

news cause he stop hope
news disappoint him

from character of the action:

with intense effect, quickly, with negative connotation

Abstraction necessarily entails a loss of information, and the paraphrases produced are general. They are close in content to the types of responses (Gentner and France 1988) observed empirically, which they classify as *minimal subtraction* (of meaning from the verbal concept in its original sense).

Nominal Metaphor

The paraphrase process for nominal metaphor, implemented in a pilot program, is somewhat more complex and open than that for verbs, since nominals do not have any inherent structure. In nominal metaphor, the components that are extended to the target nominal are salient (Winston 1978; Russell 1986), often stereotypical

¹³This method could theoretically be used to produce another *metaphoric* verb or (with dubious accuracy) a verb in a foreign language.

predicates defined for the source nominal. Such predicates are represented in terms of the described verb components. A salient property of a nominal that often enters into metaphoric interpretation is its **FUNCTION** or **TYPICAL ACTION**, for an artifact or a natural concept, respectively. For example, a **FUNCTION** of a knife is to cut, and a **TYPICAL ACTION** of a glacier is to change slowly.

As nominal metaphor involves the extension of verbal and attributive properties, a brief description of nominal metaphor interpretation will illustrate representational aspects of verbal metaphor as well. For the example “Encyclopedias are gold mines,” the abstracted **FUNCTION** of “gold mine” (one takes gold out of it) is defined with the resultant **STATE** structure and connotation:

STATE: (OBJECT:CONTROL-EXT (of gold) AT LOCATION:<user>) (EVALUATION (POSITIVE HIGH))

The conceptual domain of this **FUNCTION** is **CONTROL-EXTRINSIC** (i.e., possession) of a **PHYSICAL OBJECT**, i.e., “gold.” The abstract structure of the **FUNCTION** predicate and the **EVALUATION** are integrated into the **FUNCTION** predicate of “encyclopedia” (one reads it, i.e., takes information from it), giving the resultant **STATE** structure and connotation in the target domain:

STATE: (OBJECT:MENTAL-INTELLECT AT LOCATION:<user>) (EVALUATION (POSITIVE HIGH))

The **OBJECT** is specified here only by its conceptual domain, **MENTAL-INTELLECT**—the domain of the target resultant **STATE** itself. However, it can be lexicalized more or less reliably as a generic intellectual object, such as “information.” The paraphrase is then “One read encyclopedia has result one has information which-has high positive value.”

While the **EVALUATION** is imposed, and while there is no literal similarity between reading and mining, the underlying abstract structure of the **FUNCTION** of “gold mine” is similar to that of “encyclopedia.” This is not the case for a metaphor such as “Dumps are gold mines.” Here the entire **FUNCTION** structure of “dump” (to put things *into* it rather than literally or metaphorically take them out) is overridden; the interpretation is that something of high value can be found in dumps.

The property of **HIGH POSITIVE** value of “gold mine,” which is salient in both metaphors, is culturally based. Like other connotations of nominals, it is the kind of information that must be included in representations of nominals if metaphors are to be interpreted computationally. Nominal metaphor interpretations are considered to be only likely, not definitive, since the writer/speaker may have wished to highlight some less obvious aspect of the source concept. However, metaphors that have more obscure interpretations usually require further elaboration, requiring multi-sentence analysis.

Verbal Metaphor Examples

Following are sample paraphrases of verbal metaphor by MAP, showing its ability to produce minimal interpretations and revealing what refinements (apart from syntax) need to be made.

sentence: he plow-through elizas research-proposal

interpretation: HE CONTINUOUSLY WITH-EFFORT INTERACT-WITH ELIZAS RESEARCH-PROPOSAL

sentence: he torpedo elizas proposal

interpretation: HE SUDDENLY CAUSE NOT ELIZAS RESEARCH-PROPOSAL FUNCTION, WITH INTENSE EFFECT, WITH NEGATIVE CONNOTATION

sentence: eliza decorate idea

interpretation: ELIZA CAUSE START IDEA BE MORE-COMPLEX, WITH POSITIVE CONNOTATION

sentence: country prance-to prosperity

interpretation: COUNTRY EASILY START HAVE PROSPERITY, WITH POSITIVE CONNOTATION

sentence: his ambition blossom

interpretation: HIS AMBITION START BE GREATER, WITH POSITIVE CONNOTATION

sentence: his ambition bloat

interpretation: HIS AMBITION START BE GREATER, WITH NEGATIVE CONNOTATION

sentence: building corrode view

interpretation: BUILDING CAUSE NOT VIEW BE POSITIVE, WITH NEGATIVE CONNOTATION

sentence: democracy sleep

interpretation: DEMOCRACY NOT FUNCTION

sentence: he lean-toward anger

interpretation: HE BE-ORIENTED-TO HAVE ANGER

sentence: he market democracy

interpretation: HE TRY-TO CAUSE DEMOCRACY TO-GO TO ACCEPTING UNKNOWN

sentence: he lean-away-from her defense

interpretation: HE BE-ORIENTED-TO NOT DEFEND HER

sentence: he squeeze-out her view

interpretation: HE WITH-EFFORT CAUSE SHE NOT HAVE HER VIEW

These paraphrases are not detailed, nor can it be expected that metaphors (especially rich ones) can be exactly translated, though humans may understand them, since analysis by itself can only describe the metaphor, not convey the experience of it. However, the fact that the paraphrases, while minimal, have basic content suggests that even if the ontology is significantly expanded, it will remain relatively small and transparent.

Conclusion

The MAP program produces nonmetaphoric paraphrases of cross-modal metaphoric usages of verbs that it recognizes as coherent. A pilot program paraphrases nominal metaphor in terms of salient components of the metaphorically used nominal. The ontology for verb components accounts for both similarities (through abstract, extensible components) and differences (between conceptual domains) underlying cross-modal metaphor. Extensible components include not only abstract structures but also connotations and stereotypical experience, imposition of which restructures concepts in the target domain. As it is based on the semantics of verbs as used literally, this method has both explanatory and cognitive relevance. Experience with MAP has revealed that some of the interpretation difficulties, including coherence judgments, ellipsis, and multiple meanings, are due to linguistic factors other than metaphor. These and other complexities of computational metaphor processing and representation allow ample opportunity for future research.

References

- Aarts J, Calbert J (1979) Metaphor and non-metaphor: the semantics of adjective-noun combinations. Max Niemayer, Tübingen
- Agerri R, Barnden J, Lee M, Wallington A (2007) Invariant mappings and contexts in a computational approach to metaphor interpretation. In: IJCAI-MRCS
- Barnden J, Glasbey S, Lee M, Wallington AM (2003) Domain-transcending mappings in a system for metaphorical reasoning. In: EACL, pp 57–61
- Barnden J, Glasbey S, Lee M, Wallington AM (2004) Varieties and directions of inter-domain influence in metaphor. *Metaphor and Symbol* 19:1–30
- Bouchard D (1995) The semantics of syntax: a minimalist approach to grammar. University of Chicago Press, Chicago, IL
- Carbonell J (1980) Metaphor: A key to extensible semantic analysis. In: ACL, pp 17–21

- Carbonell J (1982) Metaphor: an inescapable phenomenon in natural-language comprehension. In: Lehnert W, Ringle M (eds) *Strategies for natural language processing*. Erlbaum, Hillsdale, NJ, pp 415–434
- Carbonell J, Minton S (1983) Metaphor and common-sense reasoning, Rep. No. CMU-CS-83-110. Carnegie-Mellon University, Pittsburgh, PA
- Fillmore C (1968) The case for case. In: Bach E, Harms R (eds) *Universals in linguistic theory*. Holt, Rinehart and Winston, New York, pp 1–88
- Gentner D, France I (1988) The verb mutability effect: studies of the combinatorial semantics of nouns and verbs. In: Small S, Cottrell G, Tanenhaus M (eds) *Lexical ambiguity resolution*. Morgan Kaufmann, San Mateo, CA
- Gruber J (1965) *Studies in Lexical Relations*. Doctoral Dissertation, MIT, Cambridge, MA. Indiana University Linguistics Club, Bloomington, IN
- Hobbs J (1992) Metaphor and abduction. In: Ortony A, Slack J, Stock O (eds) *Communication from an artificial intelligence perspective: theoretical and applied issues*, pp 35–58. Springer, Berlin
- Indurkha B (1992) *Metaphor and cognition*. Kluwer, Dordrecht
- Jackendoff R (1983) *Semantics and cognition*. MIT Press, Cambridge, MA
- Lakoff G, Johnson M (1980) *Metaphors we live by*. Chicago University Press, Chicago
- Lakoff G, Nuñez R (2000) *Where does mathematics come from? How the embodied mind brings mathematics into being*. Basic Books, New York, NY
- Martin J (1990) *A computational model of metaphor interpretation*. Academic Press, New York
- Narayanan S (1999) *Moving right along: a computational model of metaphoric reasoning about events*. In: AAAI
- Ortony A (1979) Similarity in similes and metaphors. In: Ortony A (ed) *Metaphor and thought*. Cambridge University Press, New York, NY
- Osgood, C.: The cognitive dynamics of synesthesia and metaphor. In: Honeck R, Hoffman R (eds) *Cognition and figurative language*. Erlbaum, Hillsdale, NJ, pp 203–238
- Russell SW (1986) Information and experience in metaphor: a perspective from computer analysis. *Metaphor Symbolic Activity* 1:227–270
- Russell SW (1989) Verbal concepts as abstract structures: the most basic conceptual metaphor? *Metaphor and Symbolic Activity* 4:55–60
- Russell SW (1992) Metaphoric coherence: distinguishing verbal metaphor from anomaly. *Comput Intell* 8:553–574
- Russell SW (2009) Abstraction as a basis for the computational interpretation of creative cross-modal metaphor. *Int J Speech Tech* 11:125–134
- Russell SW, Fischer I, Dormeyer R (2007) The cat and the brocaded bag. In: *NLPCS*, pp. 27–37
- Salmans S, DeFrank T, Buresh B, Hubbard H (1975) A ford in high gear. *Newsweek* 20:13
- Schank R (1975) *Conceptual information processing*. North Holland, Amsterdam
- Suwa M, Motoda H (1991) Learning metaphorical relationships between concepts based on semantic representation using abstract primitives. In: *IJCAI-CANL*, pp. 123–131
- Tagg C (2009) *A corpus analysis of sms text messaging*. Ph.D. thesis, University of Birmingham
- Winston P (1978) Learning by creating and justifying transfer frames. *Artif Intell* 10:147–172

Chapter 9

Translation of Idiomatic Expressions Across Different Languages: A Study of the Effectiveness of TRANSSEARCH

Stéphane Huet and Philippe Langlais

Abstract This chapter presents a case study relating how a user of TRANSSEARCH, a translation spotter as well as a bilingual concordancer available over the Web, can use the tool for finding translations of idiomatic expressions. We show that by paying close attention to the queries made to the system, TRANSSEARCH can effectively identify a fair number of idiomatic expressions and their translations. For indicative purposes, we compare the translations identified by our application to those returned by GOOGLE TRANSLATE and conduct a survey of recent Computer-Assisted Translation tools with similar functionalities to TRANSSEARCH.

Introduction

Idioms are commonly defined as expressions of a given language, whose sense is not predictable from the meanings and arrangement of their elements (McArthur 1992). For example, an expression like “*to be hand in glove*” meaning “*to have an extremely close relationship*” cannot have easily been deduced from what a hand and a glove are. Idioms—and more generally Multi-Word Expressions (MWEs)—pose significant problems for many applications of natural language processing since they are numerous in most languages and have idiosyncratic meanings that severely disturb deep analysis (Sag et al. 2002). The problem of MWEs—and idioms in particular—is especially acute in the case of Machine Translation (MT) where a failure of the system to detect such expressions often leads to unnatural, if not comical outputs.

S. Huet (✉)

LIA-CERI—Université d’Avignon, Avignon, France

e-mail: stephane.huet@univ-avignon.fr

P. Langlais

DIRO—Université de Montréal, Montréal, Québec, Canada

e-mail: felipe@iro.umontreal.ca

Therefore, one important component of an MT system is its lexicon of MWEs. This is true for rule-based MT systems as well as statistical MT (SMT) ones. Currently, state-of-the-art phrase-based SMT systems rely on models (pairs of phrases) that do not handle MWE specifically. Some authors have been trying to group multi-word expressions before the alignment process (Lambert and Banchs 2005) or to add a new feature encoding the knowledge that a given phrase pair is a MWE (Ren et al. 2009; Carpuat and Diab 2010). These two last works showed that MT could be improved with MWEs extracted automatically, or defined manually from WORDNET.

Not only are idioms interesting for improving MT systems, they are well known to pose problems to nonnative speakers. This is especially true when a second-language idiom is much different from its translation into the native language. For instance, French speakers might easily catch the English idiom “*play cat and mouse*” because its French translation “*jouer au chat et à la souris*” is literal in this case. On the contrary, they could find hard to understand “*He couldn’t say boo to a goose*”¹ because its translation into French “*Il est d’une timidité malade*” (literally “*He is sickly shy*”) is completely different.

Idiomatic expressions are interesting for professional translators as well. In Macklovitch et al. (2008), the authors analyzed the most frequent queries submitted by users to the bilingual concordancer TRANSSEARCH. They found that among others things, users frequently queried idiomatic phrasal verb expressions, such as “*looking forward to*.” Because they were expecting that the users would query idiomatic expressions, they did not investigate this aspect of the logfile any further, but concentrated instead on analyzing the prepositional phrases (some of which were idiomatic) frequently submitted to the system.

In this chapter, we study the problem of translating idiomatic expressions from a user perspective. We attempted to identify the translations of a number of idioms in the Translation Memory (TM) of the new version of the bilingual concordancer TRANSSEARCH. Since many idioms have inflected forms, we show the impact of different strategies for querying the database. For instance, in the (idiomatic) expression “*to keep to oneself*,” both the verb “*keep*” and the pronoun “*oneself*” can vary according to conjugation and inflection, respectively, and verbatim queries may fail to identify relevant occurrences of the expression.

The remainder of the chapter is organized as follows. Section “Idiomatic Expressions and CAT Tools” presents the variability of idiomatic expressions and the interest of Computer-Assisted Translation (CAT) tools for users to translate them. Section “TRANSSEARCH” describes TRANSSEARCH, the Web application we employed in our experiments. Section “Methodology” provides information about the data we used and the query submission process to the TM system to find translations. Section “Evaluation” is dedicated to the evaluation of the translations proposed by the system, including the comparison of TRANSSEARCH with GOOGLE

¹At the time of writing, GOOGLE TRANSLATE produces the literal translation “*Il ne pouvait pas dire boo à une oie*.”

TRANSLATE. Section “Survey of CAT Tools Comparable to TRANSSEARCH” conducts a survey of recent CAT tools with similar functionalities to TRANSSEARCH. Section “Conclusion” provides a conclusion.

Idiomatic Expressions and CAT Tools

What Is an Idiomatic Expression?

It is difficult to find a universal definition that covers the variety of what can constitute idiomatic expressions, examples of which are “*give up*” and “*his ears must be burning*.” As mentioned at the beginning of this chapter, they are often defined as sequences of words involving some degrees of semantic idiosyncrasy or non-compositionality.

In phraseology, idiomatic expressions—also named phraseological expressions or phrasemes—are defined as non-free multi-word expressions, which means that at least one of their components is selectively constrained or restricted by linguistic convention such that it is not chosen freely (Mel’čuk 1995, 2010). For example, the expression “*be in the same boat*,” meaning “*have the same problem*,” is syntactically and morphologically organized as any English phrase and can even mean “*be on a boat*.” What makes it special is the fact that it has an unpredictable sense and has components which cannot be replaced by any synonym (e.g., “*boat*” by “*ship*”) without removing the distinctive idiomatic meaning.

In Mel’čuk (2010), Mel’čuk separates phrasemes into two types of expressions based on whether they are defined at the pragmatic or semantic level. Pragmatic phrasemes (or pragmatemes) are produced when all the components of the expression are constrained by the situation. For example, a sign that informs car drivers they may not park in a given place should use the idiomatic expression “*No parking*” rather than the non-idiomatic “*Parking forbidden*” (Polguère 2008).

Semantic phrasemes, on the other hand, are produced when the choice of a meaning from a given conceptual representation is free but the selection of at least one component of the expression is not free. Semantic phrasemes include three main categories: clichés, collocations, and idioms.²

Clichés and collocations are compositional, i.e., for a given semantic phraseme AB, the meaning and the form of A and B are combined in accordance with the rules of the language. On one hand, clichés are phrasemes where none of the components is selected freely, i.e., cannot be replaced by a (quasi-)equivalent expression. For example, “*something*” cannot be used instead of “*one thing*” in “*one thing after another*,” while “*we all produce mistakes*” or “*we all make errors*” can be understood but are not as natural as “*we all make mistakes*.” On the other hand, collocations

²Idioms—named *locutions* in French—are seen in phraseology as a subcategory of phrasemes and are used in the remainder of this chapter as a synonymous of idiomatic expressions.

have one component³ chosen freely by the speaker and another component chosen as a function of the base. To characterize for example a battle as being very violent, “*fierce BATTLE*” is more standard than “*ferocious BATTLE*” or “*terrible BATTLE*,” while “*award a PRIZE to*” will be used to express “*give a PRIZE to*.”

Unlike clichés and collocations, idioms are non-compositional and none of their components is selected freely. In this work, we are interested in identifying the translation of this last category: idioms. Idioms can differ on the degree of transparency, the degree to which their meaning includes the meanings of their components. Here are some examples: “*let’s go Dutch*,” “*as well as*,” or “*throw up*” each has a meaning which does not include the meaning of one of its components (“*go*” and “*Dutch*” in the first example); “*heavy water*” or “*sea anemone*” include the meaning of only one of their lexical components (here “*water*” and “*sea*”) but not as their semantic pivot; “*start a family*” or “*shopping center*” include the meaning of all their components but have an additional unpredictable meaning (“*start a family*,” for example, means that a new family comes to existence but also that a first child was conceived with one’s spouse). These various degrees of semantic analyzability and semantic decomposability make idioms difficult to be identified by automatic methods or even by human annotators (Fazly et al. 2009).

Finding Translations of Idioms

For a human wishing to translate an idiom, probably the most natural way is to look it up in a dictionary. This may be difficult because idioms are so numerous and they are not all covered in a given dictionary. Mel’čuk, for example, estimated that the number of non-compositional idiomatic expressions is between 10,000 and 20,000 for any given language (Mel’čuk 2010), while other idiomatic expressions like collocations suffer from a lexical proliferation problem (e.g., “*take a walk*,” “*take a hike*,” and “*take a trip*”) and are much more numerous (Sag et al. 2002).

An alternative resource is a translation memory. TMs are databases that store sentences pairwise from the source and target languages. They are typically made of sentences previously translated by professional translators, which makes them more reliable than MT systems. Thus, they represent a valuable resource for translating idioms especially when they store a huge quantity of parallel corpora (Volk 1998). Many commercial CAT tools, such as SDL TRADOS,⁴ DEJA VU,⁵ LOGITERM⁶ or MULTITRANS,⁷ are available to manage and search information in a TM. They mainly operate at the level of sentences, which limits their usefulness to repetitive

³Shown in small caps in the examples.

⁴<http://www.trados.com>.

⁵<http://www.atril.com>.

⁶<http://terminotix.com>.

⁷<http://www.multicorpora>.

translation tasks. As we shall see in section “Survey of CAT Tools Comparable to TRANSEARCH,” not all TMs have this limitation. Tools such as TRANSEARCH or TRADOIT are able to operate at the word level since they typically embed word-alignment technology.

Searching a fixed idiom (e.g., “*of course*” or “*till kingdom come*”) is straightforward since it always occurs in the same form. Unfortunately, most idioms, in particular expressions of the type “Verb+Noun,” are syntactically well-formed phrases that allow some variability in expression (Fazly et al. 2009). There are several patterns of variability: many inflection forms can occur for a given idiomatic expression, such as “*have other fish to fry*” whose verb can be conjugated for person (“*have*,” “*has*”), tense (“*had*,” “*will have*” . . .), or mood (“*would have*,” “*having*” . . .). Some can also undergo passivation, or speaking in the passive voice, like “*the breeze was shot*” or topicalization like “*it is these strings that he pulled*.” Making matters even more complicated is that some words can be inserted within the construction of idioms, such as “*exact*” in the expression “*go exact halves*,” which is particularly prominent in languages such as German where verbs are often detached from their arguments (Anastasiou 2008). Lastly, some idioms in French, for example, can even allow some semantic replacements, such as “*louper*” or “*manquer*” which can be used instead of “*rater*” in the expression “*rater le coche*.”

Given that completely fixed idioms can be represented by a sequence of space-separated words in dictionaries and be searched verbatim inside a TM, we depend on more elaborate strategies for flexible expressions. To guide this search, a system may resort to detailed descriptions of known idioms. The PHRASE MANAGER system presented in Volk (1998) was for instance designed to identify multi-word expressions during dictionary look up, asking users to provide for each new idiom the headphrase in canonical form as well as the morphological restrictions for each individual word, and to assign it to a syntactic class specifying its possible transformations. Another solution previously proposed relies on the use of Part-of-Speech (PoS) taggers and morphological analyzers (Takeuchi et al. 2007). This system standardizes words by taking the basic form of verbs (infinitive), personal pronouns (“*one*,” “*oneself*”), and possessive pronoun (“*one’s*”), while the articles are expanded to match an occurrence of “*a*” with an entry having “*the*.” It also resorts to rules learned on a small corpus in order to allow insertions of words of some PoS categories, according to the PoS sequence of the idiom.

In the following section, we present the bilingual concordancer TRANSEARCH, which exhibits interesting properties when identifying the translation of a given query whether idiomatic or not.

TRANSEARCH

TRANSEARCH is a bilingual concordancer that allows its users to query large databases of past translations in order to find readymade solutions for a host of translation problems. Subscribers to the system consist mainly of professional

The screenshot shows the TRANSSEARCH3 BETA web interface. At the top, there are navigation links: UTILISATEUR : felipe, REQUÊTES, MON COMPTE, PRÉFÉRENCES, AIDE, and QUITTER. Below this, there is a search bar with the text 'Signet / Favori personnalisé : TransSearch (qu'est-ce que c'est ?)' and a 'Requête bilingue' button. The collection of documents is set to 'Les Hansards canadiens'. The search expression is 'is still in its infancy' with a 'Chercher' button. Below the search bar, a banner indicates '14 traductions de is still in its infancy dans 17 occurrences'. The results are displayed in a table with three columns. The left column shows various French translations of the query, each with a count. The middle column shows the English text with the query highlighted in orange. The right column shows the French translation of the English text, with the highlighted French text also in orange. The highlighted translations are hyperlinks to their occurrence in the original Hansards session.

| en est encore à ses premiers balbutiements | en est encore à ses premiers balbutiements | |
|--|---|---|
| 3 | While the technology is still in its infancy , autologous stem cell therapy, drawing on the patient's own stem cells, is being used in a breathtaking variety of applications to replace or repair damaged tissues, including the heart or other organs damaged by cancers, that often lead to the full recovery of the patient. | La technologie en est encore à ses premiers balbutiements , mais les traitements autologues au moyen de cellules souches, c'est-à-dire à partir des cellules souches du patient lui-même, trouvent une variété impressionnante d'applications dans le remplacement ou la régénération des tissus endommagés, y compris dans la régénération du cœur et d'autres organes endommagés par un cancer, et peuvent conduire à la guérison complète du patient. |
| 2 | The gun control program is still in its infancy , yet data suggests it has already caused a decline in gun deaths and crimes. | Le programme de contrôle des armes à feu en est encore à ses premiers balbutiements et pourtant, mais les données révèlent qu'il a déjà entraîné une baisse du nombre de décès par balles et de crimes commis à l'aide d'une arme à feu. |
| 1 | Electric technology is still in its infancy . | La technologie électrique en est encore à ses premiers balbutiements . |

Fig. 9.1 Result returned by the new TRANSSEARCH to the query “*is still in its infancy*.” The left column shows translations from the most likely to the least likely, while the main columns shows concordances. The query and the selected translation are shown in color in each of them. The highlighted translations are hyperlinks to their occurrence in the original Hansards session

translators. A recent study of the query logs of this application exhibits that TRANSSEARCH is used to answer difficult translation problems (Macklovitch et al. 2008). Among the 7.2 million queries submitted to the system over a 6-year period, 87% contain at least two words. Among the most frequently submitted queries, several appear to be idiomatic, like “*in light of*” (544 times) or “*out of the blue*” (508 times).

System Features

TRANSSEARCH, which has been made available since 1996 through a Web interface by the Université de Montréal (Macklovitch et al. 2000), has evolved into not only a bilingual concordancer but also a translation finder (Bourdaillet et al. 2010). Figure 9.1 which displays the results for the query of the idiomatic expression “*is still in its infancy*” exemplifies the new capabilities of the system. Where a simple bilingual concordancer (as were the previous versions of TRANSSEARCH) would simply display a list of parallel sentences containing the query in their English part, the new version of TRANSSEARCH highlights for each sentence pair the French part

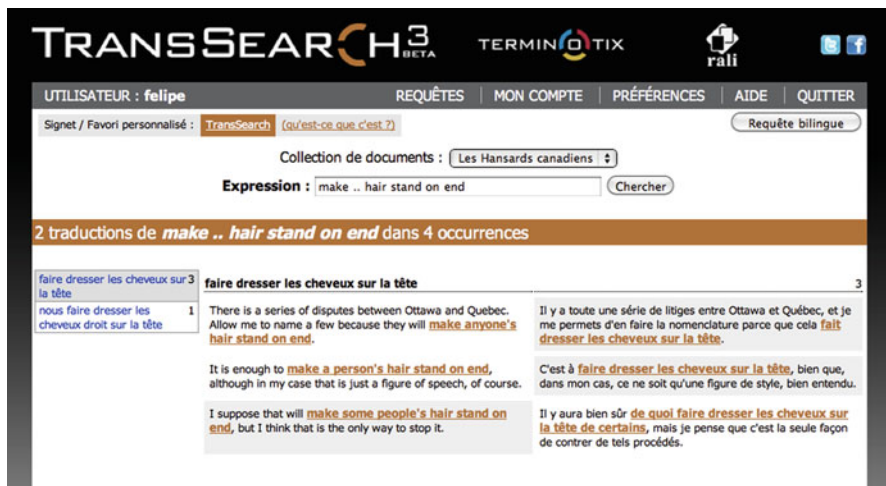


Fig. 9.2 Result returned by TRANSSEARCH to the query “make .. hair stand on end”

associated with the query. Besides, this version displays on the left-hand side the whole range of translations (automatically) found in the TM. For the first suggested translation, “*en est encore à ses premiers balbutiements*,” three of the sentence pairs containing a variant of this translation (see the merging process described in section “Processing Steps”) are displayed in context. With respect to an ordinary bilingual concordancer, where the identification of translations in sentences is left to the user, we believe the new version of TRANSSEARCH dramatically improves usability, by displaying a general view of the TM content for a given query.

The previous query example has shown that the system is able to find results for queries with several words. The user can also submit more advanced queries to search discontinuous expressions. For example, Fig. 9.2 displays the results for the query “make .. hair stand on end.” The “..” operator enables the user to indicate the system that occurrences of 2 words in the query (here “make” and “hair”) can be up to 5 words apart inside a sentence. Another operator “...” allows for searches without constraining the distance between two words. From a linguistic perspective, these two operators are useful since they enable the user to spot expressions where words may be separated by a few words, such as nominal groups in the examples of Fig. 9.2.

Another advanced type of query that is also available in TRANSSEARCH is known as morphological expansions. The system thus considers all the morphological derivations of the terms associated with the “+” symbol, when retrieving sentence pairs. Figure 9.3 shows the results for the query “take+ no for an answer.” In this example, the interface displays expressions containing different inflected forms of the verb “take.” This last operator is especially useful for morphologically rich languages like French or Spanish and allows the user to spot translations without having to worry about all possible inflections for such expressions.

The screenshot shows the TRANSSEARCH 3 BETA web interface. At the top, there are navigation links: UTILISATEUR : felipe, REQUÊTES, MON COMPTE, PRÉFÉRENCES, AIDE, and QUITTER. Below these, there is a search bar with the text 'Signet / Favori personnalisé : TransSearch (ou'est-ce que c'est ?)' and a 'Requête bilingue' button. The main search area displays 'Collection de documents : Les Hansards canadiens' and the search expression 'take+ no for an answer' with a 'Chercher' button. Below the search bar, a banner indicates '13 traductions de take+ no for an answer dans 16 occurrences'. The results are presented in a table with three columns: a list of French phrases, the English text, and a list of French phrases. The first row shows 'accepter un non comme réponse' in French, 'The older gang members, when they approach these 10 and 11 year olds, whom they want to perform certain crimes for them because they are under a certain age, do not taking no for an answer.' in English, and 'Quand ils demandent à des jeunes de 10 et 11 ans parce qu'ils veulent leur confier certaines fonctions qui leur conviendraient en raison de leur jeune âge, les plus âgés au sein de ces gangs n'acceptent pas un non comme réponse.' in French. The second row shows 'on lui oppose une réponse négative' in French, 'If the Hon. Member cannot take no for an answer, maybe he could get someone else to ask a question.' in English, and 'Si le député ne peut pas accepter un non comme réponse, il devrait peut-être demander à quelqu'un d'autre de poser une question.' in French. The table also includes a count of occurrences for each phrase.

Fig. 9.3 Result returned by TRANSSEARCH to the query “take+ no for an answer”

By default, TRANSSEARCH searches for the given expression regardless of languages (French or English). In some cases, however, it is necessary to specify the language, for instance in order to distinguish between the French and English words “tape” (“to hit” in French). Using the same mechanism, it is also possible to look up occurrences of a specific translation of a given query by filling both the French and English fields of the query form. For example, a user can check that “*les dés sont pipés*” is a correct translation of “*the dice are loaded*” by looking at both expressions in the TM sentence pairs.

Processing Steps

In order to suggest several translations for a given query, TRANSSEARCH performs several processing steps that we briefly describe hereafter. Many current computer-assisted translation tools mainly rely on sentence-level matching to exploit their translation memory. TRANSSEARCH operates at a finer-grained level using word alignment techniques, which are commonly used in SMT. The term translation spotting, coined by Véronis and Langlais (2000) and relabeled by the authors as *transpotting*, is defined as the task of identifying the target language word-tokens that correspond to a given source language query in a pair of sentences known to be mutual translations; it is a core step in the new version of TRANSSEARCH.

We call *transpot* the target word-tokens automatically associated with a query in a given pair of sentences. For instance in Fig. 9.1, “*en est encore à ses premiers balbutiements*” and “*soit encore tout nouveau*” are 2 out of 14 distinct transpots displayed to the user for the query “*is still in its infancy*.”

The method used to transpot queries in the retrieved sentence pairs is described in detail elsewhere (Bourdaillet et al. 2010). In sum, our transpotting algorithm uses statistical word-alignment models and enforces that the transpots identified are sequences of contiguous words. As mentioned in Simard (2003), contiguous tokens in the source language sentence tend to be aligned with contiguous tokens in the target language. This statement is confirmed by the good experimental results presented in the study of Bourdaillet et al. (2010).

Queries that occur frequently in the TM receive numerous translations using the transpotting methods described above, some of course being clearly wrong; others being redundant (morphological variations of the same translation). We estimate that since a user will focus on the 10 first translations presented, we want to provide as many correct and diversified translations as possible at the top of the result page. Therefore, two postprocessing steps were introduced inside the TRANSSEARCH engine. The first one filters out bad transpots using supervised learning. To do this, a classifier was trained on a corpus where transpots were manually labeled as “good” or “bad,” using features such as the ratio of grammatical words inside the hypothesized transpots. Once transpots have been filtered out, the second step merges those which are different inflectional forms of the same sequence of canonical words. For instance, “*au nom du*” and “*au nom des*” will be considered as similar, since “*du*” and “*des*” are contractions of “*de + le*” and “*de + les*,” respectively, where “*le*” and “*les*” are definite articles. Furthermore, as we noticed that translations that differ only by a few grammatical words or punctuation marks, like “*de la part de*” and “*part de*,” are often redundant for the user, those translations are combined as well. At the end of this second post-processing step, only the most frequent transpot of each merged set is displayed on the left-hand side of the user interface (see Figs. 9.1–9.3). These transpots are shown as a list sorted in the decreasing order of their transpotting frequency.

Methodology

The relevance of the results produced by the TRANSSEARCH engine is closely related to the indexed translation memory. This section presents the data used by our system and describes how queries were submitted to test the behavior of the concordancer for idiomatic expressions.

Resources

Translation Memory

The largest TM used in TRANSEARCH comes from the Canadian Hansards, a collection of the official proceedings of the Canadian Parliament. For our experiments, we used an in-house sentence aligner (Langlais 1997) to align 8.3 million French–English sentence pairs extracted from the 1986 to 2007 period of the Hansards. This bitext was indexed with Lucene⁸ to form our TM. Let us note that this corpus, although it is produced in Canada, can be mainly seen as expressed in a “standard” French rather than in a Canadian variety of the French language. Sentences are uttered in a formal context and contain very few typical Canadian expressions with respect to everyday French Canadian spoken language.

Idiom Lexicon

As mentioned above, determining whether an expression is idiomatic or not is certainly not an easy task. Therefore, we employed the phrase book (Piat 2008) written by Jean-Bernard Piat, a translation teacher as well as a translator. This book which is oriented toward general public provides a list of 1,467 idiomatic expressions in both languages (French and English) categorized by subjects (e.g., “Human body”).

According to the author, the expressions were chosen because they are frequently used. A minority of these expressions are informal (e.g., “*to be well-upholstered*”). He also mentioned that sometimes he could not find an idiom (e.g., “*travailler tard dans la nuit*”) in one language to translate idiomatic expressions from the other language (e.g., “*to burn the midnight oil*”).

Examples of entries in this book are reported in Table 9.1. In order to illustrate the problem with translating those expressions, we provide the translations produced by GOOGLE TRANSLATE. A few entries have several equivalent translations such as “*make your flesh creep*” and “*give you goose pimples*” for “*donner la chair de poule*.” Globally, there are on average 1.17 English translations and 1.01 French translations per entry.

All expressions but seven are used in the context of a sentence. According to the author, providing expressions in context makes them easier to understand and to use. The lexicon contains a high proportion of verbal phrases (around four out of five of the available entries) that are used in their inflected form, like “*He took to his heels*” for the phrase “*to take one’s heels*.” Other entries are fixed expressions such as “*When there’s a will, there’s a way*” or “*Hands off!*”

⁸<http://lucene.apache.org>.

Table 9.1 Excerpt of the entries we considered in our experiment. R stands for the reference translation, G stands for the translation made by GOOGLE TRANSLATE (which provides here literal translation). Words in parenthesis have been manually marked as contextual words that are not part of the idiomatic expression

| French | | English |
|--|---|---------------------------------------|
| Il est agile comme un singe | R | He's as nimble as a goat |
| | G | He is agile as a monkey |
| Elle était sur son trente et un | R | She was dressed to kill |
| | R | She was all dressed up |
| | G | She was on her 31 |
| (Je vais d'abord) me rincer la dalle —familiar— | R | (I'm going to) wet my whistle (first) |
| | G | First I'll rinse my slab |
| (Il aime) rouler des mécaniques —familiar— | R | (He likes) flexing his muscles |
| | R | (He likes) playing the tough guy |
| | G | He loves rolling mechanical |
| J'ai vu trente-six chandelles | R | I saw stars |
| | G | I saw 36 candles |

Preprocessing

In order to take into account contextualization that makes lexicon entries too specific, the used lexicon was manually annotated by the first author of this chapter. All words judged as extra-information with respect to the idiomatic expression were annotated as such in the lexicon. Those are the words in parenthesis in the examples of Table 9.1. They are typically modal verbs (e.g., “can,” “must”), semi-modal verbs (e.g., “am going to,” “are likely to”), catenative verbs (e.g., “want to,” “keep”), adverbs (e.g., “only,” “finally”), adverbial phrases (e.g., “in Italy,” “when he heard the news”), or noun phrases (e.g., “this poet,” “his latest book”). Finally, at least one word was classified as extra-information for 486 out of 1,467 entries.

Queries to the Translation Memory

In order to test the ability of TRANSEARCH to find translations for idioms, three types of queries were submitted to the system: queries built from either the English side or the French side of the entry, and bilingual queries where both sides were searched at the same time. As mentioned in section “Resources,” a few entries have more than one English or French reference translations. For these entries, we collected results from all the equivalent translations. Since the TRANSEARCH user interface does not allow users to write an “or” operator between several equivalent translations, we had to simulate the behavior of this operator by submitting independent translations and then by merging the results retrieved by TRANSEARCH.

Table 9.2 Percentage of the 1,467 lexicon entries found inside the translation memory using several types of query

| Query types | Bilingual (%) | English (%) | French (%) |
|--|---------------|-------------|------------|
| <i>Verbatim queries</i> | 3 | 9 | 17 |
| EN: <i>I have no axe to grind</i> | | | |
| FR: <i>Je ne prêche pas pour ma paroisse</i> | | | |
| + Manual removal of extra words | 6 | 21 | 28 |
| EN: <i>I have .. axe to grind</i> | | | |
| FR: <i>Je .. prêche .. pour ma paroisse</i> | | | |
| + Removal of extra pronouns | 8 | 30 | 35 |
| EN: <i>have .. axe to grind</i> | | | |
| FR: <i>prêche .. pour ma paroisse</i> | | | |
| + Verb lemmatization | 14 | 43 | 44 |
| EN: <i>have+ .. axe to grind</i> | | | |
| FR: <i>prêcher+ .. pour ma paroisse</i> | | | |
| + Pronoun and determiner lemmatization | 16 | 48 | 48 |
| EN: <i>have+ .. axe to grind</i> | | | |
| FR: <i>prêcher+ .. pour sa+ paroisse</i> | | | |

Table 9.2 shows the number of lexicon entries found in the TM, using bilingual (column 2), English (column 3), or French queries (column 4) and considering various ways of querying the system. As expected, building verbatim queries from the lexicon leads to retrieve information inside the TM for a small number of expressions only (line 1). After taking into account the manual preprocessing step introduced in section “Preprocessing,” that is, after removing extra words, twice as many queries had at least one hit in the TM (line 2). Still, at best, a user could retrieve no more than 28% of the French expressions by simply querying them verbatim or by removing extra words.

An inspection of the submitted queries revealed that many of them correspond to flexible idioms, that is, idiomatic expressions that can vary from one occurrence to another. In order to capture those variations and to increase the number of hits in the TM, we used a mix of linguistic information as well as the operators we described earlier. In so doing, we resisted the temptation of adjusting this process for each query and instead applied some rules in a systematic way, given a set of linguistic markers semiautomatically annotated in the lexicon.

The performed processing steps for the entry [*“I have no axe to grind,” “Je ne prêche pas pour ma paroisse”*] are illustrated in Table 9.2. A set of rules deleted personal pronouns at the beginning of an expression (see line 3); a list of pronouns to be removed has been collected for this purpose in each language. Then, lemmatized verbs were replaced by the corresponding lemma and auxiliary verbs were removed

(see line 4); we used for this an in-house lemmatization resource available for both languages. Last, we also considered lemmatizing pronouns and determiners within an expression (see line 5).

It should be noted that we chose to modify entries using a set of limited rules in order to avoid over-abstracting idiomatic expressions. For instance, we noticed that the indefinite pronoun “*it*” in English usually occurs in fixed expressions and thus cannot be replaced by another personal pronoun. As a result, we kept this pronoun verbatim in the queries made. For the same reason, we did not automatically remove negation since it may belong to the idiomatic expression. The idiom “*I did not sleep a wink*” becomes for example incorrect if “*not*” is removed. We are also aware that all verbs or nouns cannot be lemmatized for all idioms, like the verb “*to be*” in “*Enough is enough*.” We count on the fact that the incorrect inflection forms of a given expression usually do not occur in the TM.

Observations

We observe in Table 9.2 the dramatic increase of the number of hits in the TM according to the level of abstraction of the query. At best, the rewriting rules we applied allow TRANSEARCH to return sentence pairs for 700 English entries and for 705 French entries, i.e., roughly half of the lexicon. Each set of rules increases the number of queries with at least one hit. Surprisingly, verb lemmatization led to a higher improvement of the coverage for English queries than for French ones. This shows that, on the contrary to what we expected first, this process is also relevant for weakly inflected languages.

This experiment also shows that in order to get the best of the system, users should use the linguistic operators at their disposal. We know, however, that most queries made by real users of the application do not use those operators. This could mean one of two things: when users submit a query to the system without getting any answer, they might simply abandon the search for a translation or they might figure out a way to process the query in order to find a match in the TM. Inspecting the log-files of the application exhibits evidences that both strategies happen in practice. This means that automatically processing the query of a user is an interesting prospect to consider.

Another interesting outcome of the experiment we conducted is that the Hansards indexed by TRANSEARCH are good at identifying the idiomatic expressions we considered. A previous study with this corpus in the medical domain has already shown that the Hansards are a valuable source of information for specialized domains (Névél and Ozdowska 2006). In this work, we analyzed the responses of TRANSEARCH with respect to the 20 categories used for labeling the different idiomatic expressions. The main outcomes of this analysis are reported in Table 9.3. We observe a large discrepancy among classes. While nearly 70% of expressions in the “Behaving” class were retrieved from the Hansards (e.g., “*go out on a limb for someone*,” “*to jump on the bandwagon*”), only 10% of those belonging to the

Table 9.3 Coverage of the lexicon entries by the TM of TRANSEARCH for various topics

| Ranks | English queries (%) | | French queries (%) | |
|-------|--------------------------|----|----------------------------------|----|
| 1 | Behaving | 69 | Behaving | 68 |
| 2 | Discussion | 68 | Feelings and emotions | 62 |
| 3 | Time, age and experience | 65 | Discussion | 61 |
| | ... | | ... | |
| 18 | Love, sex and seduction | 14 | Human body and physical activity | 22 |
| 19 | Weather | 14 | Clothing and fashion | 13 |
| 20 | Drinking and eating | 8 | Weather | 10 |

“Weather” class were found (e.g., “*It’s biting cold*,” “*The sun is beating down*”). This strengthens the interest of including more bilingual resources inside the TM for better coverage of topics.

Evaluation

We have measured the quantity of idiomatic expressions we could find by querying the Hansards indexed by TRANSEARCH. We now turn to the evaluation of how good the application is for spotting the translations of the retrieved expressions. This evaluation encompasses three related experiments: (1) The recall of the translations identified by TRANSEARCH among the entries of the reference lexicon is first evaluated. (2) These results are then compared with the ones obtained using the widely used and effective MT engine known as GOOGLE TRANSLATE. (3) We finally provide a manual evaluation of the precision of our system.

Objective Evaluation of the Recall Capabilities of TRANSEARCH

For the French and English queries obtained after applying our rewriting rules, TRANSEARCH was able to retrieve on average respectively 36.1 and 31.7 sentence pairs from the TM. Among this material, the transpoting algorithm identified respectively 12.5 French and 14.9 English (different) translations (shown to the user on the left of the navigator). Since a manual analysis of all the suggested translations would be a tedious task, an evaluation was performed thanks to the sanctioned translations belonging to the idiom lexicon described in section “Methodology.” As shown in Table 9.2 (last line), a query and its sanctioned translation are found simultaneously in the sentence pairs returned by the system for 238 lexicon entries (16%). Therefore we restrained our objective evaluation to those 238 queries.

Table 9.4 Recall (%) measured using the lexicon sanctioned by the translation memory as a reference

| k | 1 | 2 | 3 | 5 | 10 | ∞ |
|-----------------|------|------|------|------|------|----------|
| English queries | 41.6 | 56.3 | 59.2 | 65.1 | 69.3 | 74.8 |
| French queries | 41.8 | 49.8 | 54.9 | 62.9 | 69.6 | 76.8 |

Table 9.4 provides the proportion of those queries where the k -first translations displayed by TRANSSEARCH contain (at least) one of the reference translations sanctioned by the lexicon.⁹

The recall of 75% measured when all the translations returned by the system are considered demonstrates that the embedded transpotting algorithm has the ability to find translations in the retrieved sentence pairs. The result of 41,6% obtained when considering the first translation returned by the system (that is, the most frequent one) is not bad either, especially since the reference we used is rather incomplete. For instance, our lexicon contains the translation “*être dans un état second*” for the idiom “*to be in a daze*,” while TRANSSEARCH displays this translation after “*est nébuleux*,” which is as well a good translation of the English idiom. Similarly, TRANSSEARCH returns no less than 34 different translations¹⁰ of the query “*be+ around the corner*,” most of which being perfectly legitimate translations, while our reference contains only one.

Comparison of TRANSSEARCH with GOOGLE TRANSLATE

TRANSSEARCH is able to suggest several translations, provided, however, that enough information is available in its translation memory. We compared its results with those of GOOGLE TRANSLATE online system.¹¹ As mentioned in the introduction of this chapter, MT tools do not usually adopt specific strategies for idiomatic expressions. This makes them prone to errors, as was the case in the examples of Table 9.1. Nevertheless, GOOGLE TRANSLATE has the benefit of relying on bilingual resources that are much broader than those exploited by TRANSSEARCH.

Table 9.5 shows the recalls measured for both applications. In this case, we took the full lexicon into account. These results were computed for TRANSSEARCH from the queries obtained at the last processing step described in section “Queries to the Translation Memory”; as far as GOOGLE TRANSLATE is concerned, we took the queries obtained after a manual removal of extra words of the full lexicon entries (step 2 in Table 9.2) since the MT engine does not have operators equivalent to those of our CAT tool.

⁹In order to account for inflectional variations, we compared lemmatized translations.

¹⁰The 10 most frequent ones are: *est à nos portes*, *arrive à grand pas*, *était imminent*, *nous attend*, *me guette*, *est sur le point*, *s’annonce*, *est en vue*, *sommes au bord de*, and *survenir*.

¹¹<http://translate.google.com>.

Table 9.5 Recall (%) measured taking into account the full lexicon

| k | | 1 | 2 | 3 | 5 | 10 |
|-----------------|------------------|------|------|------|------|------|
| English queries | GOOGLE TRANSLATE | 12.3 | | | | |
| | TRANSEARCH | 8.0 | 10.6 | 11.3 | 12.4 | 13.3 |
| French queries | GOOGLE TRANSLATE | 12.6 | | | | |
| | TRANSEARCH | 7.6 | 9.1 | 10.3 | 11.9 | 13.2 |

Table 9.6 Recall (%) measured on the 700 English queries and the 705 French queries found respectively in the translation memory of TRANSEARCH

| k | | 1 | 2 | 3 | 5 | 10 |
|-----------------|------------------|------|------|------|------|------|
| English queries | GOOGLE TRANSLATE | 15.7 | | | | |
| | TRANSEARCH | 16.7 | 22.3 | 23.6 | 26.0 | 27.9 |
| French queries | GOOGLE TRANSLATE | 16.7 | | | | |
| | TRANSEARCH | 15.9 | 18.9 | 21.4 | 24.7 | 27.5 |

The obtained recall values are lower for TRANSEARCH than for GOOGLE TRANSLATE, particularly at the first rank; this may be explained by the fact that for half of the queries, the concordancer could not find any information in its TM. Surprisingly, the results generated by GOOGLE TRANSLATE are higher than expected. They indicate that a large part of this lexicon is likely to be in the resources used by GOOGLE TRANSLATE and that this system is able to find the corresponding entries inside the translation table.

To alleviate the fact that TRANSEARCH is not able to suggest a translation for all queries in contrast to GOOGLE TRANSLATE, we carried out additional experiments restrained to the queries with at least one result provided by TRANSEARCH. The findings showed that the recalls (reported in Table 9.6) are close upon comparing both systems when only the first result displayed by the TM-based system is considered. Finally, TRANSEARCH suggests several translations, which increases recall from 16% to 28%. This is important because it is often the case that a typical user of the concordancer wants to collect different translations of a given expression, something that GOOGLE TRANSLATE does not facilitate.

Manual Evaluation of TRANSEARCH

While the objective evaluation of the recall capabilities of TRANSEARCH, presented in section “Objective Evaluation of the Recall Capabilities of TRANSEARCH” above, revealed the great potential of TRANSEARCH for translating of idiomatic expressions, it also showed that a manual evaluation of the system was required in order to account for the sparseness of our bilingual lexicon. As a result, we conducted a manual evaluation involving five bilingual annotators who were

| | | | |
|---|---------|---------|---------|
| appeler un chat un chat | J1 | J2 | J5 |
| ▷ we should call it what it is | correct | correct | correct |
| ▷ we can say the d word and the m word | correct | wrong | partial |
| ▷ calling manure a rose doesn't change the smell | correct | wrong | partial |
| manger à tous les râteliers | J1 | J2 | J5 |
| ▷ slurps at everyone 's trough | correct | correct | correct |
| ▷ double-dipper | partial | correct | partial |
| ▷ them pot lickers and accusing them of being at the trough and pork barrelling | wrong | partial | wrong |

Fig. 9.4 Examples of annotations of some French idiomatic queries

Table 9.7 Average percentage of translations judged correct, partially correct, or wrong per query on a sample of 100 French queries randomly selected. *avr* stands for the average number of translations produced per query, while *rank* indicates the average rank of the first translation labeled as correct by at least one annotator

| Correct | Partial | Wrong | <i>avr</i> | <i>rank</i> |
|---------|---------|-------|------------|-------------|
| 42% | 22% | 36% | 13.4 | 1.4 |

presented with lists of identified translations for 100 randomly chosen French queries and were asked to indicate in those lists those translations that they found correct, partially correct or wrong. No specific guidelines were given to explain these labels. The annotators were broken up in two groups. The first group consisted of three annotators who judged the first fifty French queries; the second group, consisting of remaining two annotators, judged the next 50 queries.

Across the board the quality appreciated by the annotators turned out to be highly variable, some annotators tending to classify more easily translations as correct. This variability in translation accuracy equated with a low Fleiss inter-annotator agreement value (Fleiss et al. 2003) of 0.25. Figure 9.4 illustrates some cases of divergence.

The results of this evaluation are reported in Table 9.7. To control for the occurrence of inter-rater variability in which a given query can be rated differently by several judges, we decided to credit divergent annotations equally. For instance, if a translation is judged correct by one annotator, and wrong by another one, a credit of 0.5 will be given to each label respectively.

For all but 7 out of 100 queries, TRANSSEARCH was able to identify a translation classified as correct by at least one annotator. For these queries, the average rank of the first correct translation was 1.4. This indicates that relevant translations can usually be found among the first two candidate translations that are displayed by TRANSSEARCH. In addition, on average, we observe that only 36% of the translations proposed to the user are labeled as wrong.

Survey of CAT Tools Comparable to TRANSSEARCH

This section reviews three recent CAT tools that are able to automatically identify translations from their TM. Similar to TRANSSEARCH, these CAT tools usually resort to statistical word-level alignment methods, bringing them much closer to the capacities of MT engines than classical CAT tools; those last systems being mainly concerned with recycling parallel sentences as a whole.

The CAT systems presented in this section do not use the same techniques to align words found in a given sentence pair and have different user interfaces as well. Unfortunately, the transpoting methods that are used are seldom described in detail and have not been evaluated with the same kind of rigor (as we showed in our evaluation of TRANSSEARCH) on a significant amount of queries, either with idiomatic expressions such as in this chapter or with other expressions considered in a previous study (Bourdaillet et al. 2010). Experiments should be therefore carried out with the CAT systems to compare them in terms of recall and precision of the identified translations. Since we did not have access to the search engines of these systems, realizing that the lack thereof undoubtedly complicates the automatic processing of the results, we decided to focus in this section on the comparison of their functionalities.

LINEAR B

The CAT tool LINEAR B is available on the Web to translate expressions between English and eight other languages: Arabic, Chinese, Dutch, French, German, Italian, Spanish, and Swedish. Figure 9.5 displays the interface of the system when the query “*is still in its infancy*” was posed to the system. It is reported at the top of the screen that 18 translations are found. Translations of the query are suggested below, with examples taken from the TM. For long sentences, only a part is displayed, with missing words substituted by “...”. Let us note that at most three occurrences are shown by default for each suggested translation while a hyperlink allows the system access to more examples if they are available inside the TM. For each sentence pair, both the query and its corresponding translation are highlighted in bold.

To the best of our knowledge, it was the first online CAT system which had the capacity to automatically identify translations of expressions using recent SMT methods. The process that builds the TM and then efficiently searches translations is described in Callison-Burch et al. (2005). In short, a phrase table is automatically built from parallel corpora aligned at the sentence level drawing on methods that are usually employed to train SMT models (Koehn et al. 2003). This phrase table is then stored in a suffix array data structure in order to efficiently look up the possible translations of a phrase.

For expressions that are not found verbatim in the TM, the system provides the list of the subsequences (sequences that are derived from another sequence) that

LINEAR B from English ▾ to French ▾ Search

Results 18 possible translations for **is still in its infancy** (phrase occurred in 23 sentences)

n' en est qu' à ses débuts - [1 sentence matched]
 social dialogue **is still in its infancy** , and it is important for us to leave it to the ...
 le dialogue social **n' en est qu' à ses débuts** et il faut que nous laissons aux partenaires sociaux le soin ...

est peu , - [1 sentence matched]
 it **is still in its infancy** but enough experience has already been gained to provide the basis for an initial assessment .
 c' **est peu** , mais c' est déjà suffisant pour un premier bilan .

est encore très jeune - [1 sentence matched]
 ... end to the democratisation process which , all said and done , **is still in its infancy** .
 ... terme au processus d' installation d' une démocratie qui , somme toute , **est encore très jeune** .

trouve encore dans l' enfance - [1 sentence matched]
 while the former has evolved to the point of putting the euro into the citizens ' pockets , the latter **is still in its infancy** .
 la première s' est développée jusqu' à l' arrivée de l' euro dans les poches des citoyens , la deuxième se **trouve encore dans l' enfance** .

est seulement à ses débuts - [1 sentence matched]
 the internet **is still in its infancy** , but statistics show that participation is growing exponentially .
 internet en **est seulement à ses débuts** mais les statistiques montrent que le nombre d' usagers croît de manière exponentielle .

sont encore au stade de la petite enfance - [1 sentence matched]
 in the netherlands , the competition authority **is still in its infancy** .
 aux pays-bas , les autorités compétentes en matière de concurrence en **sont encore au stade de la petite enfance** .

Fig. 9.5 Result returned by LINEARB to the query “*is still in its infancy*.” Only the first six suggested translations are displayed here but the next results can be accessed via a scroll bar

LINEAR B from English ▾ to French ▾ Search

Partial Match Found

proud as
as a
proud as a peacock

©2009 Linear B

Fig. 9.6 Result returned by LINEARB to the query “*proud as a peacock*”

were found inside the indexed phrase table. For example, the interface reports that a partial match was found for the query “*proud as a peacock*” when translating to French and suggests three subsets of queries (Fig. 9.6). It is important to note that no subsequence with the term “*peacock*” was found inside the TM.

LINGUEE

LINGUEE is a CAT tool combining a dictionary and a search engine with which users can search through a TM for words and expressions. Developed by Linguee GmbH, the system was officially launched online in 2010¹² to translate between English

¹²<http://www.linguee.com>.

The screenshot displays the LINGUEE website interface. At the top, there is a navigation bar with links: About Linguee, Linguee en français, Contribute, Login, Tools, Advertising, Feedback, and Help. A language selector shows English and French. The search bar contains the query "is still in its infancy" and a "Search" button. On the right, there is a "BUSINESS & MORE" logo.

Below the search bar, the "Editorial Dictionary:" section shows "No exact hit." and "Non-exact matches:" with a list of related terms and their parts of speech, such as "infancy" (noun), "enfance" (f), "bas âge" (m), "encore" (adv), "toujours" (adv), "toutefois" (adv), "pourtant" (adv), "néanmoins" (adv), "quand même", "still" (noun), "alambic" (m), "immobile" (adj), "silencieux" (adj), "tranquille" (adj), "its" (pron), "sa" (pron), "son" (pron), "en" (prep), "dans" (prep), "à" (prep), "être" (verb), and "exister" (verb).

The main section, "Translation examples from external sources for 'is still in its infancy':", displays a table of English and French sentence pairs. Each pair includes the source of the translation (e.g., fra.europa.eu, ecb.europa.eu, daccess-ods.un.org, www2.parl.gc.ca, eur-lex.europa.eu). The English sentences are on the left, and the French translations are on the right. The words "is still in its infancy" and its French equivalents are highlighted in yellow. The French translations are: "Finlande n'en sont encore qu'à leur début", "La monnaie électronique fait actuellement ses premiers pas, mais on ne peut exclure la possibilité d'une croissance exponentielle, notamment du fait de son utilisation pour effectuer de petits paiements pour des biens et des services proposés sur Internet et de l'introduction de l'euro (qui rend les transactions transfrontalières [...])", "La mesure du capital immatériel (actifs incorporels) en est encore à ses balbutiements.", "La technologie électrique en est encore à ses premiers balbutiements.", "Bien que la monnaie électronique en soit encore à ses débuts et que son développement reste, à ce jour, très modeste, on ne peut exclure l'éventualité d'une [...])", "Le secteur financier non bancaire n'en est toujours qu'à ses premiers balbutiements.", and "Ils ont été mis en place à une époque où le marché intérieur de l'énergie n'en était qu'à ses premiers balbutiements."

Fig. 9.7 Result returned by LINGUEE to the query “*is still in its infancy*”

and four other languages: French, German, Portuguese, and Spanish. According to a press article released in 2011 by the company, LINGUEE answers 1.5 million search requests every day.

Figure 9.7 shows the default view for the query “*is still in its infancy*.” Vocabulary entries from the dictionary is displayed on the left. If the queried expression had been found in the dictionary, the translation would have been shown on the top of the screen. The dictionary entries that match partially the expression are displayed below. On the bottom—not displayed in Fig. 9.7—a button “Suggest translation” allows users to type a translation for the query; this translation will be added to the dictionary after being checked by an editor. The right-hand side displays example sentences from the TM. For each sentence pair, its origin is shown (the Europarl corpus for the first sentence of our example), while the query and its hypothesized translation are highlighted. The full text that contains the sentence pair can be downloaded by clicking on the hyperlink displaying the origin. Various color intensities show the confidence in the alignment of a word of the hypothesized translation with the query. Like LINEAR B, long sentences are cut, with missing parts replaced by “[...]” The sentence pairs are not organized according to the identified translations. The user can rate a translation by clicking on the thumbs

symbol which appears on the right side of each sentence pair. By rating translations, the user can influence the order in which the search results are displayed following future search requests.

According to the LINGUEE web site, the TM is made of hundreds of millions of bilingual texts. The majority of the indexed sentence pairs are from professionally translated web sites of organizations, companies, and universities. Other sources include European Union documents and patent specifications, as well as various Web pages crawled and filtered with a machine-learning model. This model is continuously trained using the user feedback on the translation quality. Since some sources are less trustworthy, a warning sign is added before some displayed pair of sentences in order to indicate a possibly wrong translation. The highlighting of the identified translation in the example sentences is generated automatically though we were not able to find information on this process; word-based algorithms (Brown et al. 1993) used in MT are likely to be used for this purpose.

The query submission system integrates different functionalities. A “did you mean” feature suggests an expression close to the submitted query when the number of hits in the TM is too small. Possible completions of queries are also displayed when typing. Quotes can be used to search for exact phrases in order to find only sentences in which the query words occur in the exact form and order. Two operators are introduced: a “-” sign is helpful to exclude words; a “+” sign forces the inclusion of words in a certain form, which is helpful with pronouns which would otherwise be ignored by LINGUEE. Finally a morphological analysis enables the system to search some inflected forms of a given query. For example, sentence pairs containing “*was still in its infancy*” or “*is still in its infancy*” are retrieved for the query “*being still in its infancy*.”

TRADOOIT

TRADOOIT is a CAT developed by Okidoo Inc. It includes a TM, a term bank, and a bilingual concordancer. This product has been marketed since 2008 and a free version is available online since November 2011 for the English–French pair.¹³ Only this free version is reviewed here.

The system uses two kinds of resources. The first one is a TM made of various parallel corpora, delivered mainly by the Canadian government (including the Hansards), the Ontario government, the European Parliament, the World Trade Organization, UNESCO, and opensubtitles¹⁴ which provides movie subtitles. The whole TM contains more than 200 M words per language. The second resource type includes various term banks, including Wikipedia, and TERMIUM PLUS¹⁵ which

¹³<http://www.tradooit.com>.

¹⁴<http://www.opensubtitles.org>.

¹⁵<http://www.termiumpius.gc.ca>.

The screenshot shows the TRADOIT BETA search interface. At the top, there's a search bar with the query "is still in its infancy" and a magnifying glass icon. To the right of the search bar, there are language selection buttons for "English" and "French", and a "Log in / Sign up" link. Below the search bar, it indicates "36 résultats (0,448 s)".

The main content area is divided into three columns:

- Grouped Translations:** This section lists various translations of the query with their frequencies. For example, "est [2]", "est encore très jeune [2]", "est encore à ses premiers balbutiements [2]", "n'en est qu'à ses débuts [1]", "soit à ses balbutiements [1]", "n'en est qu'à ses balbutiements [1]", "qu'au tout début [1]", "enfance [1]", "sont encore au stade de la petite enfance [1]", and "trouve encore dans l'enfance [1]".
- 1001 Forms:** This section lists different forms of the query, such as "is still in its infancy [36]".
- Sources:** This section lists the sources of the translations, including "EUROPARL [24]", "HANSARD [9]", "NEWSGCCA [2]", and "NRC [1]".

On the right side, there are two columns for the source language (English) and target language (French) sentence pairs. Each pair includes the source text, the target text, and a "See bitext" link. For example, the first pair shows the English sentence "Lastly, we are being very badly affected by an imbalance, on the issue of fraud, between a repressive and largely administrative unit and a judicial unit with powers of jurisdiction that **is still in its infancy**." and the French translation "Enfin, nous souffrons très fortement d'un déséquilibre maintenu, vis-à-vis de la fraude, entre un appareil répressif essentiellement administratif et un appareil judiciaire, **encore dans l'enfance**.".

Fig. 9.8 Result returned by TRADOIT to the query “*is still in its infancy*”

is the Government of Canada’s terminology and linguistic data bank. These two kinds of resources are searched at the same time when a query is submitted. In the displayed example of Fig. 9.8, information was only found in the parallel corpora but the opposite situation may happen. For example, the query “*is hard of hearings*” does not have a single hit in the TM but translations of the related expressions “*person who is hard of hearing*” and “*employee who is hard of hearing*” are shown from the TERMIUM PLUS bank.

The system interface displays on the left-hand side various information (Fig. 9.8). The “Grouped Translations” section displays statistics on the different translations identified in the TM with their frequency. The “1001 Forms” section lists the various forms found for the searched expression, i.e., forms that differ on capitalization or inflection (conjugation or plural when the canonical form of verbs and nouns is searched). The “Sources” section allows users to refine their search by filtering results based on the corpus origin. The right-hand side displays TM sentence pairs in table format and highlights source language hits and their target language equivalent. The translation of the query is automatically identified inside the sentence pairs using a method that is not described. For each use example, the source is specified and the hyperlink “See bitext” allows users to access the sentence in the context of

the source document. Two thumb pictures are also displayed for each sentence pair. This enables users to provide a feedback on the quality of the identified translation.

The concordancer provides additional features for query processing. For example, the system is able to suggest searches that may be more successful in case the user gets too few results. This process of suggesting alternate searches also includes the use of several operators: (1) a “+” sign added at the end of a word allows search on the various inflected forms of this word; (2) a “?” sign indicates that a word is missing in the typed expression at a given position; and (3) a “*” sign means that zero or one word can occur at a given position of the expression. Since these various signs can be combined, this allows the user, for example, to search together “*write a letter*,” “*writing letters*,” or “*write detailed letters*” with the query “*write+ ? letter+.*”

Conclusion

In this work, we have studied the problem of identifying translations of idiomatic expressions in both English and French, with a brand new version of the bilingual concordancer TRANSSEARCH. We showed in our experiments that a user who would query the system verbatim would often fail to find a match in the TM. As a result, some innovation is required in order to get good use of the system, such as utilizing the morphological (“+”) and the proximity (“..”) operators available in the query language recognized by the system. We automatized the querying process and conducted experiments that search entries of a phrase book inside a TM collected from the Canadian Hansards. These experiments showed that almost half of the 1.5 thousand idiomatic expressions queried to the system finally got a match in the TM, while a high proportion of the translations returned by the automated system were correct.

A comparison of the output generated by TRANSSEARCH with the GOOGLE TRANSLATE MT system showed that in spite of a relatively small size of its TM, our concordancer has a decent recall. And in fact, it even obtained higher recall values than GOOGLE TRANSLATE if several translations are considered, which is the typical *modus operandi* of our application.

Finally, we also discussed the functionalities of recent substantial bilingual concordancers. While an evaluation of those systems for identifying idiomatic expressions was out of the scope of the present study, the existence of several such applications shows the increasing popularity of advanced TM systems in the sphere of professional translators. In fact, the TRANSSEARCH application we used in this research is now released by Terminotix.¹⁶

¹⁶<http://www.tsrali3.com>.

Acknowledgements This work was funded by an NSERC grant in collaboration with Terminotix.¹⁷ We are indebted to Sandy Dincky, Fabienne Venant, and Neil Stewart who kindly participated to the annotation task.

References

- Anastasiou D (2008) Identification of idioms by machine translation: a hybrid research system vs. three commercial systems. In: Proceedings of EAMT, pp 12–20, Hamburg, Germany, 2008
- Bourdaillet J, Huet S, Langlais P, Lapalme G (2010) TransSearch: from a bilingual concordancer to a translation finder. *Mach Translat* 24(3–4):241–271
- Brown PF, Della Pietra SA, Della Pietra VJ, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. *Comput Ling* 19(2):2
- Callison-Burch C, Bannard C, Shroeder J (2005) A compact data structure for searchable translation memories. In: Proceedings of EAMT, pp 59–65, Budapest, Hungary, 2005
- Carpuat M, Diab M (2010) Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In: Proceedings of NAACL-HLT, pp 242–245, Los Angeles, CA, USA, 2010
- Fazly A, Cook P, Stevenson S (2009) Unsupervised type and token identification of idiomatic expressions. *Comput Ling* 35(1):61–103
- Fleiss JL, Levin B, Pai MC (2003) Statistical methods for rates and proportions, 3rd edn. Wiley, New York
- Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: Proceedings of HLT-NAACL, vol 1, pp 48–54, Edmonton, Canada, 2003
- Lambert P, Banchs R (2005) Data inferred multi-word expressions for statistical machine translation. In: Proceedings of MT summit, pp 396–403, Phuket, Thailand, 2005
- Langlais P (1997) A system to align complex bilingual corpora. Technical report, CTT, KTH, Stockholm, Sweden, 1997
- Macklovitch E, Lapalme G, Gotti F (2008) TransSearch: what are translators looking for? In: Proceedings of AMTA, pp 412–419, Waikiki, Hawaii, USA, 2008
- Macklovitch E, Simard M, Langlais P (2000) TransSearch: a free translation memory on the World Wide Web. In: Proceedings of LREC, pp 1201–1208, Athens, Greece, 2000
- McArthur T (ed) (1992) The Oxford companion to the english language. Oxford University Press, Oxford
- Mel'čuk I (1995) Idioms: structural and psychological perspectives, chapter phrasemes in language and phraseology in linguistics. Lawrence Erlbaum, Hillsdale, NJ, pp 167–232
- Mel'čuk I (2010) La phraséologie en langue, en dictionnaire et en TALN. In: Actes de la 17ème conférence sur le Traitement Automatique des Langues Naturelles (TALN), Montreal, Canada, 2010
- Névéol A, Ozdowska S (2006) Terminologie médicale bilingue anglais/français: usages clinique et législatif. *Glottopol* 8:5–21
- Piat J-B (2008) It's raining cats and dogs et autres expressions idiomatiques anglaises. J'ai lu, Libro, 2008
- Polguère A (2008) Lexicologie et sémantique lexicale: notions fondamentales, 2e édition Les Presses de l'Université de Montréal, Alain Polguère, Paramètres, p 356
- Ren Z, Lü Y, Cao J, Liu Q, Huang Y (2009) Improving statistical machine translation using domain bilingual multiword expressions. In: Proceedings of the ACL-IJCNLP workshop on multiword expressions, pp 47–54, Suntec, Singapore, 2009

¹⁷<http://www.terminotix.com>.

- Sag IA, Baldwin T, Bond F, Copestake A, Flickinger D (2002) Multiword expressions: a pain in the neck for NLP. In: Proceedings of CICLing, vol 2276 of Lecture Notes in Computer Science. Springer, Mexico City, pp 1–15
- Simard M (2003) Translation spotting for translation memories. In: Proceedings of the HLT-NAACL workshop on building and using parallel texts: data driven machine translation and beyond, vol 3, pp 65–72, Edmonton, Canada, 2003
- Takeuchi K, Kanehira T, Hilao K, Abekawa T, Kageura K (2007) Flexible automatic look-up of english idiom entries in dictionaries. In: Proceedings of MT summit, pp 451–458, Copenhagen, Denmark, 2007
- Véronis J, Langlais P (2000) Evaluation of parallel text alignment systems—The Arcade Project., Chap 19. Kluwer Academic, the Netherlands, pp 369–388
- Volk M (1998) The automatic translation of idioms: machine translation vs. translation memory systems. In: Weber N (ed) Machine translation: theory, applications, and evaluations: an assessment of the state-of-the-art Gardez! Verlag, St. Augustin, pp 167–192. <http://dl.acm.org/citation.cfm?id=328552&CFID=211791839&CFTOKEN=31638199>

Part III
**Innovative NLP Tools for Physicians,
Educators and Translators**

Chapter 10

Argumentation-Based Dialogue Systems for Medical Training

Adela Grando, Laura Moss, Gemma Bel-Enguix, M. Dolores Jiménez-López, and John Kinsella

Abstract Dialogue and argumentation have been applied to the field of Artificial Intelligence in the medical domain. arguEIRA (Grando et al., Argumentation-logic for explaining anomalous patient responses to treatments, 13th conference on Artificial Intelligence in Medicine (AIME 11). Springer, 35–44, 2011) is a system based on the ASPIC argumentation engine and is able to detect anomalous patient responses using flexible reasoning processes and logical argumentation. This paper introduces an extended arguEIRA with an argumentation-based dialogue system inspired by the system proposed by Parsons et al. (J Log Comput 13:347–376, 2003) and based on a variant of Dung’s calculus (Dung, Artif Intell 77:321–357, 1995). The aim is to achieve systems for medical training that provide

A. Grando (✉)

Division of Biomedical Informatics, San Diego Super Computer Center 2.93E,
University of California, San Diego, San Diego, CA, USA
e-mail: mgrando@ucsd.edu

L. Moss

Department of Computing Science, University of Aberdeen, Aberdeen, UK

Academic Unit of Anaesthesia, Pain, and Critical Care Medicine, School of Medicine,
University of Glasgow, Glasgow, UK
e-mail: laura.moss@glasgow.ac.uk

G. Bel-Enguix

Laboratoire d’Informatique Fondamentale, CNRS-Aix-Marseille University, Marseille, France
e-mail: gemma.belenguix@gmail.com

M.D. Jiménez-López

Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, Spain
e-mail: mariadolores.jimenez@urv.cat

J. Kinsella

Academic Unit of Anaesthesia, Pain, and Critical Care Medicine, School of Medicine,
University of Glasgow, Glasgow, UK
e-mail: john.kinsella@glasgow.ac.uk

human-like mechanisms for computer–clinician interaction, potentially enhancing the acceptance of the system’s explanations while changing the clinician’s behavior. Furthermore, we aim to provide clinicians with simple mechanisms to discover through the training process if the knowledge base used by the explanation system should be updated or corrected, potentially changing the training system’s behavior.

Introduction

arguEIRA (Grando et al. 2011) was developed as a system that can help clinicians detect an anomalous patient response to treatment using a flexible reasoning process based on logical argumentation. The argument schemes used in arguEIRA are instantiated at run time with information from an ICU domain knowledge base and patient datasets. arguEIRA evaluates, using the ASPIC (Argumentation Service Platform with Integrated Components) (Amgoud et al. 2006) argumentation framework, whether an observed response to a drug is different to the expected response. The results of arguEIRA are then presented to a clinician in the form of an argumentation graph, and an equivalent textual representation. In some cases this explanation was very long and complex. The initial evaluation of arguEIRA (Grando et al. 2011) indicated that clinicians would like to selectively explore and understand part of arguEIRA’s rational process in more detail, in particular for unusual medical cases or explanations they disagreed with.

Based on what we learnt from the evaluation of arguEIRA, we introduce in this paper an argument-based system for enhancing human computer dialogues in the medical domain. Our proposed extension of arguEIRA enables clinicians to interact with the tool. This facility will allow the clinicians to request detailed information on specific arguments of arguEIRA’s explanation, to disagree with nonconclusive clinical evidence upon which arguEIRA bases its explanations, and to understand how those disagreements can influence the explanations provided by arguEIRA.

This extension allows a clinician to “critique” the system; an important feature recognized since the origins of decision support systems (Davis and Buchanan 1977). We propose to do this by introducing a dialogue-based prototype of arguEIRA (Grando et al. 2011) that uses the ASPIC rules that are already part of arguEIRA’s inferencing system. The idea is that in the proposed arguEIRA, clinicians could introduce new knowledge at the end of a dialogue, for instance to add updated medical evidence. Even in those cases when arguEIRA does restrict a clinician’s right to update the knowledge base (while a consensus amongst senior clinicians can modify arguEIRA’s knowledge base, a junior clinician or single clinician is not allowed to update it), the tool could be used to document the rationality for clinicians’ disagreements with arguEIRA.

It is anticipated that this proposed extension will enhance acceptance by clinicians of the system as (a) the use of argumentation is a very human-like approach to reasoning enabling justifications that are in terms which are easier for a clinician to understand, and (b) the use of dialogue allows the clinician to control the level

of justification; not only does this provide a method of interaction that is again human-like, but it allows the justification to be tailored to an individual clinician.

The design suggested here is in a preliminary state. We present the main lines of development, and suggestions on how the implementation should be completed to reach the expected results.

The chapter is organized as follows: in “Motivation” we provide a motivation for developing our system, highlighting the main problems that medical applications of argumentation have encountered so far. “arguEIRA and ASPIC Argumentation-Based System” explains the background and basis of arguEIRA and ASPIC. “Proposed arguEIRA Dialogue System” introduces the proposed arguEIRA dialogue system. Finally, in “Final Remarks and Future Work,” we provide some discussion and future work.

Motivation

Adoption of Clinical Decision Support Systems

In many areas of medicine complex decisions are made by clinicians, often in uncertain conditions. Sophisticated Clinical Decision Systems (CDS), also known as expert or Knowledge-Based Systems (KBS), are now widespread in clinical practice and have been shown to be advantageous (Garg et al. 2005). Primary benefits of CDSs to clinicians can be grouped into three main categories:

1. Improved patient safety (e.g., through a reduction in human cognitive errors);
2. Improved quality of care (e.g., by reducing a clinician’s workload); and
3. Improved efficiency in healthcare delivery (e.g., by more cost-effective drug prescribing).

In discussing the types of CDS, a distinction can be made between:

1. Systems that remind clinicians of things they intend to do, e.g., request a diagnostic study, and
2. Systems that provide advice to clinicians, possibly persuading them to reconsider an intended action, e.g., change the prescribed drug for the one recommended by a clinical guideline.

These latter suggestions may involve additional diagnoses to consider or a change in medications from what the physician initially prescribed. Most CDS have focused on suggesting changes to clinicians’ actions (e.g., medication alerts) or changes to their beliefs, rather than reminding clinicians of their intentions (e.g., order sets). arguEIRA’s explanation of a patient’s anomalous responses to Intensive Care Unit (ICU) treatment intends to change a clinician’s knowledge base, and subsequently, possibly their actions.

Although there are many advantages to CDS, they have two known problems: (a) the *low acceptance and adoption* of these systems in the medical domain, (b) the *maintenance of the knowledge* embedded within the CDS.

The problem of *low acceptance and adoption* of these systems in the medical domain is well known (Eslami et al. 2007; Isaac et al. 2009; Tierney et al. 2003). It is generally acknowledged that (amongst other things) it is desirable for the acceptance of a decision support system that the system can explain its conclusions (Ye and Johnson 1995). The medical domain is particularly high risk and studies have shown that users of clinical decision support systems need to be able to see an explanation of a system's reasoning process to be confident in its conclusions (Walton 2006; Berry et al. 2005). Studies of the effectiveness of CDS have shown that it is easier to remind clinicians of what they already intend to do, rather than to get them to change their plans or beliefs (Tierney et al. 2003; Teich et al. 2000). One possible option to improve adoption of CDS is to allow the user to view the information on demand rather than presenting it automatically. Indeed, studies involving ICU staff have shown that an important function of technology is to reinforce the decision making of junior staff. Junior doctors and nursing staff often consult a computerized system as a way of finding supporting evidence for the decisions they make (Alberdi et al. 2000). The evidence shows that when the users seek out CDS information themselves, they are less likely to override it than when it is automatically presented to them; however, they choose to access the information very infrequently (Tamblyn et al. 2008). Those findings are consistent with ours. In our original implementation presented in Grando et al. (2011), we provided the user with natural language explanations of the patient's anomalous response to treatment. Our evaluation of the tool showed that clinicians preferred to access this information on demand.

Another problem directly influencing the adoption of CDS is the *maintenance of the knowledge embedded within the CDS*. If advice is based on obsolete knowledge, it is no longer useful. Given that medical knowledge is constantly being updated this can become a time consuming and expensive process. Knowledge should be actualized on two different levels: domain knowledge (for instance new drugs and treatments), and procedural, rule-based knowledge (for instance guideline-based recommendations). For the refinement of the first type of knowledge, it is possible to utilize commercial knowledge bases that provide frequent updates, for instance Medi-Span (<http://www.medispans.com>). For the second type of knowledge, initiatives such as the Clinical Decision Support Consortium (CDSC) (Agency for Healthcare Research and Quality 2009) and organizations such as NICE (<http://www.nice.org.uk>) and SIGN (<http://www.sign.ac.uk>) maintain repositories of clinical guidelines. The disadvantage of the mentioned knowledge maintenance approaches is that they are central, and obtaining consensus as to what should be included in a centralized system can be a challenge.

Using Argumentation-Based Dialogues

To resolve some of the challenges described in the previous section and to allow clinicians to interact with arguEIRA we propose an argumentation-based dialogue extension to arguEIRA.

Argumentation logic (Chesnevar et al. 2000) has evolved over the past three decades as a formal framework for modeling human collaborative deliberations, interchanging arguments in favor or against some conclusion based on potentially incomplete or inconsistent information. In argumentation theory, the intention is to determine if a particular proposition follows from certain assumptions, even if some of these assumptions are disproved by other assumptions. Furthermore, arguments can have relative strengths, which provide a very human-like approach to reasoning (Fox et al. 2007).

From its origins in classical philosophy, argumentation theory has evolved to become an important research field in Artificial Intelligence (AI). The relationship between computer science and the area of philosophy that focuses on arguments has led to the emergence of a new interdisciplinary field that, according to Grasso et al. (2010), can be called *computational dialectics*, *argumentation technology*, or *argument-based computing*.

So far, the interaction between AI and medicine has been very fruitful. Argumentation has been investigated as a tool for changing health-related behaviors, tailoring explanations, advising patients on treatment regimes, as well as for designing agents working in cooperation within the healthcare team (see Grasso et al. 2000; Green 2005; Fox et al. 2001; Jenicek and Hitchcock 2005). Another interesting contribution to healthcare argumentation models was the CARREL project (Vázquez-Salceda et al. 2003) developed at the Technical University of Catalonia. This framework uses argumentation theory to support the management of tissue and organ transplantation.

Being able to design systems that can change the behavior of the clinician has motivated the dialogue-based interface that we propose in this chapter.

Changes in a clinician's behavior will perhaps occur when more intelligent systems are designed, capable of challenging a clinician's opinion, and discussing diagnosis and medical results. Such systems directly refer, according to Walton and Krabbe (1995), to persuasion dialogues. For this work we have taken into account several recent contributions on the topic. Parsons et al. (2003) designed strategies to "challenge" the arguments of the other agents. Persuasion models have already been applied to the medical domain. Tolchinsky et al. (2006) took the approach of the classical "critical questions" (Walton 1995) for optimizing the CARREL system of transplant management. Finally, there exist applications for medical diagnosis (Rahati and Kabanza 2010) based on persuasion dialogues. In the last system, diagnosis is considered as an *argumentation between the student and the intelligent tutoring system*.

arguEIRA and ASPIC Argumentation-Based System

Anomalous scenarios play a key role in knowledge discovery; Kuhn (1962) defines an anomaly as a violation of the “*paradigm-induced expectations that govern normal science*.” Anomalies are of interest as they often point to the inadequacy of a currently held theory and require refinement of the related theory; consequently this can provide the impetus for the discovery of further domain knowledge.

EIRA (Explaining, Inferencing, and Reasoning about Anomalies) (Moss 2010) helps users to detect anomalous patient responses to treatment in the ICU domain and provides decision support for clinicians by generating explanations for the anomalies. EIRA was based on complex algorithms that reflect the problem solving methods used by ICU clinicians in detecting and resolving anomalies. Whilst EIRA has proved to be very accurate (Moss et al. 2010), it lacked a justification system that could make explicit, in a user-friendly way, the complex rationality behind the algorithms.

arguEIRA (Grando et al. 2011) is a revised system where we abstract EIRA’s algorithms to detect an anomalous patient response to use a more flexible reasoning process based on logical argumentation.

Our mapping of EIRA’s algorithms into arguEIRA’s arguments required building an argumentation knowledge base differentiating between conclusive knowledge (clinical facts like the measurement of patient’s vital signs), and nonconclusive knowledge (like clinical reasoning).

In arguEIRA we have adopted Dung’s calculus of opposition to identify and express argument exchanges made by ICU clinicians. The obtained arguments and interactions (attacking, supporting arguments) were modeled using ASPIC (Amgoud et al. 2006). The ASPIC project was concerned with developing foundations for a general theory of argumentation in the context of autonomous agents. The resulting software components in ASPIC implement computational argumentation capabilities in four areas: *inference*, *decision making*, *dialogue*, and *learning*. For adopting the ASPIC argumentation-based inference system, arguEIRA was provided with a *knowledge base* and relevant *argument interactions*.

Knowledge base: it contains a knowledge base of facts and a set of rules. Facts are defined as [optional fact name] $\alpha.n$ and rules are of the form [optional rule name] $\beta < \alpha_1, \dots, \alpha_t. n$, where $\beta, \alpha, \alpha_1, \dots, \alpha_t$ are expressed as first order literals and their negations and $0 < n \leq 1$ is numeric support for the rule. Facts can have support $0 < n \leq 1$, strict rules have support 1.0 and defeasible rules can be assigned support $0 < n < 1$. The greater the numeric support, the more confidence there is in the validity of the rule. Here we follow the convention for qualitative support adopted in the Tallis decision support system (<http://www.cossac.org/tallis>): (++) for conclusive arguments, (+) for positive support, (-/+) for neutral support, (-) for negative support, and (--) for exclusive support; to do this conclusive arguments are assigned 1, positive support corresponds to 0.7, neutral support is equivalent to 0.5, negative support is assigned 0.3, and exclusive support receives 0. We have assigned to all nonconclusive knowledge a medium or neutral support. In the future, arguEIRA could assign support to arguments on the basis of evidence gathered from previously recorded deliberations.

To define our knowledge base we adopted the notion of an anomaly used in the original EIRA: an anomaly exists if a clinician has an acceptable argument for an expected effect, but there is evidence that the expected effects did not happen (arguments [a1] [a2], *defined below*). According to arguEIRA, if the clinicians administer a drug at time t_1 with a known expected effect, then they can confidently expect to see the drug's effect at a time t_2 where $t_2 > t_1$ ([a3]). We also added to the knowledge base the facts [a4][a5] to state that the observation of no increase in an attribute value is different from an observed increase. For this scenario we know that at time t_1 , patient p has severe sepsis, is vasodilated, and has been administered noradrenaline (norepinephrine), as stated by the facts [a6] to [a9]. We also know that at time t_2 , where $t_2 > t_1$, there is an increase in the patient's cardiac output, expressed in [a10][a11] and a highly expected effect of noradrenaline is not to increase the patient's cardiac output ([a12]). Finally, the most probable outcome is that the noradrenaline is not producing the expected effect for patient p at time t_1 . Below we show the arguEIRA formalization of the above arguments:

Rules:

```
[a1]anomaly <- expected(Patient, Attribute,
Expected, t2), observed(Patient, Attribute,
Observed, t2), different(Expected, Observed) 1.0.
[a2]anomaly <- expected(Patient, Attribute,
Expected, t2),observed(Patient, Attribute,
Observed, t2), different(Expected, Observed) 1.0.
[a3]expected(Patient, Attrib, Expect,t2) <-
administered(Patient, D, t1), expected_effect(D, Attrib,
Expect), greater(t2,t1) 0.7.
```

Facts:

```
[a4]different(no_increase,increase) 1.0.
[a5]different(increase, no_increase) 1.0.
[a6]patient(p) 1.0.
[a7]severe_sepsis(p,t1) 1.0.
[a8]vasodilated(p,t1) 1.0.
[a9]administered(p, noradrenaline, t1) 1.0.
[a10]observed(p, cardiac_output, increase, t2) 1.0
[a11]greater(t2,t1) 1.0
[a12]expected_effect(noradrenaline,cardiac_output,
no_increase) 0.7
```

Argument interactions: three different types of attack relations can be defined: *rebutting*, *restricted rebutting*, and *undercutting*. For instance, the literals $\sim a$ 0.3 and a 0.5 are both valid and their associated arguments rebut (or contradict) each other. Similarly, an argument formed from the fact a . and the rule $b \leftarrow a$ 0.9. rebuts an argument formed from the fact $\sim b$ 0.4. Strict arguments cannot be rebutted. Under restricted rebutting, an argument whose top rule is strict cannot be rebutted by an argument whose top rule is defeasible. Rules can be named and undercut (by a

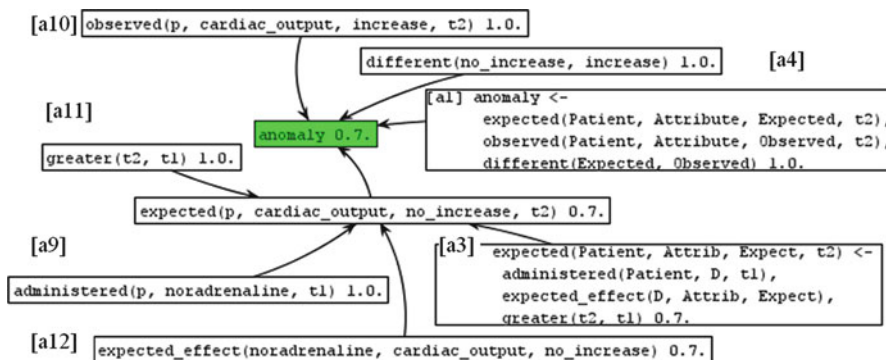


Fig. 10.1 Argumentation graph returned by the ASPIC engine when the user queries if the patient's response to treatment is anomalous

counter argument) by writing a fact or rule whose head is the contradiction of that name; if argument A undercuts (i.e., is a counter argument to) argument B, then A claims that some rule in B is not applicable. For instance, if noradrenaline is not working (i.e., not having the expected effects) then the defeasible argument that we named **a3** could be undercut:

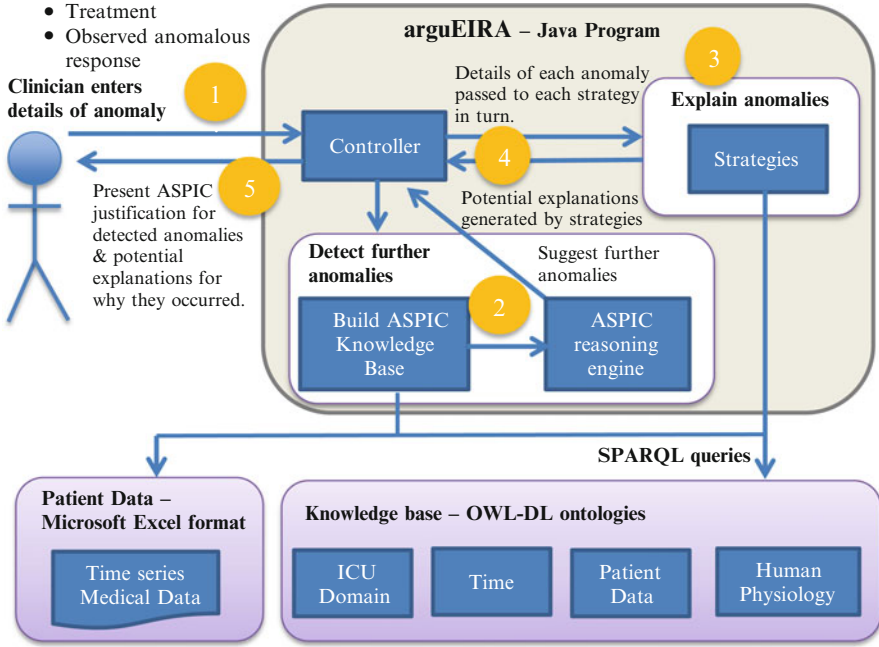
```
[a3] <- working(p,noradrenaline,t1),
expected_effect(noradrenaline, cardiac_output,
non_increase) 0.7.
```

At run time, based on the interacting arguments and Dung's calculus of opposition (Dung 1995), arguEIRA determines the status of the arguments based on Dung's grounded semantics: if argument A is attacked by some argument B, then A can be reinstated as an acceptable argument if its attacker B is itself attacked. When the query *anomaly* is introduced, the results of the inferences performed by the ASPIC engine are shown as an argumentation graph (Fig. 10.1). Here we have set the ASPIC engine to use the weakest link valuation, which assigns the support for the main argument as the minimum support over all of its subarguments. In the diagram, arrows represent inference, shaded boxes indicate an undefeated main argument, and nonshaded boxes indicate an unevaluated argument (usually a subargument). As found in the clinicians' dialogue transcript, the ASPIC engine has detected that the patient's increase in cardiac output is anomalous. One explanation, as shown in Fig. 10.1, is that after administering noradrenaline to a patient (according to the known expected effects of the noradrenaline drug in this context) the patient's cardiac output should decrease.

arguEIRA extends the original EIRA program by providing a justification for EIRA's detection of anomalies using the ASPIC argumentation framework. arguEIRA comprises the following components from EIRA: a knowledge base consisting of several instantiated OWL ontologies (<http://www.w3.org/2004/OWL/>)

Anomaly Details :

- Patient ID
- Time point in session
- Treatment
- Observed anomalous response

**Fig. 10.2** arguEIRA overview

and a Java program implementing strategies extracted from domain experts' protocols to provide explanations for the anomalies.

However, as shown in Fig. 10.2, EIRA's evaluation of whether the observed response to a drug is different to the expected response has been replaced by the ASPIC argumentation framework (through ASPIC java libraries). To create an ASPIC knowledge base, schemes have been implemented in arguEIRA based on previous modeling of clinician argumentation (Grando et al. 2011). The schemes are instantiated at run-time with information from the patient dataset and domain ontologies. For example, the scheme `administered(p, [drug]"", t1) 1.0.` is instantiated at run time as `administered(patient99, adrenaline, (12/08/2010 3:00:00)) 1.0.`

Below we provide several examples of schemes introduced in arguEIRA:

- `[scheme9] administered(p, [drug], t1) 1.0.`
- `[scheme10] observed(p, [observed parameter], [observed effect], t2) 1.0.`
- `[scheme12] expected_effect([drug], [expected effect parameter], [expected effect]) 0.7.`

| Anomaly Explanation |
|--|
| EIRA - Possible explanation for anomalies |
| Results will display here.... |
| Explaining why the anomalous effect may have occurred..... |
| Anomaly being investigated: |
| Treatment: Adrenaline Response: DecreaseMAP |
| Identifying if the patient is getting significantly better..... |
| Identifying if the patient is recovering from an illness... |
| Identifying if a conditional drug effect has been met.... |
| The anomalous effect may be explained by the patient's condition improving |

Fig. 10.3 EIRA-generated explanations for why an anomaly may have occurred

- `[scheme1]anomaly <- expected(Patient, Physiological-Effect, Expected, t2), observed(Patient, Physiological-Effect, Observed, t2), different(Expected, Observed)1.0.`
- `[scheme2]anomaly <- expected(Patient, Physiological-Effect, Expected, t2), observed(Patient, Physiological-Effect, Observed, t2), different(Expected, Observed)1.0.`
- `[scheme3] expected(Patient, Attrib, Expect, t2) <- administered(Patient, D, t1), expected_effect(D, Attrib, Expect), greater(t2,t1)0.7`

Once the knowledge base has been completed, the ASPIC argumentation engine determines the state of the arguments and whether an anomaly has occurred. `arguEIRA` then proceeds to generate context-dependent hypotheses for why each of the detected anomalies may have occurred. The hypotheses are generated by proceeding with each of the implemented strategies (algorithms). If appropriate, explanations for the anomalies are generated. These hypotheses are presented to the user (Fig. 10.3). In addition, an argumentation graph and an equivalent textual representation are presented to the user to justify why `arguEIRA` detected the additional anomalies (Fig. 10.4). The use of argumentation schemes allows for the separation of the argument's form from its content, as proposed in reference (Gordon and Walton 2009). As in (Williams 2008) the content of the arguments is generated from queries of a domain ontology used as a reference source.

Proposed `arguEIRA` Dialogue System

In this section we propose to extend `arguEIRA` with an argumentation-based dialogue system inspired by an existing agent-based dialogue system proposed by Parsons et al. (2003). As we explained in “`arguEIRA` and ASPIC Argumentation-

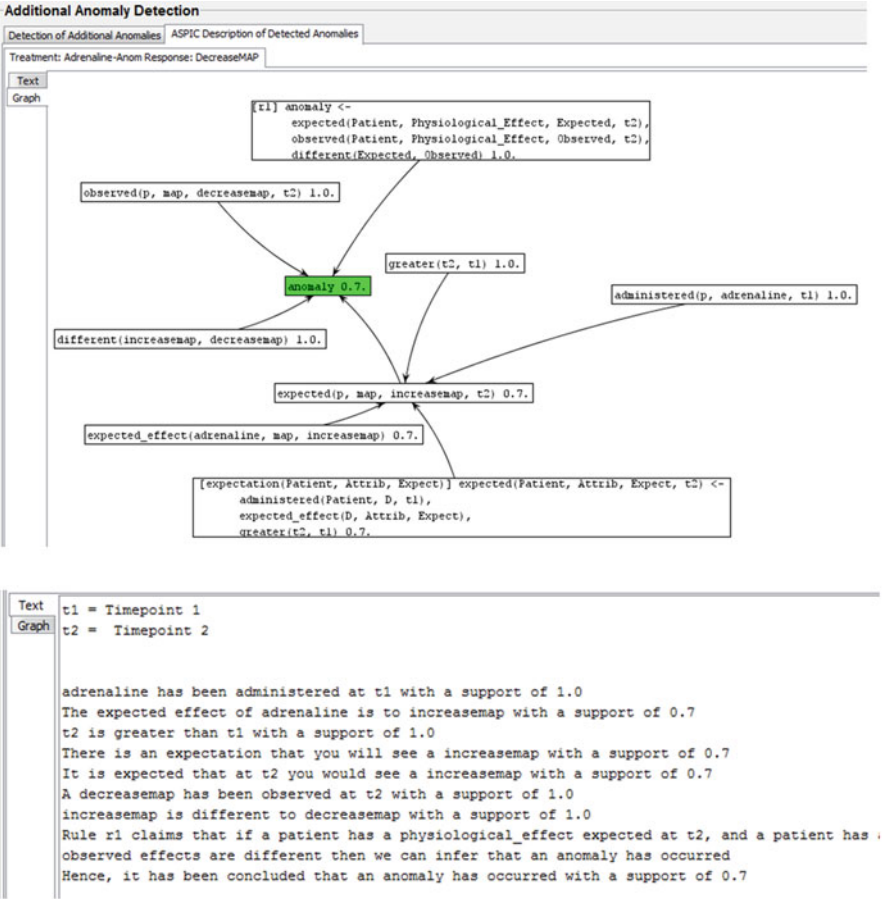


Fig. 10.4 ASPIC inferences for the detection of an anomaly

Based System” arguEIRA is based, similarly to (Parsons et al. 2003), on a variant of Dung’s calculus of opposition to determine the acceptable (winning) arguments (Dung 1995).

In our dialogue-based extension of arguEIRA we reuse the already adopted ASPIC argumentation system, this time for reasoning during the dialogue-based interaction between the tool and the clinician. arguEIRA will decide what she knows by determining for which propositions she has acceptable arguments. arguEIRA will then state propositions for which she has acceptable arguments, and accept propositions put forward by the clinician if she finds that the clinician’s arguments are acceptable. While arguEIRA has its own knowledge base, containing her beliefs, she will be also able to construct the set of commitments made in the dialogue.

With this background we can explain the speaker’s assertion and acceptance attitude, the locutions interpretation, and the dialogue protocol that we will use.

arguEIRA assertion and acceptance attitudes:

arguEIRA will adopt the ASPIC engine's assertion and acceptance attitudes, asserting propositions for which it can construct acceptable arguments. According to Parsons et al. (2003) this is considered a thoughtful assertion attitude and a skeptical acceptance attitude. The ASPIC engine will determine the status of the arguments based on Dung's grounded semantics: if argument A is attacked by some argument B, then A can be reinstated as an acceptable argument if its attacker B is itself attacked.

Clinician's assertion and acceptance attitudes:

Obviously, in the case of the clinician we can make no assumption about his acceptance or assertion attitude. The clinician can always request additional explanations, challenging arguEIRA. But we have restricted the clinician to disagree with arguEIRA only on clinical evidence that is nonconclusive (has support smaller than 1.0).

arguEIRA locutions:

We will define arguEIRA locutions in terms of preconditions and postconditions. Such that for a locution to be uttered, its precondition should be satisfied, and once the locution has been uttered the state described by the postcondition is satisfied. For obvious reasons we do not define preconditions and postconditions for the clinicians interacting with arguEIRA. We have no influence on the reasons why a clinician utters a locution, neither on the effect that a locution has in his mental state (which in arguEIRA is the commitment store).

The considered locutions are:

Assert(p, n), where p is a first order literal and $0 < n \leq 1$ is the numeric support

Precondition = arguEIRA's assertion condition

Postcondition = p, n is added to arguEIRA's commitment store

Assert(s_1, n_1), ..., Assert(s_t, n_t), where s_1, \dots, s_t are first order literals and n_1, \dots, n_t are the numeric support

Precondition = arguEIRA's assertion condition

Postcondition = $s_1, n_1 \dots, s_t, n_t$ are added to arguEIRA's commitment store

Accept(p, n), where p is a first order literal and n is the numeric support

Precondition = arguEIRA's acceptance condition

Postcondition = p, n is added to arguEIRA's commitment store

Accept(s_1, n_1), ..., Accept(s_t, n_t), where s_1, \dots, s_t are first order literals and n_1, \dots, n_t are the numeric support

Precondition = arguEIRA's acceptance condition

Postcondition = $s_1, n_1, \dots, s_t, n_t$ is added to arguEIRA's commitment store

A challenge is a means of making the other player explicitly state the argument supporting a proposition.

Challenge(p, n), where p is a first order literal and n is the numeric support.

Precondition = none

Postcondition = none

Dialogue protocol

In our dialogue system, arguEIRA will try to persuade the clinician to adopt her detection of patient's anomalous response to ICU treatment. The clinician could participate in the dialogue to understand arguEIRA's justification, and he could potentially disagree with arguEIRA. The clinician can always request additional explanations challenging arguEIRA, and could also disagree with (attack) arguEIRA's nonconclusive arguments.

We consider that A is initially arguEIRA trying to persuade clinician (B) with p equal to anomalous or not anomalous, and n the support (patient's reaction to treatment is anomalous or not anomalous). Therefore the dialogue protocol is as follows:

1. A asserts(p,n)
2. B accepts(p,n) if its acceptance attitude allows, if not B asserts(p,q) if it is allowed to, or otherwise challenges(p,n).
3. If B asserts(p,q), then goto 2 with the roles reversed and not p in place of p and q in place of n.
4. If B has challenged, then:
 - (a) A asserts(s₁,n₁), asserts(s₂,n₂), ..., asserts(s_t,n_t)
 - (b) Goto 2 for each s_i,n_i in turn.

Dialogue outcome:

If, at the end of the dialogue arguEIRA accepts the clinician's arguments and the user is a senior clinician the tool will incorporate into its knowledge base the clinician's winning argument with a support of 0.7. In contrast, if the clinician is junior, any interaction with arguEIRA is considered a training session and at the end of the dialogue the clinician's argument is not incorporated into arguEIRA's knowledge base.

Dialogue Example

The EIRA algorithms used to determine if a patient's response to treatment is anomalous have been based on preliminary studies during which ICU clinicians were asked to identify anomalous responses to treatment and subsequently, if possible, provide explanations for the anomalies. For further details see reference (Moss 2010). Part of these preliminary studies consisted of sessions with ICU clinicians from Glasgow Royal Infirmary to understand how counter arguments could be generated against a suspected anomaly. For example, one clinician

(clinician 1) believed that the patient's reaction to treatment was anomalous: the patient's cardiac output¹ and index² had improved when the drug noradrenaline³ was increased.

Clinician 1 *"...but then we obviously do something because the cardiac output and the cardiac index get a bit better and the thing that we seem to have done is put the noradrenaline up to a high dose, but that isn't necessarily quite what we would expect from a high dose of noradrenaline"*

Whilst two other clinicians (clinicians 2 and 3) believed that the patient's reaction is not anomalous. Clinician 2 has provided two explanations for the anomaly. The first is that the patient's overall condition has improved (thereby increasing the cardiac output and index); the second is that the patient has sepsis⁴ and previously had a myocardial event (i.e., the myocardial event caused the cardiac index to be reduced and then it subsequently improved).

Clinician 2 *"So, the patient may have changed, there may have been more sepsis perhaps which causes systemic vascular resistance to fall or maybe the patient was just starting to get better. I think the patient is considerably better by the end of day 32, I think that's what happened, the patient's underlying condition has changed and the patient has just improved for one reason or another because they are a lot better at the end of day 32 that they were at the end of day 31"*

"...I mean it may have been that there has been some event you see, the combination of sepsis and after the myocardial infarction because they had a low cardiac index and a high systemic vascular resistance. So it's possible that they had a cardiac event, the explanation would be that sometime, a little bit previously, perhaps at the end of day 30 into 31 they had a cardiac event and 24, 48 hours [after] they had recovered from this, that's a possible explanation in somebody who has got sepsis"

Additionally, Clinician 3 has suggested that when noradrenaline is given at a large dose, it can act to increase the force of each contraction of the heart, consequently increasing the cardiac output.

Clinician 3 *"The only thing that I can think of is that noradrenaline is actually an inotrope. In a low dose, it tends to be a vasoconstrictor, in higher doses it's an*

¹Cardiac output is the volume of blood being pumped by the heart in a particular minute.

²Cardiac index is a parameter which relates a patient's cardiac output to their body surface area. It enables a clinician to relate the performance of the patient's heart to the patient's body size.

³Noradrenaline is generally given to increase a patient's blood pressure. The blood pressure is increased because the noradrenaline constricts the patient's blood vessels.

⁴Sepsis is a life-threatening condition caused by an overactive immune system which overwhelms normal processes in a patient. Widespread inflammation and small blood clots form, which can cause obstruction to organs and can quickly, lead to organ failure. Sepsis commonly stems from a bacterial infection. A myocardial event is often something which causes damage to the muscle of the heart, e.g., heart attack.

inotrope. So it might just be that, that dose for that particular patient is enough to, as well as causing tightening, is enough to cause an increase force of contraction as well"

Below we show how to simulate the proposed extension of arguEIRA using the previously explained dialogue:

1. Initially arguEIRA adopts the place of clinicians 2 and 3 stating that "The cardiac index and cardiac output have improved but this is not anomalous the patient's reaction to treatment is not anomalous," i.e., arguEIRA asserts(`anomaly`,0.5)

This assertion is supported by arguEIRA's ASPIC-based argument system.

2. Clinician 1 can ask for a reason "Why do you think that it is not anomalous?," i.e., Clinician 1 challenges(`anomaly`,0.5).
3. arguEIRA can then proceed through the strategies and present its explanations to Clinician 1, saying: "There are multiple potential explanations. First the cardiac index and cardiac output could have improved because the patient is overall getting better. Second the cardiac index and cardiac output could have been recovering from a myocardial event. Third, the dose of noradrenaline could be additionally acting as an inotrope and increasing the cardiac output and index." Asserts(`expected_effect(noradrenaline, cardiac_output, no_increaseCardiacOutput)`, 0.7.)
Asserts(`observed(p, cardiac_output, increaseCardiacOutput, t2)`, 1.0.)
Asserts(`different(increaseCardiacOutput, no_increaseCardiacOutput)`, 1.0.)
Asserts(`overallImprovement`, 0.5.)
asserts(`recovering_from_condition (cardiogenic_shock)`, 0.5.)
asserts(`conditionalDrugExplanation`, 0.5.)
4. Clinician 1 agrees that the patient is recovering from a myocardial event (e.g., cardiogenic shock) and that noradrenaline could be acting as an inotrope. Accepts(`recovering_from_condition (cardiogenic_shock)`, 0.5.)
Accepts(`conditionalDrugEffectExplanation`, 0.5.)
5. But Clinician 1 challenges the argument that the patient has improved overall.
"Can you explain why the patient has improved overall?"
Challenges(`overallImprovement`, 0.5.)
6. In response arguEIRA explains that "it has been observed that the patient's FiO₂, temperature, and heart rate have improved, therefore the patient has improved."
Asserts(`improvement(fio2)`, 0.5.)
Asserts(`improvement(temperature)`, 0.5.)
Asserts(`improvement(heart rate)`, 0.5.)
7. The clinician can ask "Why is there an improvement in heart rate?"
Challenges(`Improvement(heart rate)`, 0.5.)

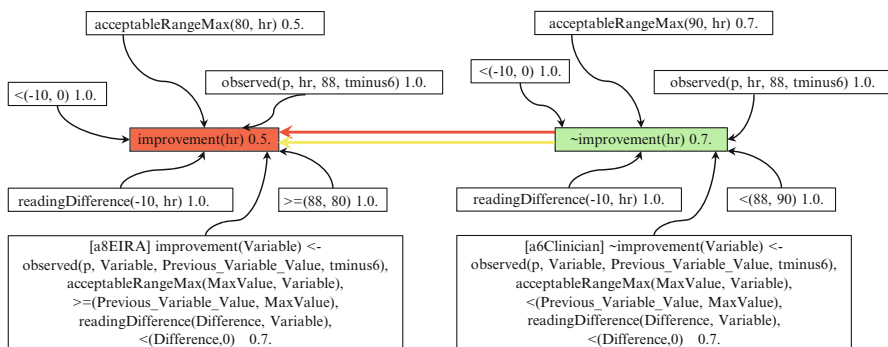


Fig. 10.5 ASPIC visualization of the argument interaction which concludes that there has been no improvement in heart rate

8. arguEIRA explains that “the acceptable range for heart rate is between 50 and 80 beats per minute. The patient’s heart rate improved because their heart rate started off higher than the maximum acceptable range at 88 and then decreased to 78, hence it improved”

```
Asserts(observed(p, heart rate, 88, tminus6), 1.0.)
Asserts(acceptableRangeMax(80, heart rate), 0.5.)
Asserts(>=(88, 80), 1.0.)
Asserts(readingDifference(-10, heart rate), 1.0.)
Asserts(<(-10, 0), 1.0.)
```

9. Because `acceptableRangeMax(80, heart rate)0.5` is not a conclusive clinical fact, then clinician 1 can attack arguEIRA’s argument by saying “I do not agree that the acceptable range for heart rate is 80.” Therefore clinician 1 can assert (`acceptableRangeMax(80, heart rate), 0.5`)
10. Then arguEIRA challenges the clinician’s statement. “Why do you not agree that the acceptable range for heart rate is 80?” *Challenges* (`acceptableRangeMax(80, heart rate), 0.5`) .
11. The clinician asserts “In fact I believe that the acceptable range for heart rate is 90,” asserts (`acceptableRangeMax(90, heart rate), 0.7`)
12. Finally arguEIRA accepts the clinician’s assertion and the dialogue finishes (Fig. 10.5). Accepts (`acceptableRangeMax(90, heart rate), 0.7`)

This now means that both the clinician and arguEIRA agree that the patient has not improved overall. This argument interaction is visualized in Fig. 10.6.

arguEIRA will then rerun the hypothesis generation and while it stills believes that the patient’s reaction to treatment is not anomalous, after the discussion with the clinician the overall patient improvement is not a feasible explanation anymore. Basically arguEIRA thinks that the patient’s response to treatment is anomalous based on the following arguments:

```
recovering_from_condition(cardiogenic_shock) 0.5
conditionalDrugEffectExplanation 0.5
```

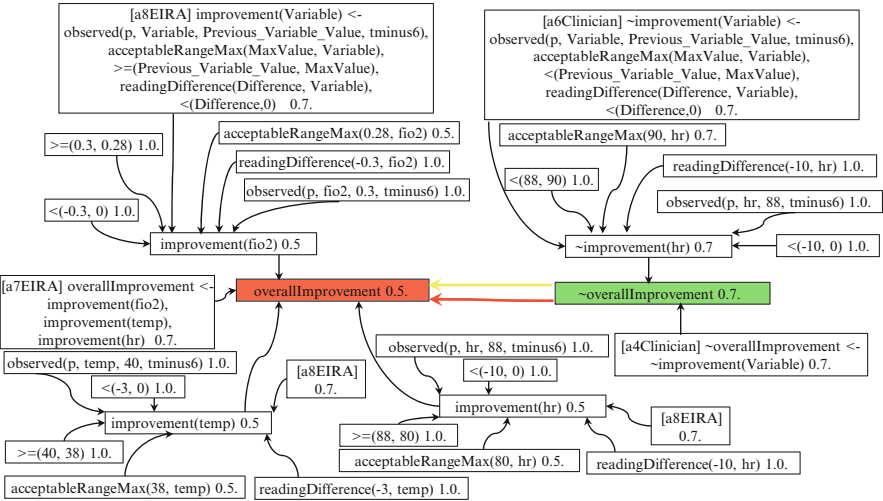



Fig. 10.6 Argument interaction resulting from the dialogue-based interaction of arguments between a clinician and arguEIRA

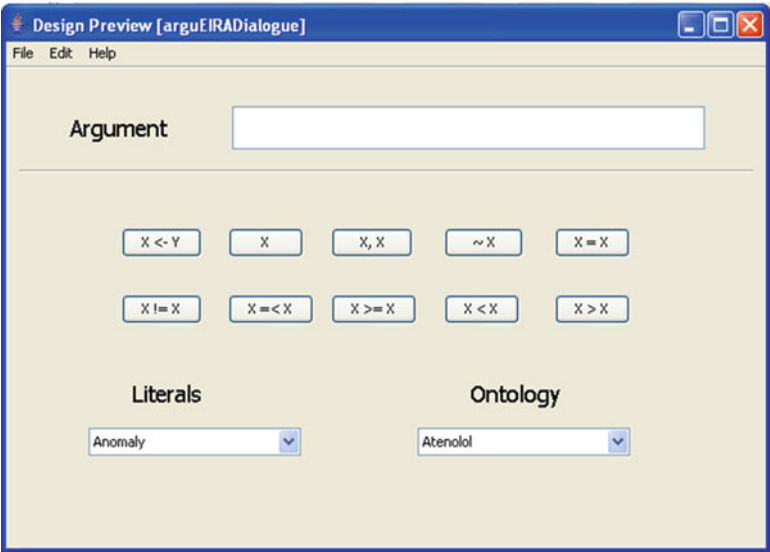


Fig. 10.7 Interface to allow clinicians to interact with arguEIRA

Our prototype of arguEIRA has a simple user interface which allows the clinicians to specify arguments when they interact with arguEIRA. The interface shown in Fig. 10.7 provides a user-friendly mechanism to specify well-formed arguments in terms of arguEIRA’s existing knowledge base, following ASPIC’s syntax. Our interface is inspired by a similar interface provided by Tallis

(<http://www.cossac.org/tallis>) to create medical arguments. Tallis is a software tool to support the management of medical procedures and argumentation-based clinical decision making at the point of care.

Final Remarks and Future Work

In the future the system must be fully implemented to be tested as a training tool by ICU clinicians. An initial retrospective comparative study between arguEIRA and the proposed arguEIRA extension could show: (a) if clinicians' initial concern (Grando et al. 2011) of being able to access this information provided by arguEIRA on demand was addressed with the dialogue-based explanation system, and (b) if clinicians find the new dialogue-based training environment more user-friendly than previous text-based explanations.

In addition, a more in-depth study should be carried out to understand the practical effects, in highly sensitive contexts such as the ICU, of the adoption of simple mechanisms for updating and refining a CDS's knowledge base. In CARREL+ (Vázquez-Salceda et al. 2003) this issue was resolved by checking the consistency of the user's arguments with respect to the organ transplantation guidelines. Similarly, for the planned arguEIRA extension we will explore the implementation of a mechanism to periodically curate the refined knowledge. This process could potentially be very beneficial for the medical staff and could help them to identify, interpret, and adopt new medical evidence.

Acknowledgements Prof. Derek Sleeman (Department of Computing Science, University of Aberdeen). Staff of the ICU, Glasgow Royal Infirmary. Kathryn Henderson and Jennifer McCallum (CareVue Project, Glasgow Royal Infirmary). Matthew South (Robotics Research Group, University of Oxford). This work was an extension of the routine audit process in Glasgow Royal Infirmary's ICU; requirements for further Ethical Committee Approval have been waived.

References

- Agency for Healthcare Research and Quality (2009) Clinical decision support initiative. <http://healthit.ahrq.gov/portal/server.pt?open=512&objID=654&&PageID=13665&mode=2&inhiiuserid=3882&cached=true>. Accessed 11 Jan 2009
- Alberdi E, Gilhooly K, Hunter J, Logie R, Lyon A, McIntosh N, Reiss J (2000) Computerisation and decision making in neonatal intensive care: a cognitive engineering investigation. *J Clin Monit Comput* 16(2):85–94
- Amgoud L, Bodenstaff L, Caminada M, McBurney P, Parsons S, Prakken H, Veenen J, Brrrdeijk G (2006) Report on formal argumentation system for ASPIC Project. Technical report UCLS-07-005, University of Liverpool. <http://www.csc.liv.ac.uk/research/techreports/techreports.html>
- Berry D, Gillie DT, Banbury S (2005) What do patients want to know: an empirical approach to explanation generation and validation. *Expert Syst Appl* 8:419–428

- Chesnevar C, Maguitman A, Loui R (2000) Logical models of argument. *ACM Comput Surv* 32(4):337–383
- Davis R, Buchanan BG (1977), Meta-level knowledge: overview and applications. In: *Proceedings of the 5th international joint conference on Artificial Intelligence*, 1977
- Dung PM (1995) On the acceptability of arguments and its fundamental role in nonmonotonic reason, logic programming, and N-person games. *Artif Intell* 77:321–357
- Eslami S, Abu-Hanna A, de Keizer NF (2007) Evaluation of outpatient computerized physician medication order entry systems: a systematic review. *J Am Med Inform Assoc* 14(4):400–406
- Fox J, Glasspool D, Bury J (2001) Quantitative and qualitative approaches to reasoning under uncertainty in medical decision making. In: Quaglini S, Barahona P, Andreasson S (eds) *Proceedings of the 8th conference on Artificial Intelligence in Medicine in Europe (AIME)*. Springer, Berlin, pp 272–282
- Fox J, Glasspool D, Grecu D, Modgil S, South M, Patkar V (2007) Argumentation-based inference and decision making—a medical perspective. *IEEE Intell Syst* 22(6):34–41
- Garg A, Adhikari N, McDonald H (2005) Effects of computerized clinical decision support systems on practitioner performance and patient outcomes. A systematic review. *J Am Med Assoc* 293(10):1223–1238
- Gordon, TF, Walton D (2009) Legal reasoning with argument schemes. In: *12th international conference on Artificial Intelligence and Law*, 2009, pp 137–146
- Grando MA, Moss L, Glasspool D, Sleeman D, Sim M, Gilhooly C, Kinsella J (2011) Argumentation-logic for explaining anomalous patient responses to treatments. In: *13th conference on Artificial Intelligence in Medicine (AIME 11)*. Springer-Verlag, Berlin, Heidelberg, pp 35–44
- Grasso A, Cawsey A, Jones R (2000) Dialectical argumentation to solve conflicts in advice giving. A case study in the promotion of healthy nutrition. *Int J Hum Comput Stud* 53:1077–1115
- Grasso F, Rahwan I, Reed C, Simari GR (2010) Introducing argument & computation. *Argument Comput* 1(1):1–5
- Green N (2005) A Bayesian network coding scheme for annotating biomedical information presented to genetic counseling clients. *J Biomed Inform* 38:130–144
- Isaac T, Weissman JS, Davis RB, Massagli M, Cyrulik A, Sands DZ, Weingart SN (2009) Overrides of medication alerts in ambulatory care. *Arch Intern Med* 169(3):305–311
- Jenicek M, Hitchcock D (2005) *Logic and critical thinking in medicine*. AMA Press, Chicago
- Kuhn D (1962) *The structure of scientific revolutions*. University of Chicago Press, Chicago and London
- Moss L (2010) *Explaining anomalies: an approach to anomaly-driven revision of a theory*. University of Aberdeen
- Moss L, Sleeman D, Sim M, Booth M, Daniel M, Donaldson L, Gilhooly C, Hughes M, Kinsella J (2010) Ontology-driven hypothesis generation to explain anomalous patient responses to treatment. *Knowl Based Syst* 23(4):309–315
- Parsons S, Wooldridge M, Amgoud L (2003) Properties and complexity of some formal inter-agent dialogues. *J Log Comput* 13(3):347–376
- Rahati A, Kabanza F (2010) Persuasive dialogues in an intelligent tutoring system for medical diagnosis. In: *Proceedings of international conference on Intelligent Tutoring Systems (ITS)*
- Tamblyn R, Huang A, Taylor L et al (2008) A randomized trial of the effectiveness of on-demand versus computer-triggered drug decision support in primary care. *J Am Med Inform Assoc* 15(4):430–438
- Teich JM, Merchia PR, Schmitz JL et al (2000) Effects of computerized physician order entry on prescribing practices. *Arch Intern Med* 160(18):2741–2747
- Tierney WM, Overhage JM, Murray MD et al (2003) Effects of computerized guidelines for managing heart disease in primary care. *J Gen Intern Med* 18(12):967–976
- Tolchinsky P, Cortés U, Modgil S, Caballero F, López-Navidad A (2006) Increasing human-organ transplant availability: argumentation-based agent deliberation. *IEEE Intell Syst* 21(6):30–37
- Vázquez-Salceda J, Padget JA, Cortés U, López-Navidad A, Caballero F (2003) Formalizing an electronic institution for the distribution of human tissues. *Artif Intell Med* 27(3):233–258

- Walton DN (1995) *Argumentation schemes for presumptive reasoning*. Routledge
- Walton R (2006) An evaluation of CAPSULE, a computer system giving advice to general practitioners about prescribing drugs. *J Informat Prim Care* 2006:2–7
- Walton DN, Krabbe EC (1995) *Commitment in dialogue. Basic concepts of interpersonal reasoning*. State University of New York Press, Albany
- Williams MH (2008) PhD thesis: integrating ontologies and argumentation for decision-making in breast cancer, University College London, London
- Ye RL, Johnson PE (1995) The impact of explanation facilities on user acceptance of expert systems advice. *MIS Q* 19(2):157–172

Chapter 11

Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to-Human Tutoring

Sidney D'Mello and Art Graesser

Abstract This chapter provides an overview of dialog-based intelligent tutoring systems (ITSs), which are learning technologies that help learners develop mastery of difficult subject matter by holding conversations in natural language. The first section discusses some of the basic issues in the design of dialog-based ITSs, while the second section highlights recent advances in this area. The first section begins with an analysis of human–human tutorial dialogs followed with a discussion of the six major components of most dialog-based ITSs: input transformation, speech-act classification, learner modeling, dialog management, output rendering, and domain modeling. These abstract components are concretized within the context of one of the first dialog-based ITSs, AutoTutor. The second section discusses recent advances in the area with an emphasis on systems that model learners' emotional states in addition to their cognitive states. These include a system that automatically adapts its dialogs based on whether the learner is bored, confused, or frustrated, a system with unique mechanisms to monitor and correct learners' disengagement behaviors by tracking eye gaze, and a system that strategically plants confusion in the minds of learners to engender deeper modes of thinking. We conclude the chapter by discussing some of the open issues in dialog-based ITSs, such as identifying benefits of spoken versus typed input, understanding when

S. D'Mello (✉)

Department of Psychology and Computer Science, University of Notre Dame,
384 Fitzpatrick, Notre Dame, IN 46556, USA
e-mail: sdmello@nd.edu

A. Graesser

Department of Psychology, Institute of Intelligent Systems,
University of Memphis, Memphis, TN, USA

Department of Education, University of Oxford, Oxford, UK
e-mail: art.graesser@gmail.com

imperfect natural-language understanding is sufficient, contrasting the importance of the message vs. the medium in influencing learning, and identifying conditions in which dialog-based tutoring is effective.

Introduction

There is somewhat of a paradox in the information age. Hardware is light, cheap, and portable while software is efficient, adaptive, and increasingly intelligent. Books are online, breaking news is streamed on our handheld devices, and billions of disparate units of information are integrated, indexed, and easily accessible with a click of a mouse. The computer is now a virtual extension of our human brains. Yet our computer interfaces are still primitive, nonintuitive, difficult to use, and socially challenged. Humans communicate with computers through windows, icons, menus, and pointing devices (i.e., the WIMP paradigm). But humans communicate with each other primarily through speech and a host of nonverbal cues such as facial expressions, paralinguistic features of speech, oculesics, posture, and gesture. Computer systems that are able to recognize and respond to these communication channels will presumably provide more effective, meaningful, and natural interaction experiences. The ability to communicate with the computer through natural-language dialogs represents a significant advancement towards narrowing the communicative bandwidth between the human and the computer. This was once no more than fanciful science fiction (e.g., HAL 9000), but the recent successes of IBM's Jeopardy-playing AI, Watson, and Apple's Speech Interpretation and Recognition Interface (SIRI), have reignited the imagination about effective natural-language communicative interfaces.

One class of technologies that would benefit from natural-language interactions is intelligent tutoring. Intelligent tutoring systems (ITSs) mimic one-on-one human tutoring, a proven method for promoting active construction of knowledge beyond textbooks and traditional classroom environments (Graesser et al. 2009a; VanLehn 2011; Woolf 2009). ITSs have implemented several systematic strategies for increasing learning gains, such as error identification and correction, building on prerequisites, frontier learning (expanding on what the learner already knows), learner modeling (inferring what the learner knows and having that information guide tutoring), and building coherent explanations (Gertner and VanLehn 2000; Graesser et al. 2012; Sleeman and Brown 1982; VanLehn 2006; Woolf 2009).

ITSs are effective in promoting learning because they are continually assessing learner knowledge and tailoring the instruction in a manner that is sensitive to individual learners. VanLehn (2011) recently compared the effectiveness of answer-based systems, step-based ITSs, and human tutoring for promoting learning beyond instructional controls that teach equivalent content without tutoring. Answer-based systems (e.g., most Computer-Assisted Instruction (CAI), Computer-Based Training (CBT), and Computer-Assisted Learning (CAL) systems) typically pose problems and evaluate learners' final answers to those problems; the learner does

all of the reasoning. The performance of answer-based systems was disappointing: 0.31 sigma mean effect vs. the human-tutoring mean effect of 0.79 sigma. Step-based ITSs provide learners multiple reasoning opportunities either through tutorial dialog or through prompt-based interactions as learners work through problems. The mean effect for step-based ITSs was an impressive 0.76 sigma (Cohen's *d*) which rivaled the human-tutoring effect of 0.79 sigma.

This chapter discusses a subset of ITSs that implement natural-language dialogs that mimic conversations found in human tutoring. These include AutoTutor (Graesser et al. 2004a; VanLehn et al. 2007), ITSPOKE (Litman et al. 2006), why-Atlas (Graesser et al. 2001; VanLehn et al. 2002), CIRCSIM-Tutor (Evens et al. 1997), DC-Trains (Pon-Barry et al. 2004), My Science Tutor (Ward et al. 2011), Research Methods Tutor (Arnott et al. 2008), and Mission Rehearsal (Gratch et al. 2002). These computer tutors vary in the extent to which they simulate human dialog mechanisms, but all of them attempt to comprehend natural language, formulate adaptive responses, and implement pedagogical strategies to facilitate learning.

The effectiveness of dialog-based ITSs can be understood from the broad perspective of constructivism, which is a theoretical framework adopted by many researchers who are exploring ways to increase learning at deeper levels of comprehension (Biggs 1996; Bransford et al. 1991; Chi et al. 1994; Piaget 1952; Vygotsky 1978). According to many of these constructivist approaches, the learner needs to actively construct coherent, explanation-based meanings and knowledge by interacting with the world and other people. Learning environments should stimulate active construction of knowledge and provide feedback and explanations on these constructions rather than being mere information delivery systems. Dialog-based ITSs that adhere to constructivist principles attempt to get learners to do most of the talking by pumping for information and by providing hints, prompts, forced choices, and other pedagogical scaffolds. The onus of knowledge construction is placed on the learner and involves cognitive processes, such as perception, management of working memory, planning, the production of language and discourse constituents, and the consolidation of subject-matter knowledge. Dialog-based ITSs are thought to be effective because they help scaffold the co-construction of knowledge via mixed-initiative, natural-language dialogs.

The present chapter describes how dialog-based ITSs leverage recent advances in natural-language processing (NLP) to effectively simulate human-human tutorial dialogs. It is beyond the scope of this chapter to delve into the specifics of the large number of successful dialog-based ITSs that have been implemented and tested (Arnott et al. 2008; Evens et al. 1997; Graesser et al. 2001; Gratch et al. 2002; Litman et al. 2006; Pon-Barry et al. 2004; VanLehn et al. 2002; Ward et al. 2011). Instead, we focus on some of the general principles of dialog-based ITSs and anchor our discussions to specific dialog-based ITSs that we have been developing over the last 15 years.

The remainder of the chapter is organized as follows: We begin with an analysis of human-human tutorial dialogs with an emphasis on curriculum scripts, the five-step tutoring frame, expectation and misconception-tailored dialogs, and conversational-turn management. We then outline six major components that are

common to most dialog-based ITSs. These include input transformation, speech-act classification, learner modeling, dialog management, output rendering, and domain modeling. Next, we flesh out these components within the context of AutoTutor, which is one of the first dialog-based ITSs we developed (Graesser et al. 1999). We also provide a synthesis of several studies that have evaluated the ability of AutoTutor to increase learning by appropriately modeling learner knowledge, dynamically tailoring the interaction to individual learners' knowledge and abilities, and generating natural-language dialogs that are conversationally smooth.

We move beyond the basic AutoTutor system to novel dialog systems that model and respond to emotions and disengagement behaviors in addition to knowledge levels. These include systems that (a) respond to learners' emotional states and synthesize emotions, (b) sense emotions by analyzing textual cues and other discourse features, (c) detect and respond to disengagement behaviors, and (d) induce cognitive disequilibrium in the minds of learners. We conclude by discussing some of the open issues in dialog-based ITSs such as identifying benefits of spoken versus typed input, understanding when imperfect natural-language understanding is sufficient to produce meaningful interactions, contrasting the importance of the content of the dialogs versus the communicative medium for learning, and identifying some of the conditions under which differing forms of dialog-based tutoring have proven to be effective.

Structure of Human–Human Tutorial Dialogs

Graesser and Person analyzed dialog patterns of typical human tutors in middle school and in college (Graesser and Person 1994; Graesser et al. 1995). Similar analyses have been conducted by other researchers on natural tutoring corpora (Chi et al. 2004; D'Mello et al. 2010; VanLehn et al. 2003). The following dialog structures are prominent in human–human tutoring and are implemented in many dialog-based ITSs: (a) a curriculum script with didactic content and questions, (b) a five-step tutoring frame, (c) expectation and misconception tailored (EMT) dialog, and (d) conversational turn management.

Curriculum Script

The tutor covers a curriculum with didactic content and a set of questions or problems that address the content. Didactic content can be presented in a mini-lecture, ideally at the appropriate time for each individual learner (D'Mello et al. 2010). The questions/problems require learners to actively apply their knowledge. The curriculum script includes expected answers, misconceptions, hints, prompt questions, and other information.

Five-Step Tutoring Frame

When a challenging main question (or problem) is selected to work on, the question is answered through an interaction that is structured by a five-step tutoring frame. The five steps are:

1. The tutor presents a main question.
2. The learner gives an initial answer.
3. The tutor gives short feedback on the quality of the learner's initial answer.
4. The tutor and learner collaboratively improve on the answer in a turn-by-turn dialog that may be lengthy (100–200 turns).
5. The tutor evaluates whether the learner understands (e.g., asking “Do you understand?” or testing with a follow-up task).

In the spirit of constructivism, this five-step tutoring frame involves collaborative discussion, joint action, and encouragement for the learner to construct knowledge rather than merely receiving knowledge.

Expectation and Misconception Tailored Dialog

Human tutors typically have a list of *expectations* (i.e., anticipated good answers or steps in a procedure) and a list of anticipated *misconceptions* (incorrect information) associated with each main question. The expectation content needs to be covered (i.e., articulated over the course of the dialog) in order address the main question that is selected. The tutor guides the learner in articulating the expectation through a number of dialog moves. They include *pumps* (“What else?”), *hints* (“What about X?”), *prompt questions* to extract specific information from learners (“X is a type of what?”), *assertions* that capture particular expectations (“X is a type of Y”), *answers* to learners' questions, and *summaries* that synthesize content.

Conversational Turn Management

Human tutors structure their conversational turns systematically. Nearly every turn of the tutor has three information slots. The first slot is feedback on the quality of the learner's last turn. This feedback is either positive (e.g., “very good,” “yeah”), neutral (e.g., “uh huh,” “I see”), or negative (e.g., “not quite,” “not really”). The second slot advances the interaction with a prompt for specific information, a hint, an assertion with correct information, a correction of misconceptions, or an answer to the learner's question. The third slot is a cue for the floor to shift from the tutor as the speaker to the learner.

Major Components of Dialog-Based ITSs

The behavior of most ITSs can be characterized by an outer and an inner loop (VanLehn 2006). The outer loop consists of a series of didactic lessons and challenging problems related to the topic being tutored. For example, a tutor of cell biology might cover cell structures, the cell cycle, mitosis, and cytokinesis. The order of topics can be dynamically selected based on the profile of learner abilities or can be rigidly scripted on the basis of a content analysis that attempts to identify dependencies among topics. The inner loop consists of collaborative tutorial interactions geared towards covering steps or pieces of the content of one topic in the outer loop. It is through this inner loop that the tutor draws out more of the learner's knowledge with probing questions, helps fill in missing information with explanations, repairs misconceptions, and answers the learner's questions. The interactive dialog occurs during the co-construction of knowledge in this inner loop but not during the didactic delivery of information (e.g., reading text, viewing a diagram).

We now turn to the six major components that an ITS needs in order to implement the dialog structure human tutors. These include mechanisms to: (1) transform learner responses, (2) classify responses into speech acts, (3) model learner knowledge, (4) manage the dialog, (5) render the tutor's actions, and (6) represent the domain knowledge needed by the aforementioned five mechanisms. Figure 11.1 provides a sketch of how these components might interact in a dialog-based ITS. Of course, the exact nature of the ITS will govern how information passes through these components.

Input Transformation

This component transforms learners' responses into a form that can be interpreted by the ITS. The input transformation component for ITSs that only support typed input is quite minimal and might involve converting the responses to lower case, correcting misspellings, and converting words to root forms (i.e., stemming). ITSs that process spoken input (D'Mello et al. 2010a; Litman et al. 2006) need an additional step of translating learner speech into text via automatic speech recognition.

Speech Act Classification

In most cases, learners' responses primarily consist of answers to questions posed by the tutor. These responses (called *contributions*) have substantive content potentially relevant to the answer and therefore are analyzed in the subsequent learner modeling

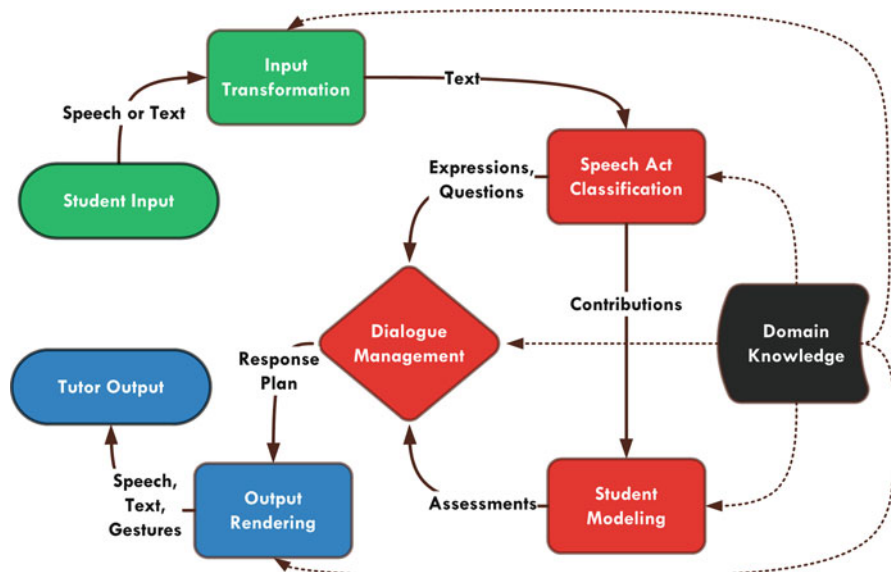


Fig. 11.1 Sketch of six major components of a dialog-based ITS and how they might interact

phase. However, learners occasionally provide content-free *frozen expressions* that signify particular discourse functions. These might include *conversational acknowledgements* (e.g., “ok”), short responses (e.g., “yes,” “no”), *metacognitive statements* (e.g., “I need help,” “I don’t know”), *metacommunicative statements* (e.g., “please repeat,” “could you say that in another way”), and *gripes* (e.g., “ugh”). In addition to these frozen expressions, learners sometimes take initiative by asking questions. The Graesser and Person (1994) taxonomy identifies 16 question categories that occur in educational settings. Examples include *verification questions* that invite yes or no responses (e.g., “is velocity the same as speed?”), *definition questions* (e.g., “What does momentum mean?”), *comparison questions* (e.g., “how is mass similar to weight”), and *causal antecedent questions* (e.g., “why did the pumpkin land on the runner’s head?”).

Frozen expressions and questions are exceptions to the basic dialog cycle which mainly consists of tutors asking questions, learners providing contributions, and tutors providing feedback. Both frozen expressions and questions need to be handled in separate sub-dialogs that are related to, but break away from, the main dialog. Hence, most dialog-based ITSs need some form of Speech Act Classifier (SAC) in order to differentiate the various types of learner responses. At a coarse-grained level, the SAC needs to distinguish the more common contributions from frozen expressions and questions. At a more fine-grained level, the SAC also needs to classify the different types of frozen expressions and questions.

Learner Modeling

Learner modeling is one of the most significant components of any ITS because, by definition, an ITS must tailor its feedback and instruction to individual learners (Psotka et al. 1988; Sleeman and Brown 1982; Woolf 2009). A dialog-based ITS generally needs to model learners at two levels of granularity. Modeling at a *local level* involves comparing learners' *immediate* contribution to the expected answer (i.e., a specific keyword or phrase). This information is used to provide feedback and make local dialog decisions, such as whether to follow a hint with a prompt or simply assert the information. Learner modeling at the *global level* consists of assessing the learners' evolution of knowledge of the topic being tutored. This is needed to identify when a particular expectation is covered, which expectations are remaining to be covered, and which expectation should be covered next. Modeling at both levels might also involve comparing the extent to which learners' responses align with specific misconceptions, so that these might be immediately or eventually corrected. Many ITSs in well-defined domains like mathematics use some form of Bayesian knowledge tracing (Conati et al. 2002; Corbett and Anderson 1994) for learner modeling; the task is considerably more difficult for dialog-based ITSs in ill-defined domains. We have addressed the problem of modeling learner knowledge in certain ill-defined domains with semantic processing algorithms as will be subsequently discussed.

Dialog Management

Effective pedagogy and smooth conversation require dialog management. The dialog manager needs to balance the competing goals of being flexible, opportunistic, goal-oriented, and conversationally smooth. Flexibility is needed in order to accommodate virtually any input of the learner without having the conversation break down. ITSs also need to be opportunistic by capitalizing on learning opportunities as they arise. For example, learners rarely take the initiative by asking questions, yet some types of questions are strongly correlated with deeper levels of cognition per the Bloom's taxonomy of educational objectives in the cognitive domain (Bloom 1956). Similarly, misconceptions and other impasses provide some of the most promising learning opportunities because they force learners to stop to think (D'Mello et al. *in press*; VanLehn et al. 2003). An effective dialog manager will take advantage of these learning opportunities by dynamically launching question-answering and misconception-correction sub-dialogs. However, in contrast to being opportunistic, the dialog manager must also ensure that existing dialog plans are completed without too many interruptions, otherwise the dialog will appear fragmented. Hence, a dialog manager needs to decide when it is appropriate to be more goal-oriented versus opportunistic, and it needs mechanisms

to temporarily shelve , but eventually return to, existing plans when sub-dialogs that take advantage of learning opportunities are launched.

Finally, a dialog manager also needs to emulate the pragmatics of human–human tutorial dialogs. A collaborative exchange between a dialog-based ITS and the learner requires a mutual understanding of the turn-taking process. In human-to-human conversations, speakers signal to listeners that they are relinquishing the floor and that it is the listener’s turn to say something (Clark 1996; Sacks et al. 1974). However, human-to-computer conversations lack many of the subtle signals inherent to human conversations, so there need to be discourse markers and other cues that guide the learner in the exchange (Freedman 1996).

Output Rendering

This component is responsible for communicating the tutor’s actions to the learners. In the simplest case, this can involve displaying the content of the tutor’s utterance on the screen. An alternate possibility is to use computer-generated animated agents that have facial features synchronized with speech and, in some cases, appropriate gestures. Learning environments with animated conversational agents have become more popular in recent years and it has been well documented that such systems improve learning (Baylor and Kim 2005; Johnson et al. 2000; Moreno and Mayer 2007). With this approach, the computer controls the eyes, eyebrows, mouth, lips, and other parts of the face in a fashion that is appropriately meshed with the content and intent of the speaker. Animated pedagogical agents can be an important feature of dialog-based ITSs because they help ground the conversation between the tutor and learner. The nonverbal facial cues are known to be an important form of backchannel feedback during tutoring (Fox 1993), as well as in other contexts of conversation (Clark 1996). Similarly, pitch, pause, duration, amplitude, and intonation contours are among the paralinguistic cues that signal backchannel feedback, affect, and emphasis (Brennan and Williams 1995). Gestures are particularly effective at directing attention, simulating chains of cause and effect, and playing other crucial roles in communication (Alibali and Nathan 2012; Goldin-Meadow 2003).

Domain Model

The five components specified above describe an architecture that is common to many dialog-based ITSs. However, the architecture needs various forms of content to produce behavior. Pronunciation and language models are needed to recognize spoken speech. Templates are required for speech-act classification. Semantic models are needed for learner modeling. A curriculum script with didactic descriptions, tutor-posed questions, example problems, figures, and diagrams (along

Table 11.1 Excerpt of tutorial dialog between a learner and AutoTutor

| Speaker | Utterance |
|-----------|---|
| AutoTutor | When you turn on the computer, how is the operating system first activated and loaded into RAM? <i>[this is the main question]</i> |
| Learner | When the computer is turned on, a file is automatically booted up |
| AutoTutor | Anything else? <i>[this is a pump for more information]</i> |
| Learner | This file contains the operating system |
| AutoTutor | What about storage? <i>[this is a hint]</i> |
| Learner | The operating system is then read into RAM and activated |
| AutoTutor | Right. The CPU executes instructions in ROM, which helps the operating system to start. <i>[after giving positive feedback, there is an assertion]</i> |
| Learner | After being read from ROM the OS is activated |
| AutoTutor | Right. Let’s review what we’ve just gone over. The operating system must first be stored on the hard disk. When you turn on the computer, the CPU executes instructions stored in ROM. These instructions help boot the operating system. The operating system is then loaded from the hard disk into RAM. <i>[after positive feedback comes a summary]</i> |

with anticipated good responses to each topic) is also needed. There should also be a glossary of technical terms with definitions in order to answer learner questions (particularly “What does X mean?” questions). Finally, different animations that synchronize speech, facial expressions, and gestures are needed when animated pedagogical agents are used to model the tutor.

Fleshing Out the Major Components of Dialog-Based ITSs in AutoTutor

We have discussed some of the dominant dialog patterns in human–human tutoring and have provided a high-level sketch of how these might be implemented in a computer tutor. We now provide a more concrete sketch of these components by using AutoTutor as an example of a fully functional system. AutoTutor is a dialog-based ITS that simulates a human tutor by holding a conversation with learners in natural language (Graesser et al. 2004a, 2005a). AutoTutor has been implemented and tested for the domains of Newtonian physics, various aspects of computer literacy, and research methods. An excerpt of an exchange between AutoTutor and a learner is shown in Table 11.1.

As with most ITSs, AutoTutor’s behavior can be characterized by an outer and an inner loop (VanLehn 2006). The outer loop of AutoTutor consists of a series of didactic lessons and challenging problems or *main questions* (such as *why*, *how*, *what-if*). An example main question is “When you turn on the computer, how is the operating system first activated and loaded into RAM?” An ideal answer to a main question requires several sentences of information. After the learner enters an initial response, AutoTutor assists the learner in constructing a more accurate/better answer.

The inner loop of AutoTutor consists of this kind of collaborative interaction for answering a main question (or solving a problem). It is this inner loop that is the distinctive hallmark of AutoTutor. The tutor draws out more of the learner's knowledge (through hints and prompts), helps fill in missing information, repairs misconceptions, and answers learner questions. The inner loop dialog between AutoTutor and the learner takes approximately 100 turns to answer a single challenging question, approximately the length of a conversation with a human tutor (Graesser et al. 1995).

Input Transformation

In most versions of AutoTutor, learners type their responses using a keyboard. However, we have developed a version that handles spoken input from the learner with the Dragon Naturally Speaking™ (version 6) speech recognition system (D'Mello et al. 2010a).

Speech Act Classification

AutoTutor's Speech-Act Classifier performs a two-step analysis of learners' responses. Each speech act is assigned to 1 of 20 speech-act categories: contributions, 16 categories of questions, short responses, meta-cognitive expressions, and meta-communicative expressions. The classifier does a part of speech tagging on the utterance followed by a cascade of finite-state transducers that operate on the tagged text (Olney et al. 2003). Each transducer defines a particular speech-act category. For example, the finite-state transducer to recognize a concept-completion question (e.g., "When is the Operating System loaded?") would resemble "[^](Who|What|When|Where)." The transducers rely heavily on the output of the part of speech tagger, which in turn relies on the syntactic integrity of the input utterance.

Learner Modeling

Learner modeling in AutoTutor requires semantic matching algorithms that compare the learner input with AutoTutor's expectations and misconceptions. AutoTutor incorporates several semantic evaluation algorithms when performing these matches, but most notably Latent Semantic Analysis (LSA) (Landauer et al. 2007). LSA is a statistical technique that measures the conceptual similarity of two text sources. In this semantic matching algorithm, a vector representing the semantic content of the contribution is created and compared to vectors that represent the

semantic content of expectations and misconceptions. The cosine between the two vectors is calculated to produce a match-similarity score from 0 to 1 (negative cosines are rare and are converted to 0 in AutoTutor).

The LSA algorithm in AutoTutor computes the extent to which the information within the learner turns (i.e., an individual turn, a combination of turns, or collective sequence of turns) semantically matches each expectation in the ideal answer. Expectation E_i is considered covered if the content of the learner's cumulative set of turns meets or exceeds a threshold T in its LSA cosine value (which varies from near 0 to 1). That is, E_i is covered if the cosine match between E_i and the learner input I (including turns 1 through N) is high enough: $\text{cosine}(E_i, I) \geq T$. The threshold has varied between 0.40 and 0.75 in various instantiations of AutoTutor.

In the course of the dialog- and learner-modeling, the system periodically identifies a missing expectation and posts the goal of covering the expectation. When expectation E_i is missed (and therefore posted), AutoTutor attempts to get the learner to articulate it by generating hints and prompts affiliated with E_i to help the learner fill in missing words and propositions. Prompts and hints are selected to maximize an increase in the LSA cosine match score (hereafter called the *match score*) when answered successfully. Stated differently, hints and prompts are selected to maximize pattern completion.

Sometimes the learner expresses misconceptions during the dialog. This happens when the learner input I matches a misconception M with a sufficiently high match score. At that point AutoTutor corrects the misconception and goes on.

The selection of the next E_i to cover follows the principle of the zone of proximal development or what some call frontier learning (Brown et al. 1998; Vygotsky 1986). That is, AutoTutor builds on what the learner has managed to articulate. More formally, AutoTutor selects the next E_i from the set of expectations that (a) has the highest match score and (b) has a subthreshold match-score (i.e., the expectation has not yet been covered). This *subthreshold expectation selection* algorithm assumes that the expectations should not be covered in a prescribed sequential order. However, ordering constraints may also be considered in a *sequential expectation selection* algorithm. Some subject matters have ordering constraints but others do not.

While earlier versions of AutoTutor relied exclusively on LSA for learner modeling, newer versions incorporate an inverse word frequency weighted overlap (IWFWO) algorithm as well. The IWFWO algorithm is a word-matching algorithm in which each word is weighted on a scale from 0.0 to 1.0, relative to its inverse frequency in the English language using a corpus like CELEX (Baayen et al. 1995). As a consequence, higher frequency words such as closed-class function words (e.g., and, but, a, and the) have comparatively low weights and little effect on the IWFWO match score. Lower frequency words (e.g., RAM, system, and speed) have higher weights and exert more influence on the IWFWO match score. Similar to LSA, the IWFWO algorithm also generates a semantic-match similarity score between 0 and 1 for each expectation.

Both LSA and IWFWO have strengths and weaknesses which are extensively discussed in previous publications (Hu et al. 2007). Hence, some versions of

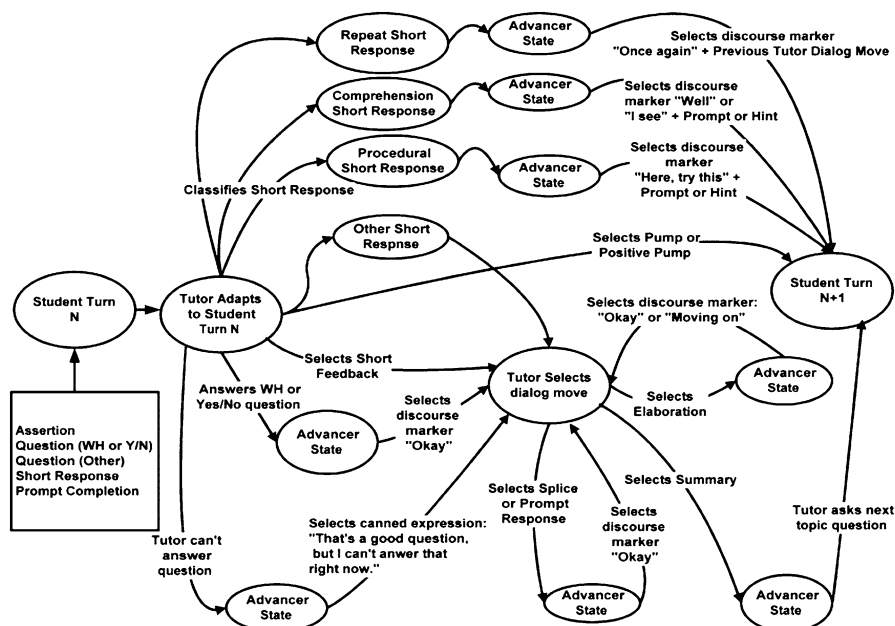


Fig. 11.2 Sample dialog advancer network

AutoTutor use a hybrid match score ($\text{HYBRID} = 0.33 \times \text{LSA} + 0.67 \times \text{IWFOWO}$) to leverage the benefits of both systems while minimizing the effects of their flaws. The most recent systems use a combination of LSA, IWFOWO and regular expressions to assess semantic matches (Cai et al. 2011). The regular expressions provide a significant added boost in the accuracy of the semantic matches. In contrast, the addition of a syntactic parser has had a nonsignificant impact on match scores. The impact of syntax is small because a large percentage of student contributions are ungrammatical. Syntax plays an important role in speech act classification, but not in semantic matching algorithms for learner modeling.

Dialog Management

The dialog-management module in most versions of AutoTutor is an augmented, finite-state transition-network (see Fig. 11.2). The nodes in the network refer to knowledge goal-states (e.g., expectation E is under focus and AutoTutor wants to get the learner to articulate it) or dialog states (e.g., the learner just expressed an assertion as his or her first turn in answering the question). The arcs refer to categories of tutor dialog moves (e.g., feedback, pumps, prompts, hints, summaries) or discourse markers that link dialog moves (e.g., "okay," "moving on,"

“furthermore”). A particular arc is traversed when particular conditions are met. For example, a pump arc is traversed when it is the learner’s first turn and the learner’s assertion has a low LSA match value.

Arc traversal is normally contingent on outputs of computational algorithms and procedures that are sensitive to the dynamic evolution of the dialog. These algorithms and procedures operate on the snapshot of parameters, curriculum content, knowledge goal-states, learner knowledge, dialog states, LSA measures, and so on, which reflect the current conversation constraints and achievements. For example, there are algorithms that select dialog move categories intended to get the learner to fill in missing information in *E* (the expectation under focus). There are several alternative algorithms for achieving this goal.

One of the early algorithms that we adopted relied on fuzzy production rules. If the learner had almost finished articulating *E* but lacked a critical noun or verb, then a prompt category would be selected because the function of prompts is to extract single words from learners. The particular prompt selected from the curriculum script would be tailored to extract the particular missing word through another module that fills posted dialog move categories with particular content. If the learner is classified as having high ability and has failed to articulate most of the words in *E*, then a hint category might be selected. Fuzzy production rules made these selections. A more elaborated algorithm to fleshing out *E* uses one or two cycles of hint–prompt–assertion. That is, AutoTutor’s selection of dialog moves over successive turns follows a particular order: first hint, then prompt, then assert in cycle 1 and optionally in cycle 2 another hint, then prompt, then assert. AutoTutor exits the one or two cycles as soon as the learner articulates *E* to satisfaction (i.e., the semantic-match threshold is met) or the cycles are exhausted.

Some versions of AutoTutor have a more complex dialog-management mechanism that involves more complex planning and a modular architecture (Graesser et al. 2005a). However, it is beyond the scope of this chapter to address such enhancements.

Output Rendering

AutoTutor communicates through an animated conversational agent with speech, facial expressions, and rudimentary gestures. The various versions of AutoTutor have used several text-to-speech engines and facilities to render facial expressions and body movements. From the standpoint of the present chapter, we have relied on simple solutions to rendering the output. Regarding AutoTutor’s spoken utterances, we have primarily used text-to-speech output of available speech synthesizers rather than dynamically computing mark-up codes to make the speech more discourse sensitive. Our attempts to have discourse-sensitive mark-up (for emphasis on words and changes in intonation) have proven futile because of the impenetrable mechanisms of the text-to-speech engines. However, given that some words are not

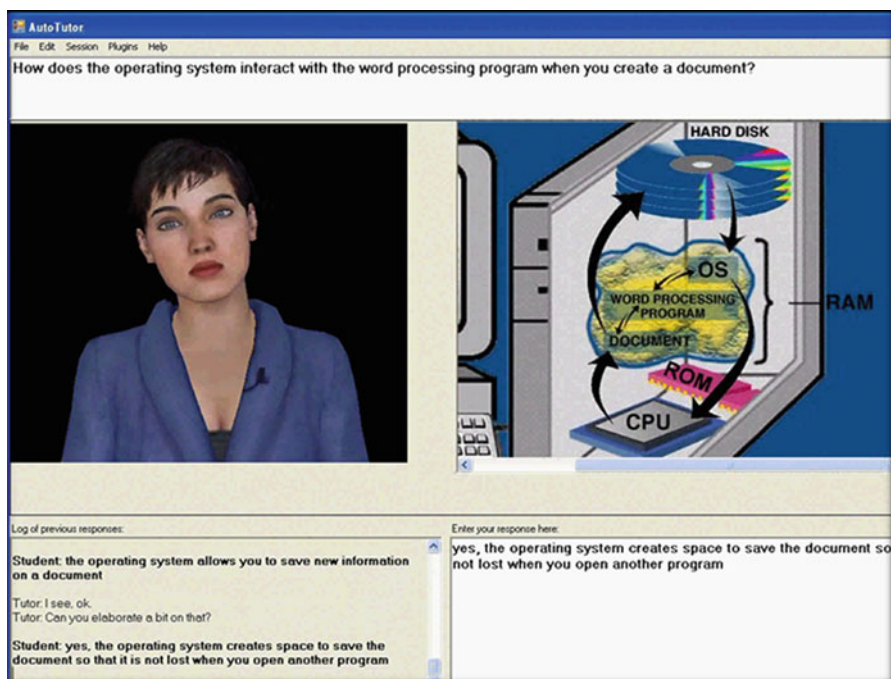


Fig. 11.3 Screenshot of AutoTutor interface

pronounced correctly by the synthesizers, we have routinely composed a library of mark-up codes or have changed the spellings of these words to make them sound more natural. Regarding facial expressions and gestures, we have a library of facial expressions and gestures that are coordinated with specific speech acts or discourse markers; again these are not dynamically composed in a discourse-sensitive manner. For example, the tutor delivers positive feedback with a verbal message (e.g., “good job”) and a nod. We also have algorithms for randomly having the agents blink their eyes or make movements to make the demeanor look alive.

A screenshot of one of the more recent versions of the AutoTutor interface is shown in Fig. 11.3. The interface typically has five major windows shown in the figure. Window 1 (top of screen) is the main question that stays on the computer screen throughout the conversation about the question. Window 2 (left middle) is the animated conversational agent that speaks the content of AutoTutor’s turns and gestures appropriately. Window 3 (right middle) is either blank or has auxiliary diagrams. Window 4 (right bottom) displays the learners’ answers as they type them in. Window 5 (left bottom) displays the dialog history of the learner and the tutor.

Domain Knowledge

Content in AutoTutor is primarily represented as a set of words, sentences, or paragraphs in a free text format. This makes it easy for a lesson planner to create new topics and content without having to craft the content in structured LISP or Prolog code, as is the case with most ITSs. The system was designed this way so that AutoTutor could be used for virtually any topic except those that require the precision of mathematics, and so lesson planners could develop the content with minimal knowledge of discourse or computer programming.

The curriculum script in AutoTutor organizes the topics and content of the tutorial dialog. Each script contains the content associated with a question or problem. For each, there is (1) the ideal answer, (2) a set of expectations, (3) families of potential hints, correct hint responses, prompts, correct prompt responses, and assertions associated with each expectation, (4) a set of misconceptions and corrections for each misconception, (5) a set of key words and functional synonyms, (6) a summary, and (7) markup language for the speech generator and gesture generator for components in (1) through (6) that require actions by the animated agents.

AutoTutor utilizes large text bases for the semantic-processing algorithms. All versions of AutoTutor represent world knowledge as *LSA* spaces, but some versions of AutoTutor and its progeny have incorporated other forms of world-knowledge representation, such as textbooks, glossaries, and conceptual graph structures. Some of this information (e.g., texts and keywords) is also used to build custom language models for the versions that incorporate spoken input.

In addition to the domain-specific content contained in the curriculum script and the text bases, AutoTutor has a repository of different domain-independent data structures. There are *conversation rules* that are represented as production rules, finite-state transition networks, or recursive, augmented state transition networks. There are different templates of speech acts that are used by the speech-act classifier. There are different categories of feedback (e.g., positive, neutral, negative), each with a set of exemplars that are randomly selected at runtime. Finally, there are discourse markers that help connect clauses in order to form syntactically correct expressions.

Evaluating Dialog-Based ITS

Dialog-based ITSs service multiple goals so they need to be evaluated along multiple dimensions. Perhaps the most important goal is to promote learning gains while simultaneously keeping learners engaged. This goal is serviced by sub-goals such as recognizing speech (when spoken input is enabled), classifying utterances, comprehending contributions to provide appropriate feedback, modeling knowledge, and generating conversationally smooth tutorial dialogs that mirror

human–human tutoring dialogs. Using AutoTutor as a case study, we describe how dialog-based ITSs can be evaluated along these different dimensions. We begin with evaluations of the individual components followed by more global evaluations of overall effectiveness.

Recognizing Spoken Responses

Word error rate (WER) and word recognition rate (WRR) are standard metrics for assessing the reliability of automatic speech-recognition systems. $WER = [S + D + I]/N$, where S , D , and I are the number of substitutions, deletions, and insertions in the automatically recognized text (with errors) when compared to the ideal text (no errors) of N words. $WRR = 1 - WER$. The word-recognition rate (WRR) for the automatic speech-recognition system used in speech-enabled version of AutoTutor (D'Mello et al. 2010a) has ranged from 0.017 to 0.887 with a mean of .542 ($SD = 0.270$). Most of the errors appear to be substitution (52.3%) and insertion errors (39.1%) while deletion errors were comparatively rare (8.8%).

WER and WRR metrics frame automatic speech-recognition as a speech-to-text translation problem. The problem can also be framed as an information-extraction problem if one is simply interested in quantifying how many words were correctly recognized, without regard to the ordering of words. When word order is ignored, on average 75% ($SD = 15.3\%$) of the words were correctly recognized by our ASR system. Recognition is somewhat higher for content words (81.5%) compared to all words (76.5%).

Taken together, these results suggest that the automatic speech-recognition system would be problematic for a system that requires a syntactically intact utterance in order to evaluate a learner's response. However, performance is expected to be relatively stable for ITSs that rely on shallow NLU techniques because these algorithms match key words and phrases while ignoring the syntax of the utterance.

Classifying Learners' Responses

Olney et al. (2003) evaluated how accurately AutoTutor's speech-act classifier could classify approximately 10,000 learner utterances extracted from AutoTutor's log files. The classifier discriminated among contributions, frozen expressions (metacognitive and metacommunicative statements), and the 16 questions from the Graesser and Person (1994) taxonomy. The system achieved a modest average F -measure of 0.54. This was attributed to the fact that approximately 97% of the learner utterances were contributions and several of the question categories rarely occurred (example questions never occurred). To account for this, a measure that weighted accuracy rates of each category in a manner that reflects occurrence of

the categories was computed. The average weighted F -measure was 0.97, which is quite impressive and more reflective of real-world performance.

Comprehending Learners' Responses

We have analyzed the accuracy of the semantic-matching algorithms by comparing AutoTutor's scores to judgments of subject-matter experts (Graesser et al. 2000, 2007b). For example, we have analyzed the complete answers that learners gave as an answer to one of the challenging physics questions, recorded AutoTutor's match score for each expectation/misconception, and collected ratings from five expert physicists as to whether each expectation/misconception was present in the learners' answers. The correlations between these match-evaluation scores and expert ratings have varied between 0.35 and 0.50, depending on the criterion, semantic algorithm, and other details that need not be considered here.

In other studies, graduate students, instead of experts, rate the extent to which learner essays express particular sentence-like expectations. Similarly, the LSA component evaluates the extent to which the expectations are covered. The correlations between computer and these novice humans are significant and impressive, ranging from 0.5 to 0.7, when computing the proportion of expectations covered in an essay. In general, the accuracy of AutoTutor's LSA component is on par with graduate-level research assistants but not quite as good as accomplished experts (Graesser et al. 2005b).

Cai et al. (2011) recently conducted an analysis on semantic-match scores with the AutoTutor facilities in Operation ARIES!, a new system we developed, as described in subsequent sections. The semantic-match algorithm included regular expressions in addition to LSA. The performance was quite impressive, showing a correlation between the humans and the computer of 0.66 whereas two humans correlated 0.68. Indeed, the regular-expression component had a substantial improvement in accuracy over LSA alone.

Modeling Learner Knowledge

If AutoTutor is performing effective user modeling, then the dialog moves it selects should be correlated with the learners' prior knowledge. In one analysis of conceptual physics, we collected pretest scores on a psychometrically validated test called the Force Concept Inventory (Hestenes et al. 1992). The learners' physics knowledge had a significant positive correlation with positive feedback moves ($r = 0.38$) and a negative correlation with negative feedback ($r = -0.37$) (Jackson and Graesser 2006). Another example applies to the corrections that AutoTutor made when identifying learner errors and misconceptions. The correlation between prior knowledge and corrections was negative ($r = -0.24$), and marginally significant.

In addition to feedback and misconception-correction, AutoTutor's major dialog (pumps, hints, prompts, and assertions) should also be sensitive to learners' knowledge of physics. There is a continuum from the learner-provided information to tutor-provided information as we move from pumps, to hints, to prompts, to assertions. The correlations with learner knowledge reflected this continuum perfectly, with values of 0.49, 0.24, -0.19 , and -0.40 , respectively. Thus, for learners with more knowledge of physics, all AutoTutor needs to do is primarily pump and hint, thereby encouraging or nudging the learner to supply the answer to the question and articulate the expectations. For learners with less knowledge of physics, AutoTutor needs to generate prompts for specific words or to assert the correct information, thereby extracting knowledge piecemeal or telling the learner the correct information.

Generating Naturalistic Dialogs

Person and Graesser (2002) performed a *bystander Turing test* (described below) to evaluate the naturalness of AutoTutor's dialog moves. They randomly selected 144 tutor moves in the tutorial dialogs between learners and AutoTutor. Six human tutors were asked to fill in what they would say at these 144 points. So, at each of these 144 tutor turns, there was a turn with what the human tutor generated versus what AutoTutor generated. They subsequently tested whether a group of learners could discriminate between dialog moves that were generated by a human versus a computer; half in fact were generated by the human tutors and half were by AutoTutor. It was discovered that the bystander learners were unable to discriminate whether particular dialog moves had been generated by a computer versus a human; the d' discrimination scores were actually a bit negative (-0.08), but not significantly. This rather impressive outcome supports the claim that AutoTutor produces a reasonably good simulation of human tutorial dialog. However, there is one note of qualification in this conclusion. The learners would surely be able to discriminate whether they are talking with a human versus a computer if they had a lengthy conversation with the tutor.

Learner's Perceptions of the System

Learner ratings of AutoTutor have also been collected in order to get their impressions of the tutoring environment. The ratings lean toward the positive side, but there have been no systematic comparisons with human tutors or alternative learning environments. We have compared different versions of AutoTutor, but a provocative finding has made us somewhat skeptical about relying on ratings of learner impressions. Specifically Jackson and Graesser (2007) documented that there was a

negative relationship between deep learning and enjoyment: learners least preferred those versions from which they learned most. Learners' metacognition of learning is limited (Graesser et al. 2009b) so it is perhaps not surprising that their ratings of liking were not positively correlated with learning. Deep learning is challenging and sometimes painful, which may clash with an enjoyable experience for some groups of learners.

Increasing Learning

The learning gains of AutoTutor have been evaluated in over 20 experiments since its inception in the areas of computer literacy (Graesser et al. 2004a) and Newtonian physics (VanLehn et al. 2007). Training times in these studies varied from 30 min to 4 h and tutorial sessions were sometimes split across multiple days. Measures of learning have varied in scope, depth, difficulty, and open-endedness. They have included: (1) multiple-choice questions on shallow knowledge that tap definitions, facts and properties of concepts, (2) multiple-choice questions on deep knowledge that tap causal reasoning, justifications of claims, and functional underpinnings of procedures, (3) essay quality when learners attempt to answer challenging problems, (4) a cloze task that has learners fill in missing words of texts that articulate explanatory reasoning on the subject matter, and (5) performance on problems that require problem solving.

AutoTutor improves learning between 0 and 2.1 sigma, depending on the learning performance measure, the comparison condition, the subject matter, and the version of AutoTutor. The mean effect size of 0.8 sigma (a large effect) is on par with learning effects achieved by several other ITSs and by human tutors (VanLehn 2011).

Assessments of learning gains obviously depend on the comparison conditions. The learning gains are approximately 0.8 for AutoTutor compared to a do-nothing control or a condition of reading from a textbook on the same topics for an equivalent amount of time. The learning gains are approximately the same for AutoTutor and an expert human tutor who interacts with the learner by computer-mediated communication (as opposed to face-to-face).

The largest learning gains from AutoTutor have been on deep-reasoning measures rather than measures of shallow knowledge (VanLehn et al. 2007). AutoTutor is most effective when there is an intermediate gap between the learner's prior knowledge and the ideal answers of AutoTutor; AutoTutor is not particularly effective in facilitating learning in learners with high domain knowledge, nor when the material is too much over the learner's head.

Limitations and Points of Failure

We have conducted very detailed analyses of AutoTutor that point to aspects of the dialog and tutoring mechanism that could be improved. One problem lies in errors in interpreting the content of learners' utterances. The pattern matching operations between learner contributions in a turn and AutoTutor's expectation statements (i.e., E1, E2, E3) are not perfect, so some learners get frustrated and conclude that AutoTutor is not listening. This interpretation problem can be mitigated by improving the depth of the pattern matching algorithms, including some facilities for inferences and entailment (Rus and Graesser 2007).

A second problem consists of misclassification of the speech acts. The dialog coherence breaks down when some misclassification errors occur, which ends up confusing learners. More efforts are needed to improve the speech act classification accuracy and to manage the dialog to minimize unwanted consequences.

A third problem occurs when the AutoTutor does not generate relevant and informative answers to the learners questions. AutoTutor can handle roughly half of the learner questions, so half of AutoTutor's replies are either incorrect, constitute requests for clarification (*I don't understand your question, so could you rephrase it?*), or pass the burden onto the learner (*That's a good question, so how would you answer it?*). Improvements in the question answering facilities are needed to minimize this third problem. Fortunately, this third problem does not seriously compromise the dialog because the learners rarely ask questions (Graesser and Person 1994).

Beyond the Basic Components and AutoTutor to the Emotional Realm

Our discussion so far has primarily focused on the basic AutoTutor technology that was developed between 1997 and 2005. We have made considerable enhancements to this core system over the last decade. For example, Hu and colleagues (Hu et al. 2009) developed a minimalistic version of AutoTutor called AutoTutor-Lite. We have also developed a version of AutoTutor, called AutoTutor-3D, that guides learners on using interactive simulations of physics microworlds (Graesser et al. 2005a; Jackson and Graesser 2006). In addition to these enhancements, AutoTutor has also inspired a number of natural-language learning systems that go beyond the basic components of dialog-based ITSs in a number of significant ways. These systems take the vision of learning by holding a conversation with a computer to a number of new realms. One system, Operation ARIES!, extends one-on-one dialogs to include one-on-two dialogs, which we call trialogs (Millis et al. 2011). There are systems that process learner responses using deeper NLP techniques (DeepTutor), train learners to deploy effective metacognitive and self-regulated learning behaviors (MetaTutor; Azevedo et al. (2008)), model expert instead of

novice human tutors (Guru Tutor; Olney et al. (2012)), and simulate human mentors during multiparty computer-supported learning activities (AutoMentor).

It is beyond the scope of this chapter to describe these next-generation ITSs in sufficient detail. Therefore, we focus on one subset of enhancements that attempt to move the learner models that focus on cognitive states into the realm of emotional states. One system called the Affective AutoTutor automatically adapts its dialogs based on whether the learner is bored, confused, or frustrated (D'Mello and Graesser *in press*). Another system, called the Gaze Tutor, has unique mechanisms to monitor and correct learners' disengagement behaviors by tracking their eye gaze in near real time (D'Mello et al. 2012). There is also a system that strategically plants seeds of cognitive disequilibrium and confusion in the minds of learners to engender deeper modes of thinking (D'Mello et al. *in press*). The remainder of this section describes these systems along with a discussion of how textual cues and characteristics of the emerging tutor-student discourse can be used to track emotions.

Responding to Learners' Emotional States

ITSs have come a long way towards providing individualized instruction to fit the cognitive needs of individual learners. However, ITSs can be more than mere cognitive machines, because emotions can have a major impact on engagement and learning (Calvo and D'Mello 2011; Pekrun and Stephens 2012). Affect sensitivity is important for ITSs that aspire to model human tutors because it has been claimed that accomplished tutors are able to recognize and appropriately respond to learners' emotional states (Lepper and Woolverton 2002). An affect-sensitive ITS would incorporate assessments of the learners' cognitive and affective states into its pedagogical and motivational strategies in order to keep learners engaged, boost self-confidence, heighten interest, and presumably maximize learning.

We have recently developed a version of AutoTutor (called the Affective Tutor or Supportive Tutor) that automatically detects learners' affective states (specifically boredom, confusion, frustration, and neutral) by monitoring conversational cues and other discourse features (discussed in more detail in the next section), gross body language, and facial features (D'Mello and Graesser 2010; D'Mello and Graesser *in press*). Each channel independently provides its own diagnosis of the learner's affective state. These individual diagnoses are combined with a decision-level fusion algorithm that selects a single affective state and a confidence value of the detection. The algorithm relies on a voting rule enhanced with a few simple heuristics.

The Affective AutoTutor has a set of production rules that map dynamic assessments of learners' cognitive and affective states with tutor actions to address the presence of boredom, confusion, and frustration. There are five parameters in the learner model and five parameters in the tutor model. The parameters in the learner model are: (a) the current affective state detected, (b) the confidence level of that affect classification, (c) the previous state detected, (d) a global measure of learner ability (dynamically updated throughout the session), and (e) the conceptual

quality of the learner's immediate response. The Affective AutoTutor incorporates this cognitive-affective assessment of the learner and responds with: (a) feedback for the current answer, (b) an affective statement, (c) the next dialog move, (d) an emotional display on the face of the tutor agent, and (e) an emotional modulation of the voice produced by AutoTutor's text-to-speech engine.

The tutor responds to the sensed emotions with empathetic, encouraging, and motivational dialog-moves and emotional displays. An emotion generator is also needed for the Affective AutoTutor because the system is expected to synthesize emotions as well. Therefore, the agent needs to speak with intonation that is properly integrated with facial expressions that display emotions. For example, an enthusiastic nod accompanied positive feedback after the learner provided a correct response. In contrast, there was a shaking of the head when the learner response was low quality, and a skeptical look when the tutor detected that the learner was hedging. What we see is that even a small set of emotion displays like in these examples given here can go a long way in conveying the tutor's emotions.

We tested the effectiveness of the Affective Tutor in improving learning over the Regular AutoTutor in a study where 84 learners completed two 30-min training sessions with either tutor (D'Mello et al. 2010b). The results indicated that the Affective tutor helped learning for low-domain knowledge learners during the second 30-min learning session. The Affective tutor was less effective at promoting learning for high-domain knowledge learners during the first 30-min session. Importantly, learning gains increased from Session 1 to Session 2 with the Affective tutor whereas they plateaued with the Regular tutor. Learners who interacted with the Affective Tutor also demonstrated higher performance on subsequent transfer tests than those who interacted with the Regular AutoTutor. A follow-up analysis into learners' perceptions of both tutors indicated that their perceptions of how closely the computer tutors resembled human tutors increased across learning sessions, was related to the quality of tutor feedback, and was a powerful predictor of learning (D'Mello and Graesser 2012). Interestingly, the increase was greater for the Affective tutor.

Sensing Emotion from Discourse Features

A one-on-one tutoring session with AutoTutor yields a rich trace of contextual information, characteristics of the learner, episodes during the coverage of the topic, and social dynamics between the tutor and learner. These conversational cues cover a broad and deep feature set that includes assessments of deep meaning, world knowledge, and pragmatic aspects of communication. They might also be predictive of learner affect. To test this hypothesis, after completing a learning session with the tutor, learners' emotions were judged by the learners themselves (self-judgments), untrained peers, and trained judges using a retrospective affect judgment protocol (Graesser et al. 2006). Several conversational features and discourse markers (collectively called dialog features or discourse features) were extracted from AutoTutor's log files and were utilized to infer learner affect. The dialog features

were computed for each learner-tutor turn (i.e., learner submits response, tutor provides feedback, tutor presents next question). They included *temporal* features (e.g., time on problem, response time), assessments of *response verbosity* (e.g., number of characters, speech act), assessments of the *conceptual quality* of the learner's response obtained by Latent Semantic Analysis (LSA), *conversational directness* (i.e., how much information the tutor is explicitly providing to the learner), and *tutor feedback* (negative, neutral, positive). The full list of features can be found in D'Mello et al. (2008).

Analyses regressing the presence or absence of each emotion on this set of dialog features illuminated a number of interesting patterns. Learners are prone to be confused early on when a problem or difficult question is presented, whereas they tend to become more bored as the tutoring session continues. They provide shorter answers when confused but are more verbose when they are actively engaged in the learning session. Confusion and frustration tend to occur when the tutor gives negative feedback, whereas positive feedback sometimes triggers delight. The tutor's hints (e.g., "What about X?") have an interesting impact on affect. Hints are indirect speech and are attempts by the computer tutor to lead the learner down a productive line of reasoning, but without being too obvious as to what the correct answer is. The hope is that the learners will actively construct answers and figure out the answers as opposed to the computer simply delivering information. However, the indirectness of hints is sometimes confusing or frustrating to learners, as was confirmed in our analyses. Another discovery is that learners get frustrated when they believe that they expressed something, but the computer fails to understand them and acknowledge their contributions.

We compared the accuracy by which 17 standard classifiers (e.g., Naive Bayes logistic regression, support vector machines) could detect the affective states from the dialog features (D'Mello et al. 2008). Machine learning experiments with ten-fold cross validation indicated that standard classifiers were moderately successful in discriminating the affective states of boredom, confusion, flow/engagement frustration, and neutral, yielding a peak accuracy of 42% with neutral (*chance* = 20%) and 54% without neutral (*chance* = 25%). Individual detections of boredom, confusion, flow, and frustration, when contrasted with neutral, had accuracies of 69%, 68%, 71%, and 78%, respectively (*chance* = 50%). Follow-up classification analyses that assessed the degree to which machine-generated affect labels correlated with affect judgments provided by humans revealed that human-machine agreement was on par with novice judges (self and peer) but quantitatively lower than trained judges. This dialog-based affect detector plays a very prominent role in the Affective AutoTutor system.

Sensing Emotion from Textual Cues

While the previous analysis focused primarily on discourse patterns and other contextual cues, we recently explored the possibility of predicting learner emotions

(boredom, flow/engagement, confusion, and frustration) by performing a deeper analysis of the text of the tutorial dialogs (D'Mello and Graesser in press). Transcripts from the tutorial dialogs were analyzed with four methods that included (a) identifying direct expressions of affect, (b) aligning the semantic content of learner responses to affective terms with Latent Semantic Analysis (LSA) (Landauer et al. 2007), (c) identifying psychological and linguistic terms that are predictive of affect with the *Linguistic Inquiry and Word Count* (Pennebaker et al. 2007), and (d) assessing cohesion relationships that might reveal learner affect with *Coh-Metrix* (Graesser et al. 2004b).

A detailed description of these four methods is beyond the scope of this chapter, so we proceed with brief descriptions only. The direct-expression models simply use regular expressions to detect affect. This approach is only viable if learners directly express affect in their responses (e.g., “This stuff is so *confusing*”). We used LSA to assess whether affect can be detected by semantically aligning learner responses (e.g., “This problem is a drag”) with affective terms (e.g., “boredom”). The Linguistic Inquiry and Word Count (LIWC) is a validated computer tool that analyzes bodies of text using dictionary-based categorization. LIWC-based affect-detection methods attempt to identify particular words that are expected to reveal the affective content in the text. Features of LIWC that were examined included affect terms (e.g., “sad,” “happy”), cognitive terms (e.g., “cause,” “maybe”), and several linguistic terms (e.g., pronouns, adverbs, prepositions). Finally, Coh-Metrix is a computational facility for automatically analyzing bodies of text on over 600 dimensions. Coh-Metrix provides over 100 measures of various types of cohesion, including co-reference, referential, causal, spatial, temporal, and structural cohesion. Coh-Metrix also has hundreds of measures of linguistic complexity (e.g., syntactic complexity), characteristics of words, and readability scores. The measures that were selected included co-reference cohesion (noun overlap, stem overlap, etc.), pronoun-referential cohesion, semantic cohesion, connectives, and other measures of word concreteness and readability.

Our results confirmed that learners rarely articulated their emotions to the tutor, thereby rendering the direct expression model inadequate. Nor did their responses semantically align with emotional terms. Instead, a deeper analysis of textual cues via LIWC and Coh-Metrix were needed to infer learner affect. Models constructed by regressing the proportional occurrence of each emotion on textual features derived from these methods yielded large effects ($R^2 = 38\%$). Although learners rarely articulate their emotions to the tutor and their responses mainly consist of words related to the content of the tutoring session (e.g., RAM, operating systems), some of the content words used, function words that connect content words, and the cohesiveness of their responses, ultimately revealed their affective states. The next step of this research is to develop finer-grained text-based affect detectors that operate on the turn level so that AutoTutor might be responsive to learner emotions without the need for expensive posture sensors and facial feature tracking.

Gaze Tutor: Responding to Learners' Disengagement Behaviors

It is generally acknowledged that engagement is an important precursor to learning, but until recently most ITSs have made little effort to meaningfully engage learners. Although learners might begin a learning session with an ITS with some level of interest and enthusiasm, boredom inevitably creeps in as the session progresses, when the novelty of the system and content fades, and when learners have difficulty comprehending the material (Larson and Richards 1991; Pekrun et al. 2010). When boredom strikes, any further instruction is essentially futile.

Attention to task-related thoughts is one critical precursor of engagement in a learning activity. Therefore, developing interventions that monitor periods of waning attention and attempt to encourage more productive use of *attentional resources* might be one promising way to increase engagement and promote learning. We tested this claim by developing a dialog-based learning system, called the Gaze Tutor, that used a commercial eye tracker to monitor learners' gaze patterns in order to identify when they were bored, disengaged, mind wandering, or zoning out (D'Mello et al. 2012). The tutor attempted to reengage learners with dialog moves that directed learners to reorient their attention towards the animated pedagogical agent embodying the tutor. These gaze-reorienting dialogs consisted of short direct messages that instructed learners to pay attention to the tutor or to certain parts of the interface.

We evaluated the efficacy of the gaze-reactive tutor in promoting learning, motivation, and engagement in an experiment where 48 learners were tutored on four biology topics with both gaze-reactive and non gaze-reactive (control condition) versions of the tutor. The results indicated that the gaze-sensitive dialogs were successful in dynamically reorienting learners' attentional patterns to the important areas of the interface. The effectiveness of gaze-orientation faded over time but did not entirely diminish. Although gaze-reactivity did not impact self-reported motivation and engagement, posttest scores for deep reasoning questions were higher when learners interacted with the gaze-sensitive tutor. Interestingly, individual differences in scholastic aptitude moderated the impact of gaze-reactivity on learning gains. Gaze-reactivity was associated with a small improvement in overall learning for learners with average aptitude, but learning gains were substantially higher for learners with high aptitude and somewhat lower for their counterparts. Future enhancements of the system include replacing the expensive eye-tracker with web-cameras, more fine-grained modeling of disengagement behaviors, a larger repertoire of context-specific gaze-reactive dialogs, and incorporating individual differences in selecting appropriate gaze-sensitive responses.

Planting Cognitive Disequilibrium and Confusion

A corollary of theories that highlight the benefits of impasses (VanLehn et al. 2003), cognitive conflict (Limón 2001), cognitive dissonance (Festinger 1957), and cognitive disequilibrium (Piaget 1952) is that events that productively confuse learners are beneficial to learning by providing learning opportunities. This is because learners need to engage in deep cognitive activities in order to resolve their confusion. Confusion resolution requires the learners to stop, think, engage in careful deliberation, problem solve, and revise their existing mental models. It is likely not the confusion itself, but the cognitive activities that accompany confusion resolution that are responsible for learning gains. There is some correlational evidence that is suggestive of a positive relationship between impasses, states of uncertainty, confusion and learning gains (Craig et al. 2004; D’Mello and Graesser 2011; Graesser et al. 2007a; VanLehn et al. 2003), but confusion has not been causally linked to positive learning outcomes.

We tested the hypothesis that confusion can impact learning by modifying ARIES (Millis et al. 2011) to systematically induce confusion (D’Mello et al. *in press*). We did this by manipulating whether or not the tutor agent and the learner agent contradicted each other during the trialog by expressing points that were incorrect. The human learner was asked to intervene after each point of possible contradiction; the agents turned to the human and asked “So what would your decision be, <name of learner>?” If the human learner experiences uncertainty and is confused, this should be reflected in the incorrectness/uncertainty of the human’s answer. Uncertainty is a likely opportunity to scaffold deep comprehension by forcing learners to stop and think.

The data confirmed that the contradictions and false information had an impact on learners’ confusion. Importantly, levels of confusion moderate the impact of the contradictions on learning. Specifically, the contradictions had no effect on learning when learners were not confused by the manipulations, whereas performance on multiple-choice posttests and on transfer tests was substantially higher when the contradictions were successful in confusing learners. This suggests that there are some benefits to inducing confusion if learners are *productively* instead of *hopelessly* confused. By productive confusion, we mean that the source of the confusion is closely linked to the content of the learning session, the learner attempts to resolve their confusion, and the learning environment provides necessary scaffolds to facilitate the confusion resolution process.

Some Open Issues for Dialog-Based ITSs

This section discusses some of the recurring issues that arise in research on dialog-based ITSs. We shed light on four issues that we deem critical, knowing fully well that these four items are only a small subset of the open issues in the field. These

items can be framed in terms of the following four questions: (a) Is spoken input more advantageous than typed input? (b) is imperfect natural-language processing good enough for meaningful interactions? (c) do the merits of dialog-based ITSs lie in the medium or the message? and (d) when are natural-language dialogs beneficial for learning?

Is Spoken Input More Advantageous than Typed Input?

The recent advance of commercially available spoken interfaces raises the question of whether there are advantages of learners speaking their responses instead of typing them in. A *speech facilitation* hypothesis predicts that spoken input will increase learning, whereas a *text facilitation* hypothesis predicts typed input will be superior. Alternatively, a *modality equivalence hypothesis* claims that learning gains will be equivalent because the advantages and disadvantages offered by both input modalities will cancel each other out. Spoken language is, of course, easier to produce because the expression gap between thought and speech is much less than the gap between thought and writing. Because of the ease of spoken responses, the volume of content is typically longer when spoken than typed. Given that learning is correlated with the volume of responses by the learners (Chi et al. 2004), following a constructivist framework (Dalgarno 2001; Moshman 1982), it would be predicted that spoken responses would yield higher learning gains. On the other hand, typed responses offer some advantages as well. While spoken utterances are evanescent, i.e., they disappear shortly after they are spoken, typed contribution remains on the screen for learners to evaluate and revise. The additional time to reflect on their responses affords added perceptual processing, rereading, and memory encoding, thereby increasing learning gains if learners use these textual representations to process the material more deeply. Therefore, there appears to be a tradeoff between the ease of a learner producing a spoken response and the quality of the learner's typed response.

Previous research that has tested these hypotheses within the context of dialog-based ITSs enabled with automated speech-recognition systems has supported the modality-equivalence hypothesis (D'Mello et al. 2010a; Litman et al. 2006). The one study that compared spoken versus typed input during human-human tutoring sessions offered evidence in support of the speech facilitation hypothesis (Litman et al. 2006). There were many differences between the human-human and human-computer tutorial sessions that could potentially explain the discrepant findings. However, the fact that the automated speech recognition systems used by the dialog-based ITSs had substantial error rates that were detected by learners might represent one of the most significant factors. This concern was addressed in two additional experiments that used a Wizard of Oz procedure involving a human who intercepted the learner's speech and transcribed the utterance before submitting it to the computer tutor (D'Mello et al. 2011). This afforded relatively error-free speech recognition, thereby correcting a confound with the previous experiments. The

results of these two experiments also resulted in null effect, once again supporting the modality equivalence hypothesis. The only exception was that highly motivated learners reported lower cognitive load and demonstrated increased learning when typing compared to speaking. Apart from this exception that warrants replication, the four experiments that compared spoken and typed responses with dialog-based ITSs suggest that the *content* is more important than the *mode* of communication (Graesser et al. 2003), a theme which we return to below.

Is Imperfect Natural-Language Processing Good Enough?

Critics of natural-language interfaces have sometimes argued that language is too vague, imprecise, and ambiguous for computers to deeply understand. According to these skeptics, decoding language requires vast amounts of world knowledge and socially and culturally specific information that far exceeds the scope of current natural-language processing techniques. Although it is clear that human-like speech and language comprehension clearly surpasses what can be achieved by a computer, the dialog-based ITSs that have been developed suggest that much can be achieved by imperfect speech recognition and shallow language understanding. Examples of systems that are effective despite ASR and NLU imperfections include Litman's ITSPOKE system (Litman et al. 2006), Mostow et al.'s Reading Tutor (Mostow and Aist 2001), the Scot system (Pon-Barry et al. 2004; Schultz et al. 2003), the Tactical Language and Culture Training System (Johnson and Valente 2008), and AutoTutor (D'Mello et al. 2010a).

Of course imperfect ASR and NLU are unlikely to suffice for all applications, thereby raising the question of when imperfect natural-language processing capabilities are good enough to sustain effective interactions. One way to predict the suitability of natural-language interfaces with imperfect capabilities is to consider the interaction between the expected precision of the system and the degree of common ground (shared knowledge) between the user and the interface. By expected precision we mean the accuracy of comprehending the user's utterance. Common ground refers to the degree of shared knowledge between the user and the interface (Clark 1996). According to this two dimensional framework, the simplest cases occur when the expected precision is high. In these situations, the quality of the interaction is not expected to be adversely affected by the lack of common ground between the conversational participants, because any deficits in shared knowledge can be easily corrected through a conversation that both partners can effortlessly comprehend.

The more interesting situations arise when the expected precision is low. If the common ground between the conversational partners is high, the conversation will ultimately break down because one of the participants will be aware of the other's natural language understanding deficits. But what about dialog-based ITSs with low to moderate precision? Our position is that these systems can be functional despite natural-language understanding errors provided two conditions are met:

First, while the precision of the system can be low, it should be significantly greater than zero, i.e., the system should be able to recognize some meaning even if only at shallow levels of comprehension (e.g., content words, key phrases). Second, the degree of common ground between the learner and the ITS should also be low. Low domain-knowledge learners rarely answer questions with complete answers or even complete sentences. Instead, they utter a few key phrases, make a few statements that are close to their vernacular discourse and for the most part rarely articulate academically precise statements. Since low domain-knowledge learners share a small degree of common ground with the tutor, they can be expected to be oblivious to moderate natural-language understanding errors. It should be noted that this focus on low domain-knowledge learners does not reduce the impact of ITSs, because it is well known that ITSs are typically more effective for low domain-knowledge learners (VanLehn et al. 2007). Furthermore, the low domain-knowledge learners are the ones in need of one-on-one tutoring.

The above discussion raises the important question of how to design dialog-based ITSs that can compensate for partial failure in the quality of interpreting the learner's responses. A solution that we have explored involves a combination of shallow natural language processing and soft, constraint-satisfaction models. Shallow NLP techniques differ from deep NLP approaches as they do not perform a thorough linguistic analysis in understanding a user's utterance. Instead, shallow NLP approaches focus more on the semantics than on the syntax of an utterance. Shallow NLP techniques can compensate for partial failures in the user's utterance if the failure is handled with soft-constraint satisfaction approaches. According to soft-constraint satisfaction (SCS) models, the performance of an intelligent system should not rely on the integrity of any one level or module, but rather should reflect the confluence of several levels/modules that are statistically combined. For example, natural language understanding involves a multilevel computational analysis including phonemes, morphemes, words, syntax, sentence semantics, discourse, pragmatics, world knowledge, and genre. According to a SCS model, when the fidelity at one level fails, the other levels fill in. When there is an ambiguity at one level, the other levels resolve the ambiguity (i.e., context can be recruited to resolve ambiguity in word meaning).

Do the Merits of Dialog-Based ITSs Lie in the Medium or the Message?

The basic one-on-one tutorial dialogs that model learner knowledge that attempt to eke out what the learner knows, identify and correct misconceptions, and help learners construct mental models, can be augmented in a number of ways. Explanations can be accompanied by animations of dynamical processes (e.g., Moreno and Mayer 2007). Learners can use simulations to model complex systems and observe how different inputs affect outputs (e.g., Graesser et al. 2005a). The

content of the tutor's utterances can be delivered by animated pedagogical agents with sophisticated facial expressions, speech, and gestures (e.g., Johnson et al. 2000). Learners can speak their responses to the tutor instead of typing them in (e.g., Litman et al. 2006). Progress can be directly conveyed to the learner by graphically presenting the results of the learner model to the student (e.g., Jackson and Graesser 2007). The interaction can be made more game-like by displaying points, levels, and challenge problems (e.g., Jackson et al. *in press*). In other words, the basic one-on-one interaction can be decorated in a number of ways.

To what extent do these *decorations* increase learning gains beyond what can be obtained from text-only tutorial dialogs? Although it is too early to say for many other systems, our experience with AutoTutor indicates that they do not add much. For example, comparisons of the effectiveness of presenting AutoTutor's speech acts in print instead of the animated agent yielded a very small 0.13 sigma effect (Graesser et al. 2003). The interactive AutoTutor-3D version with simulations has a 0.22 effect size over the normal conversational AutoTutor (Graesser et al. 2005a). As described above, comparisons between spoken and typed input has consistently resulted in null effects (D'Mello et al. 2010a, 2011; Litman et al. 2006). Similarly, versions that varied different levels of feedback had negligible impacts on learning gains (Jackson and Graesser 2007).

The fact that these different enhancements have generally yielded negligible to small effects in learning gains, suggests that the *content* is more important than the *medium* of communication. Put simply, when it comes to learning, "*The medium is not the message—the message is the message.*" We suspect that these enhancements might service alternate goals such as promoting long-term engagement and task persistence, a feat that appears to be difficult to sustain via the tedium of text-based interactions.

When Are Natural Language Dialogs Beneficial for Learning?

This chapter, thus far, has extolled the virtues of dialogs-based ITSs, with an emphasis on the ones we have developed and tested. But it is important to realize that these systems are not the panacea to the problems of learning. As with any complex phenomenon, there are conditions where dialog-based ITSs are *effective* and conditions where they are not. There is still considerable research needed before one can prescribe conditions when dialog-based ITSs are maximally effective; however, some recent research has uncovered some important insights.

VanLehn et al. (2007) conducted seven experiments that compared the effectiveness of dialog-based ITSs to expert human tutors, reading text, and doing nothing. They discovered that the alignment of learner knowledge with the content of the instruction was the best predictor of outcomes. Dialog-based ITSs showed significant advantages over reading text when there was misalignment between prior knowledge and complexity of the content. Specifically, novice learners benefitted from ITSs when the content was at the intermediate level. However, the tutorial

dialogs yielded null effects (compared to reading text) when prior knowledge and content complexity aligned (i.e., when novices read materials designed for novices and intermediates read materials designed for intermediates).

The results of our experiments with the Affective AutoTutor and the gaze-sensitive tutor provide additional boundary conditions on the effectiveness of these systems. The Affective AutoTutor was more effective than the regular tutor for low domain-knowledge learners in the second session, but not the first session. Furthermore, learners with more knowledge never benefited from the motivational messages of the Affective AutoTutor. These learners don't need the emotional support, but rather they need to go directly to the content. Therefore, there appears to be a tradeoff to quick support and empathy compared to no affect sensitivity.

Interestingly, a reverse pattern was obtained from the gaze-sensitive tutor. The somewhat more stern, gaze-reorienting messages were more effective in promoting learning for the gifted learners than their counterparts. Similar findings have also been noted for tutors who either adopt politeness principles (Brown and Levinson 1987) when formulating their dialog moves compared to more direct and less socially intelligent styles of interaction (McLaren et al. 2011; Wang et al. 2008). Learners who make the most errors appear to benefit from tutors with a degree of social intelligence and polite responses.

These three lines of research are suggestive of an important aptitude-treatment interaction. Struggling learners might benefit from encouraging, motivational, empathetic, and polite interactions when they encounter troublesome impasses, get stuck, and risk disengagement. On the other hand, more direct and assertive interaction styles might be beneficial for the more gifted learners.

Concluding Remarks

With much pomp and pomposity, US. Industries Inc. unveiled their Auto Tutor machine at the 1964 World's Fair. Though quite advanced for its time, this was a rather crude machine with buttons and 35 mm slides. Despite its rudimentary interaction mechanisms, its main advantage was that it allowed learners to advance at their own pace, thereby benefitting both beginners and advanced students. In some ways, much has changed over the last few decades. We have shown that it is feasible to program digital computers to effectively mirror the nuances of one-on-one human tutorial dialogs. Despite imperfect and imprecise natural-language understanding capabilities, these systems can provide meaningful interactions that yield learning gains that rival the experts. However, in other ways, these systems are still lacking in some fundamental aspects. Most importantly, they are a bit lackluster, conversationally brittle, and somewhat tedious to use, all factors that do not facilitate long-term use and acceptance. Put rather crudely, our systems work, but the average student does not want to use them over long periods of time. While no one would deny that an increase in learning ability is an important criterion of success, it can be argued that sustaining engagement with the student over long-term interactions is

likewise of equal importance. Unfortunately, until recently the challenge of keeping learners engaged over extended interactions has been a somewhat neglected area of research. Therefore, in our view, the next grand challenge of dialog-based ITSs is to implement mechanisms that promote long-term learning by maintaining levels of interest and engagement that persist after the initial novelty of the systems fades. Indeed, the promise of dialog-based ITSs will only be realized once an optimal balance between the dual goals of engagement and learning is achieved.

Acknowledgements This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1235958) and Institute of Education Sciences (R305B070349). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF or IES.

References

- Alibali MW, Nathan MJ (2012) Embodiment in mathematics teaching and learning: a view from students' and teachers' gestures. *J Learn Sci* 21(2):247–286
- Arnott E, Hastings P, Allbritton D (2008) Research methods tutor: evaluation of a dialogue-based tutoring system in the classroom. *Behav Res Methods* 40(3):694–698
- Azevedo R, Witherspoon A, Graesser A, McNamara D, Rus V, Cai Z, Lintean M, Siler E (2008) MetaTutor: an adaptive hypermedia system for training and fostering self-regulated learning about complex science topics. In: Pirrone R, Azevedo R, Biswas G (eds) *Papers from the AAAI Fall Symposium on cognitive and metacognitive educational systems*. AAAI Press, Menlo Park, CA, pp 14–19
- Baayen R, Piepenbrock R, Gulikers L (1995) *The CELEX lexical database* (CD-ROM). University of Pennsylvania, Philadelphia
- Baylor AL, Kim Y (2005) Simulating instructional roles through pedagogical agents. *Int J Artif Intell Educ* 15(2):95–115
- Biggs J (1996) Enhancing teaching through constructive alignment. *High Educ* 32(3):347–364
- Bloom B (1956) *Taxonomy of educational objectives: the classification of educational goals. Handbook I: cognitive domain*. McKay, New York
- Bransford J, Goldman S, Vye N (1991) Making a difference in people's ability to think: reflections on a decade of work and some hopes for the future. In: Sternberg R, Okagaki L (eds) *Influences on children*. Erlbaum, Hillsdale, NJ, pp 147–180
- Brennan S, Williams M (1995) The feeling of another's knowing—prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *J Mem Lang* 34(3):383–398
- Brown P, Levinson S (1987) *Politeness: some universals in language usage*. Cambridge University Press, Cambridge
- Brown A, Ellery S, Campione J (1998) Creating zones of proximal development electronically. In: Greeno J, Goldman S (eds) *Thinking practices in mathematics and science learning*. Lawrence Erlbaum, Mahwah, NJ, pp 341–367
- Cai Z, Graesser A, Forsyth C, Burkett C, Millis K, Wallace P, Halpern D, Butler H (2011) Trialog in ARIES: user input assessment in an intelligent tutoring system. In: Chen W, Li S (eds) *Proceedings of the 3rd IEEE international conference on intelligent computing and intelligent systems*. IEEE Press, Guangzhou
- Calvo R, D'Mello S (2011) *New perspectives on affect and learning technologies*. Springer, New York
- Chi M, Deleeuw N, Chiu M, Lavancher C (1994) Eliciting self-explanations improves understanding. *Cognit Sci* 18(3):439–477

- Chi M, Siler S, Jeong H (2004) Can tutors monitor students' understanding accurately? *Cogn Instr* 22(3):363–387
- Clark H (1996) *Using language*. Cambridge University Press, Cambridge
- Conati C, Gertner A, VanLehn K (2002) Using Bayesian networks to manage uncertainty in student modeling. *User Model User Adapted Interact* 12(4):371–417
- Corbett AT, Anderson JR (1994) Knowledge tracing—modeling the acquisition of procedural knowledge. *User Model User Adapted Interact* 4(4):253–278
- Craig S, Graesser A, Sullins J, Gholson J (2004) Affect and learning: an exploratory look into the role of affect in learning. *J Educ Media* 29:241–250. doi:[10.1080/1358165042000283101](https://doi.org/10.1080/1358165042000283101)
- D'Mello SK, Graesser AC (2012) Language and Discourse are Powerful Signals of Student Emotions during Tutoring. *IEEE Transactions on Learning Technologies*, 5(4):304–317
- D'Mello SK, Graesser AC (2012) AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. *ACM Transactions on Interactive Intelligent Systems* 2(4), 23:2–23:39
- D'Mello S, Graesser A (2011) The half-life of cognitive-affective states during complex learning. *Cogn Emotion* 25(7):1299–1308
- D'Mello S, Graesser A (2012) Malleability of students' perceptions of an affect-sensitive tutor and its influence on learning. In: Youngblood G, McCarthy P (eds) *Proceedings of 25th Florida Artificial Intelligence Research Society Conference*. AAAI Press, Menlo Park, CA, pp 432–437
- D'Mello S, Olney A, Person N (2010) Mining collaborative patterns in tutorial dialogues. *J Educ Data Mining* 2(1):1–37
- D'Mello S, Lehman S, Pekrun R, Graesser A (in press) Confusion can be beneficial for learning. *Learn Instruct*
- Dalgarno B (2001) Interpretations of constructivism and consequences for computer assisted learning. *Br J Educ Technol* 32(2):183–194
- D'Mello S, Graesser A (2010) Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Model User Adapted Interact* 20(2):147–187
- D'Mello S, Craig S, Witherspoon A, McDaniel B, Graesser A (2008) Automatic detection of learner's affect from conversational cues. *User Model User Adapted Interact* 18(1–2):45–80
- D'Mello S, King B, Graesser A (2010a) Towards spoken human-computer tutorial dialogues. *Human Comput Interact* 25(4):289–323
- D'Mello S, Lehman B, Sullins J, Daigle R, Combs R, Vogt K, Perkins L, Graesser A (2010b) A time for emoting: when affect-sensitivity is and isn't effective at promoting deep learning. In: Kay J, Alevan V (eds) *Proceedings of the 10th international conference on intelligent tutoring systems*. Springer, Berlin, pp 245–254
- D'Mello S, Dowell N, Graesser A (2011) Does it really matter whether students' contributions are spoken versus typed in an intelligent tutoring system with natural language? *J Exp Psychol Appl* 17(1):1–17
- D'Mello S, Olney A, Williams C, Hays P (2012) Gaze tutor: a gaze-reactive intelligent tutoring system. *Int J Human Comput Stud* 70(5):377–398
- Evens MW, Chang RC, Lee YH, Shim LS, Woo CW, Zhang Y, Michael JA, Rovick AA (1997) CIRCSIM-Tutor: an intelligent tutoring system using natural language dialogue. In *Proceedings of the fifth conference on applied natural language processing*. Stroudsburg, PA, Association for Computational Linguistics, pp 13–14
- Festinger L (1957) *A theory of cognitive dissonance*. Stanford University Press, Stanford, CA
- Fox B (1993) *The human tutorial dialogue project*. Lawrence Erlbaum Associates, Hillsdale, NJ
- Freedman R (1996) *Interaction of discourse planning, instructional planning, and dialogue management in an interactive tutoring system*. Doctoral Dissertation, Northwestern University, Evanston, IL.
- Gertner A, VanLehn K (2000) Andes: a coached problem solving environment for physics. In: Gauthier G, Frasson C, VanLehn K (eds) *Proceedings of the 5th international conference on intelligent tutoring systems*. Springer, Berlin, pp 133–142

- Goldin-Meadow S (2003) *Hearing gesture: how our hands help us think*. Harvard University Press, Cambridge, MA
- Graesser A, Person N (1994) Question asking during tutoring. *Am Educ Res J* 31(1):104–137
- Graesser A, Person N, Magliano J (1995) Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Appl Cogn Psychol* 9(6):495–522
- Graesser A, Wiemer-Hastings K, Wiemer-Hastings P, Kreuz R (1999) AutoTutor: a simulation of a human tutor. *Cogn Syst Res* 1(1):35–51
- Graesser A, Wiemer-Hastings P, Wiemer-Hastings K, Harter D, Person N (2000) Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interact Learn Environ* 8(2):129–147
- Graesser A, VanLehn K, Rose CP, Jordan PW, Harter D (2001) Intelligent tutoring systems with conversational dialogue. *AI Mag* 22(4):39–51
- Graesser A, Moreno K, Marineau J, Adcock A, Olney A, Person N (2003) AutoTutor improves deep learning of computer literacy: is it the dialog or the talking head? In: Hoppe U, Verdejo F, Kay J (eds) *Proceedings of the 11th international conference on artificial intelligence in education*. Ios Press, Amsterdam, pp 47–54
- Graesser A, Lu SL, Jackson G, Mitchell H, Ventura M, Olney A, Louwerse M (2004a) AutoTutor: a tutor with dialogue in natural language. *Behav Res Methods Instrum Comput* 36:180–193
- Graesser A, McNamara D, Louwerse M, Cai Z (2004b) Coh-Metrix: analysis of text on cohesion and language. *Behav Res Methods Instrum Comput* 36:193–202
- Graesser A, Chipman P, Haynes B, Olney A (2005a) AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Trans Educ* 48(4):612–618. doi:[10.1109/TE.2005.856149](https://doi.org/10.1109/TE.2005.856149)
- Graesser AC, Hu X, McNamara DS (2005b) Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics. In: Healy AF (ed) *Experimental cognitive psychology and its applications: Festschrift in honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*. American Psychological Association, Washington, DC
- Graesser A, McDaniel B, Chipman P, Witherspoon A, D’Mello S, Gholson B (2006) Detection of emotions during learning with AutoTutor. In: Sun R, Miyake N (eds) *Proceedings of the 28th annual conference of the Cognitive Science Society*. Cognitive Science Society, Austin, TX, pp 285–290
- Graesser A, Chipman P, King B, McDaniel B, D’Mello S (2007a) Emotions and learning with AutoTutor. In: Luckin R, Koedinger K, Greer J (eds) *Proceedings of the 13th international conference on artificial intelligence in education*. Ios Press, Amsterdam, pp 569–571
- Graesser A, Penumatsa P, Ventura M, Cai Z, Hu X (2007b) Using LSA in AutoTutor: learning through mixed-initiative dialogue in natural language. In: Landauer T, McNamara D, Dennis S, Kintsch W (eds) *Handbook of latent semantic analysis*. Erlbaum, Mahwah, NJ, pp 243–262
- Graesser A, D’Mello S, Cade W (2009a) Instruction based on tutoring. In: Mayer RE, Alexander PA (eds) *Handbook of research on learning and instruction*. New York, Routledge, pp 408–426
- Graesser AC, D’Mello S, Person N (2009b) Meta-knowledge in tutoring. In: Hacker DJ, Dunlosky J, Graesser AC (eds) *Handbook of metacognition in education*. Erlbaum, Mahwah, NJ, pp 361–412
- Graesser AC, Conley MW, Olney AM (2012) Intelligent tutoring systems. In: Harris KR, Graham S, Urdan T (eds) *The APA educational psychology handbook, vol 3*. American Psychological Association, Washington, DC, pp 451–473
- Gratch J, Rickel J, André E, Cassell J, Petajan E, Badler N (2002) Creating interactive virtual humans: some assembly required. *IEEE Intell Syst* 17(4):54–63
- Hestenes D, Wells M, Swackhamer G (1992) Force concept inventory. *Phys Teach* 30:141–158
- Hu X, Cai Z, Wiemer-Hastings P, Graesser A, McNamara D (2007) Strengths, limitations, and extensions of LSA. In: Landauer T, McNamara D, Dennis S, Kintsch W (eds) *Handbook of latent semantic analysis*. Lawrence Erlbaum Associates, Mahwah, NJ, pp 401–426
- Hu X, Cai Z, Han L, Craig SD, Wang T, Graesser AC (2009) AutoTutor Lite. In: Dimitrova V, Mizoguchi R, du Boulay B, Graesser A (eds) *Proceedings of the 14th international conference on artificial intelligence in education*. Ios Press, Amsterdam, p 802

- Jackson G, Graesser A (2006) Applications of human tutorial dialog in AutoTutor: an intelligent tutoring system. *Revista Signos* 39(60):31–48
- Jackson GT, Graesser AC (2007) Content matters: an investigation of feedback categories within an ITS. In: Luckin R, Koedinger K, Green JJ (eds) *Proceedings of the 13th international conference on artificial intelligence in education*. Ios Press, Amsterdam, pp 127–134
- Jackson GT, Dempsey KB, McNamara DS (2012) Game-based practice in reading strategy tutoring system: Showdown in iSTART-ME. In H. Reinders (ed) *Computer games* (pp 115–138). Bristol, UK: Multilingual Matters
- Johnson W, Valente L (2008, July) Tactical language and culture training systems: using artificial intelligence to teach foreign languages and cultures. Paper presented at the Proceedings of the 20th national artificial intelligence conference, Menlo Park, CA
- Johnson W, Rickel J, Lester J (2000) Animated pedagogical agents: face-to-face interaction in interactive learning environments. *Int J Artif Intell Educ* 11:47–78
- Landauer T, McNamara D, Dennis S, Kintsch W (eds) (2007) *Handbook of latent semantic analysis*. Erlbaum, Mahwah, NJ
- Larson RW, Richards MH (1991) Boredom in the middle school years—blaming schools versus blaming students. *Am J Educ* 99(4):418–443
- Lepper M, Woolverton M (2002) The wisdom of practice: lessons learned from the study of highly effective tutors. In: Aronson J (ed) *Improving academic achievement: impact of psychological factors on education*. Academic, Orlando, FL, pp 135–158
- Limón M (2001) On the cognitive conflict as an instructional strategy for conceptual change: a critical appraisal. *Learn Instruct* 11(4–5):357–380. doi:[10.1016/s0959-4752\(00\)00037-2](https://doi.org/10.1016/s0959-4752(00)00037-2)
- Litman D, Rose C, Forbes-Riley K, VanLehn K, Bhembé D, Silliman S (2006) Spoken versus typed human and computer dialogue tutoring. *Int J Artif Intell Educ* 16(2):145–170
- McLaren BM, DeLeeuw KE, Mayer RE (2011) A politeness effect in learning with web-based intelligent tutors. *Int J Human Comput Stud* 69(1–2):70–79. doi:[10.1016/j.ijhcs.2010.09.001](https://doi.org/10.1016/j.ijhcs.2010.09.001)
- Millis K, Forsyth C, Butler H, Wallace P, Graesser A, Halpern D (2011) Operation ARIES! A serious game for teaching scientific inquiry. In: Ma M, Oikonomou A, Lakhmi J (eds) *Serious games and edutainment applications*. Springer, London, pp 169–196
- Moreno R, Mayer R (2007) Interactive multimodal learning environments. *Educ Psychol Rev* 19(3):309–326. doi:[10.1007/s10648-007-9047-2](https://doi.org/10.1007/s10648-007-9047-2)
- Moshman D (1982) Exogenous, endogenous, and dialectical constructivism. *Dev Rev* 2(4):371–384
- Mostow J, Aist G (2001) Evaluating tutors that listen: an overview of project LISTEN. In: Forbus K, Feltovich P (eds) *Smart machines in education: the coming revolution in educational technology*. MIT Press, Cambridge, MA, pp 169–234
- Olney A, Louwerse M, Mathews E, Marineau J, Hite-Mitchell H, Graesser A (2003) Utterance classification in AutoTutor. Paper presented at the Proceedings of the human language technology conference of the North American chapter of the Association for Computational Linguistics, Edmonton, Canada.
- Olney A, D'Mello A, Person N, Cade W, Hays P, Williams C, Lehman B, Graesser A (2012) Guru: a computer tutor that models expert human tutors. In: Cerri S, Clancey W, Papadourakis G, Panourgia K (eds) *Proceedings of the 11th international conference on intelligent tutoring systems*. Springer, Berlin, pp 256–261
- Pekrun R, Stephens EJ (2012) Academic emotions. In: Harris K, Graham S, Urdan T, Graham S, Royer J, Zeidner M (eds) *APA educational psychology handbook, Vol 2: individual differences and cultural and contextual factors*. American Psychological Association, Washington, DC, pp 3–31
- Pekrun R, Goetz T, Daniels L, Stupnisky RH, Perry R (2010) Boredom in achievement settings: exploring control-value antecedents and performance outcomes of a neglected emotion. *J Educ Psychol* 102(3):531–549. doi:[10.1037/a0019243](https://doi.org/10.1037/a0019243)
- Pennebaker J, Chung C, Ireland M, Gonzales A, Booth R (2007) *The development and psychometric properties of LIWC2007*. The University of Texas at Austin and the University of Auckland, New Zealand, Austin, TX

- Person N, Graesser A (2002) Human or computer? AutoTutor in a Bystander Turing Test. In: Cerri S, Gouarderes G, Paraguacu F (eds) Proceedings of the 6th international conference on intelligent tutoring systems. Springer, Berlin, pp 821–830
- Piaget J (1952) The origins of intelligence. International University Press, New York
- Pon-Barry H, Clark B, Schultz K, Bratt EO, Peters S (2004) Advantages of spoken language interaction in dialogue-based intelligent tutoring systems. In: Lester J, Vicari R, Paraguacu F (eds) Proceedings of seventh international conference on intelligent tutoring systems. Springer, Berlin, pp 390–400
- Posota J, Massey D, Mutter S (1988) Intelligent tutoring systems: lessons learned. Lawrence Erlbaum Associates, Hillsdale, NJ
- Rus V, Graesser A (2007) Lexico-syntactic subsumption for textual entailment. In: Nicolov N, Bontcheva K, Angelova G, Mitkov R (eds) Recent advances in natural-language processing IV: Selected papers from RANLP 2005. John Benjamins Publishing Company, Amsterdam, pp 187–196
- Sacks H, Schegloff EA, Jefferson G (1974) A simplest systematics for the organization of turn-taking for conversation. *Language* 50(4):696–735
- Schultz K, Bratt E, Clark B, Peters S, Pon-Barry H, Treeratpituk P (2003) A scalable, reusable spoke conversational tutor: Scot. Paper presented at the Workshop on Tutorial Dialogue Systems: with a view toward the classroom. In conjunction with the 11th international conference on artificial intelligence in education, Sydney, Australia
- Sleeman D, Brown J (eds) (1982) Intelligent tutoring systems. Academic, New York
- VanLehn K (2006) The behavior of tutoring systems. *Int J Artif Intell Educ* 16(3):227–265
- VanLehn K (2011) The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ Psychol* 46(4):197–221
- VanLehn K, Jordan P, Rose C, Bhembé D, Bottner M, Gaydos A (2002) The architecture of Why2-Atlas: a coach for qualitative physics essay writing. In: Cerri SA, Gouarderes G, Paraguacu F (eds) Proceedings of the sixth international conference on intelligent tutoring system. Springer, Berlin, pp 158–167
- VanLehn K, Siler S, Murray C, Yamauchi T, Baggett W (2003) Why do only some events cause learning during human tutoring? *Cogn Instruct* 21(3):209–249. doi:[10.1207/S1532690XCI2103_01](https://doi.org/10.1207/S1532690XCI2103_01)
- VanLehn K, Graesser A, Jackson G, Jordan P, Olney A, Rose CP (2007) When are tutorial dialogues more effective than reading? *Cognit Sci* 31(1):3–62
- Vygotsky L (1978) *Mind in society: the development of higher psychological processes*. Harvard University Press, Cambridge, MA
- Vygotsky L (1986) *Thought and language*. MIT Press, Cambridge, MA
- Wang N, Johnson WL, Mayer RE, Rizzo P, Shaw E, Collins H (2008) The politeness effect: pedagogical agents and learning outcomes. *Int J Human Comput Stud* 66(2):98–112. doi:[10.1016/j.ijhcs.2007.09.003](https://doi.org/10.1016/j.ijhcs.2007.09.003)
- Ward W, Cole R, Bolaños D, Buchenroth-Martin C, Svirsky E, Vuuren SV, Weston T, Zheng J, Becker L (2011) My science tutor: a conversational multimedia virtual tutor for elementary school science. *ACM Trans Speech Lang Process* 7(4):18
- Wolf B (2009) *Building intelligent interactive tutors*. Morgan Kaufmann Publishers, Burlington, MA

Chapter 12

TCAD: Vocabulary Acquisition Tool for Motivating Bilingual Pupils with Hearing Impairment in Learning English

Santichai Wicha, Bernadette Sharp, Anthony S. Atkins, Pradorn Sureephong,
and Nopasit Chakpitak

Abstract This chapter describes a software system, Total Communication Animated Dictionary (TCAD), developed to support the acquisition and retention of English vocabulary of pupils with hearing impairment in their bilingual education program. TCAD employs a variety of visual aids and animated features (e.g. lip reading, finger spelling, sign and animation) to optimise their vocabulary acquisition. To enhance their retention TCAD includes stories and games which are based on the learnt vocabulary and set within the local context and culture of primary school pupils. TCAD is tested with a set of primary pupils with hearing-impaired from a school based in Chiang Mai, Thailand. The initial investigation showed promising results in the acquisition and retention of English vocabulary. Teachers have noted an improved motivation and enthusiasm among their pupils in learning English as TCAD has provided a forum for social interaction and communication among the pupils.

S. Wicha

Faculty of Computing, Engineering and Technology, Staffordshire University,
Octagon Stafford, UK
e-mail: santichai@kic.camt.info

School of Information Technology, Mae Fah Luang University, Chiang Rai, Thailand
College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand

B. Sharp (✉) • A.S. Atkins

Faculty of Computing, Engineering and Technology, Staffordshire University,
Octagon Stafford, UK
e-mail: b.sharp@staffs.ac.uk; a.s.atkins@staffs.ac.uk

P. Sureephong • N. Chakpitak

College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand
e-mail: dorn@camt.info; nopasit@camt.info

Introduction

The primary focus of this research is the acquisition and retention of English vocabulary to improve the reading and writing performance of Thai primary school children with hearing impairment. The first important challenge is to develop a system which motivates pupils to make vocabulary learning a fun activity and thus improve their vocabulary retention. The second challenge is to ensure that the system is responding to the pupil's needs by providing them with a meaningful and accessible exposure to the English language and culture. This chapter describes the development of an animated software system, Total Communication Animated Dictionary (TCAD), designed to support the teaching of English as a second language to these pupils with hearing impairment. The review of literature reveals positive results from using computers to help develop vocabulary knowledge in pupils with autism (Bosseler and Massaro 2003) and with high school students with a learning disability (Gersten et al. 1987). This has inspired us to develop TCAD, which is based on the total communication and situated learning theories. TCAD embraces diverse communication modes (visual, written and animation) and methods (e.g. signs, gestures and finger spelling) to optimise the language acquisition of hearing-impaired learners. In this project, hearing-impaired is the term used to denote hearing loss regardless of the degree of severity.

It is well documented that literacy is the strongest predictor of success of hearing-impaired children's education, and it relies heavily on vocabulary knowledge. However vocabulary development for the hearing-impaired children is quantitatively reduced as compared to that of typical hearing peers (Moore 1978) as they have smaller lexicons, and acquire new words at slower rates (Cole and Flexer 2007; Paul 2009). Barker (2003) also explains that vocabulary and language deficiencies have deep effects not only in language-based activity but also in the interpersonal and private realms.

It is also well acknowledged in the literature that the hearing-impaired learners have to face major challenges in their bilingual education, and in particular in their acquisition of reading vocabulary (Hermans et al. 2008). Schmitt (2000) demonstrates clearly the central role of vocabulary knowledge in second language learning and teaching. He explains that vocabulary and lexical expressions can support a great deal of rudimentary communication; grammatical or any other type of linguistic knowledge cannot be used in communication without the mediation of vocabulary.

The chapter is structured as follows. "TCAD Educational Approach" describes the underlying educational philosophy of TCAD. "TCAD System Design" describes the tools, TCAD, TCAD+ and TCAD++. "Experimental Results" and "Discussion" discuss the results and the impact of the tools on the learners.

TCAD Educational Approach

Total Communication is an educational philosophy which promotes the use of the most effective communication methods (visual and auditory cues) to suit the needs of learners. It opens all avenues and modes of communication for hearing-impaired pupils. For example, teachers may employ visual imagery, gestures, lip reading, finger spelling sign and writing to maximise the language skills of hearing-impaired pupils. They may accompany their speech with vocabulary words/signs from American/British sign language, signing them in English word order, and then using invented signs to represent grammatical English morphemes (e.g. determiners, tense markers) and even signed English morphology. These are known as Manually Coded English (Lou 1988; Schick and Moeller 1992). Studies have demonstrated its benefit in all aspects, not only psychosocial but also linguistic and academic achievements (Vernon and Andrews 1990).

Total Communication is a term coined by Roy Holcomb in 1967 proposing that sign language can support the use of the oral method of communication and the use of any residual hearing to develop language skills. It can make the structure of spoken language visually accessible to hearing-impaired learners (Moores 2001). It is a response to the apparent reported low educational attainment, poor literacy and lack of mastery of English language by hearing-impaired students. It also acknowledges the failure in claiming that verbal communication, and in particular spoken communication, is the predominant medium of social exchange to support hearing-impaired learning (DES 1968; Babbidge 1965).

The past two decades have seen a major shift in learning theory from psychological views of learning as mechanistic and individualistic to emergent and social learning approaches (Greeno 1998; Salomon 1996), which are aimed at immersing learners in an environment that reflects the way they will be used in real life (Collins 1988). Situated learning is a general theory of knowledge acquisition and has been applied in the context of technology-based learning activities for schools focusing on problem-solving skills (McLellan 1995). Situated learning places the learner in an actual domain situation (authentic context) and interrelates with other people or the learning environment with real lesson which allows learning to happen. Lave (1988) argues that learning is best achieved when situated and embedded within activity, context and culture. Mondola and Doehler (2004) argue that learning a language means learning how to deal with contextualised and interactionally oriented discourse activities. Situated learning provides educators with a new perspective at engaging the learner in tasks and activities in a given social context. It has been successfully applied to mathematics education (Lave 1988), instructional design (Yusoff et al. 2010) and computer science education (Ben Ari 2004). It can also offer similar benefits in second language learning.

TCAD System Design

Blamey (2003) and Moores and Sweet (1990) explain that children with hearing impairment have smaller spoken-language vocabularies than hearing children. It is well established that world knowledge plays a central role in reading comprehension and therefore any software tool designed to improve pupils' reading comprehension must include a component for vocabulary building (Davis 1944).

The first phase of this research project addresses one of the major challenges for hearing-impaired children in their education: the improvement of the acquisition of reading vocabulary. The second phase is designed to extend the acquisition of single words into vocabulary groups based on the theory of situated learning. According to the main principles of situated learning, knowledge needs should be presented in an authentic context, i.e. settings and applications that would normally involve that knowledge. The third phase aims to provide a learning environment which requires social interaction and collaboration. That is accomplished by including games and stories based on the learnt vocabulary from the previous phases. These phases have been implemented through the conceptual tools, namely, TCAD, TCAD+ and TCAD++ described as follows. The system-architecture of TCAD consists of a database-management system, a server managed by an administrator and a user interface (Fig. 12.1). It is implemented using a variety of IT tools such as motion-capture to support the animation of realistic and effective sign language, an SQL database to build the dictionary and PHP and Java Script to build the user interface.

In the first phase of TCAD an e-dictionary is developed consisting of 300 English words. Each of these words is introduced to the learner through an animated

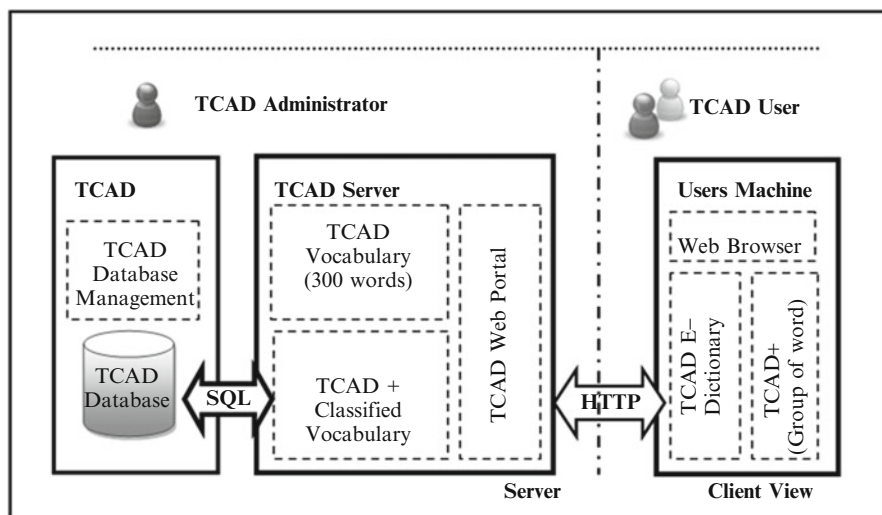


Fig. 12.1 TCAD system architecture



Fig. 12.2 E-dictionary screen shot for the lexical item “house”

character whose gender is selected by the learner. For each word the animated character reveals not only the corresponding Thai sign language and picture but also its International Phonetic Alphabet (IPA), the appropriate finger spelling and lip reading and a situated learning contextual illustration (Fig. 12.2). TCAD also provides the learner with word searching and browsing facilities and the teachers with administrator privileges so that they can enter new words and/or delete existing words.

TCAD+ presents the words in their authentic context to the learner. For example the lexical item “house” is associated with a screen that displays a picture of the



Fig. 12.3 Words in their authentic context

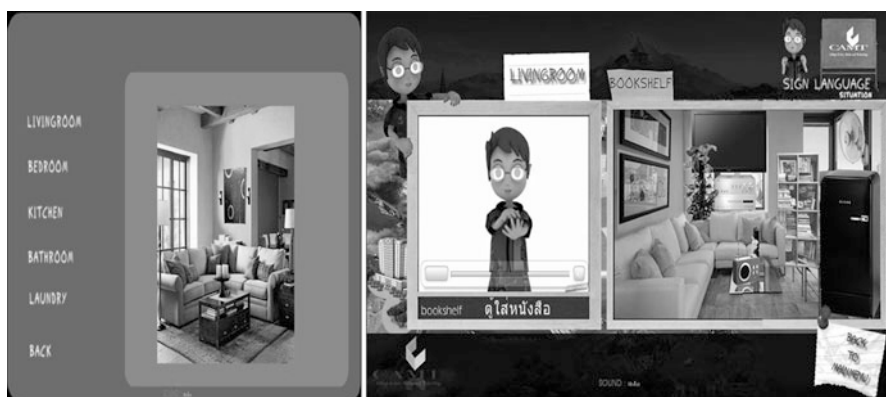


Fig. 12.4 Exploring the lexical item “house”

house with other buildings, inviting the learner to click on other pictures depicted in the screen (Fig. 12.3). When the learner clicks on any of these pictures TCAD+ reveals its identity by displaying the name in English, and if required the associated Thai sign language, its IPA, its appropriate finger spelling and lip reading.

The learner can also click on the word “house” to explore other lexical items associated with the house (Fig. 12.4).

Whilst TCAD and TCAD+ focus solely on the learner TCAD++ allows the educator to develop activities that integrate the rural local culture of pupils into the teaching of English vocabulary. Educators can create learning materials, such as stories and games in Thai and English, based on the words in the TCAD e-dictionary. These interactive stories and games are developed to engage the learner

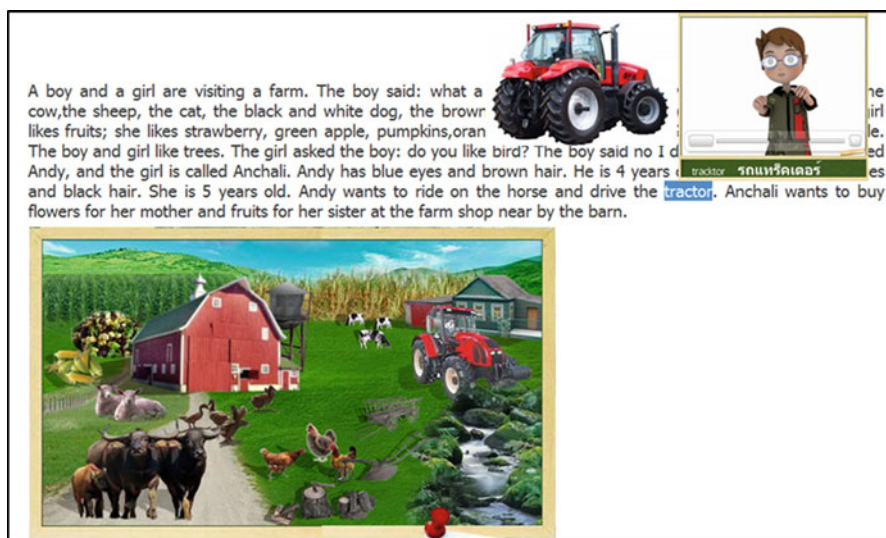


Fig. 12.5 TCAD++ vocabulary management component



Fig. 12.6 TCAD++ farm story

by utilising the acquired vocabulary and thereby help them become independent learners. For example, the pointing game, which is the simplest game, allows learners to point at any object in the story in response to questions asked by the testing management component (Figs. 12.5 and 12.6).

The testing component assesses the learning of colour concepts as well as of object positions. Its purpose is to evaluate the understanding of prepositions (Fig. 12.7). The farm game is a farming simulation game, inspired from the social

Visiting the farm

A boy and a girl are visiting a farm. The boy said: what a lovely sun! The boy likes animals. He likes the cow, the sheep, the black and white dog, the brown horse, the swan and the duck. The girl likes strawberry, green apple, pumpkins, and lemons. She does not like red apple. The girl like trees. The girl asked the boy: do you like bird? The boy said no I don't like it. The boy is called Andy, and the girl is called Anchali. Andy has blue eyes and brown hair. He is 4 years old. Anchali has brown eyes and black hair. She is 5 years old. Andy wants to ride on the horse and drive the tractor. Anchali wants to buy flowers for her mother and fruits for her sister at the farm shop near by the barn.

Updated : 29.05.2012 16:38:49

Student Name _____
Class room _____

| | |
|-----------------------|--------------------------------|
| ข้อที่ 1 | What colour is the strawberry? |
| <input type="radio"/> | pink |
| <input type="radio"/> | white |
| <input type="radio"/> | red |
| <input type="radio"/> | green |

| | |
|-----------------------|-----------------------|
| ข้อที่ 2 | Where is the tractor? |
| <input type="radio"/> | in the field |
| <input type="radio"/> | next to the cows |
| <input type="radio"/> | next to the dog |
| <input type="radio"/> | next to the sheep |

| | |
|-----------------------|--------------------|
| ข้อที่ 3 | Who has blue eyes? |
| <input type="radio"/> | Amnuay |
| <input type="radio"/> | Andy |
| <input type="radio"/> | Anchali |
| <input type="radio"/> | Angy |

Fig. 12.7 TCAD++ test sample



Fig. 12.8 TCAD++ farm scenario

network open game available from Facebook and Microsoft’s MSN Games (Zynga 2009). It involves learners in manipulating concepts and objects depicted in the farm and performing activities such as ploughing land, and buying and selling animals, vegetables and fruits (Fig. 12.8).

Experimental Results

A pilot study was carried out to assess the effectiveness of TCAD, TCAD+ and TCAD++ in optimising vocabulary acquisition and retention. To evaluate the impact of the first phase two teams of 18 hearing-impaired primary-pupils between 9 and 11 years old were selected from a hearing-impaired school based in Chiang Mai, Thailand. Team A used TCAD to learn new English vocabulary, while a tutor led team T using the traditional classroom approach based on sign language, flashcards and finger spelling. Both teams undertook a pretest consisting of a set of 20 random words extracted from the e-dictionary and were asked to match a picture with the correct word. Nine months later they were asked to repeat the test and these results were recorded as the long-term posttest. Figure 12.9 shows that team A outperformed team T significantly and even further improved its score after the 9-month period. Figure 12.10 highlights the statistical significance difference between the traditional approach where t is -1.739 and TCAD where t is $+0.512$ (Wicha et al. 2012).

The second pilot study involved 141 hearing-impaired pupils, aged 9–11 years, from the same Chiang Mai School; they spent 1 h each day over a period of 2 months to learn a set of 300 new words from the e-dictionary. These pupils were asked to take a test consisting of 36 pictures depicting words randomly selected from the stories; pupils were asked to choose the word that best matched a given picture. The posttest mean score (14.7) is again higher than the pretest mean (11.9) and the values of t and confidence interval are 6.489 and 1.645, respectively, with $df = 140$, demonstrating the effectiveness of improving vocabulary acquisition and retention using the TCAD+ approach (Tables 12.1 and 12.2).

Encouraged by the results and pupils' motivation to use TCAD software a further pilot study was carried out with 26 primary pupils aged between 9 and 11 years who

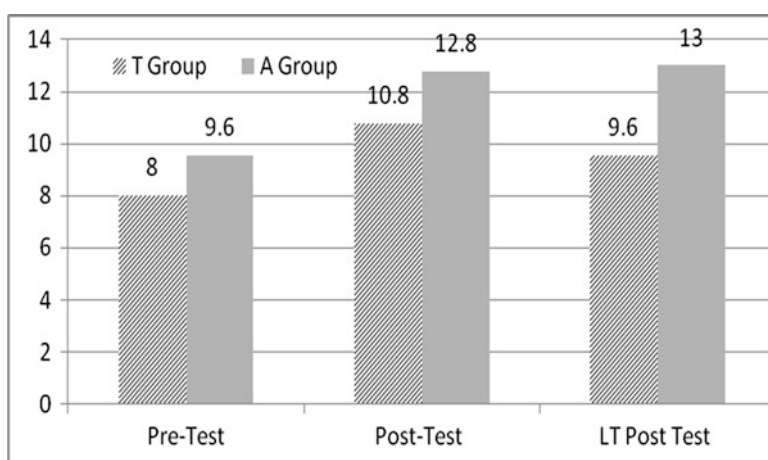


Fig. 12.9 TCAD memory test

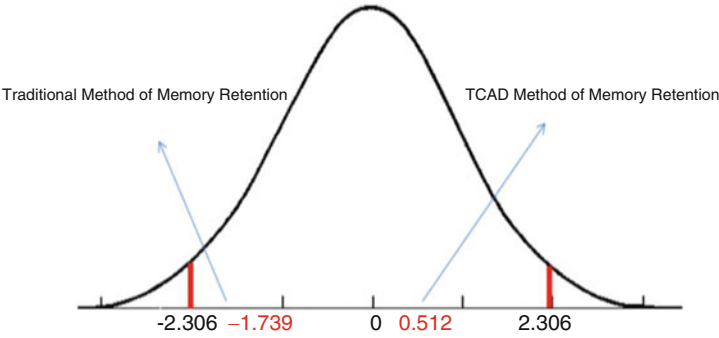


Fig. 12.10 Posttest and long-term test paired *t*-test results

Table 12.1 TCAD+ posttest and pretest results

| | | Mean | <i>N</i> | Std. deviation | Std. error mean |
|--------|----------|---------|----------|----------------|-----------------|
| Pair 1 | Posttest | 14.7447 | 141 | 4.85564 | 0.40892 |
| | Pretest | 11.9433 | 141 | 4.24057 | 0.35712 |

Table 12.2 TCAD+ *t*-test

| <i>t</i> value for pretest and posttest | df | Statistical significance at 95 % confidence interval |
|---|-----|--|
| 6.489 | 140 | 1.645 |

interacted with TCAD+ + 1 h a day for a period of 2 months. Pupils were then asked to complete a multiple-choice test consisting of 20 questions related to the farm story. The results of the pretest and posttest, shown in Fig. 12.11, indicate a positive and significant improvement in understanding and retention as also outlined in the *t*-test where the values of *t* and confidence interval are 7.994 and 1.708, respectively, with *df* = 25 (Table 12.3).

Discussion

The TCAD tool is an example of how the technology can provide a dynamic and effective instructional environment to support the needs of hearing-impaired pupils in developing their bilingual education programme. Currently, the TCAD system is limited to improving vocabulary acquisition and retention of English words for the topic of farming, which is a familiar context to this group of pupils.

It is well known that hearing loss can affect pupils’ ability to communicate, which may lead to difficulties developing relationships with other pupils and to being unable to fully participate in social activities (Barker 2003). Teachers at Chiang Mai primary school noted improved motivation and enthusiasm among their pupils for learning English because they found pupils interacting with the system during break

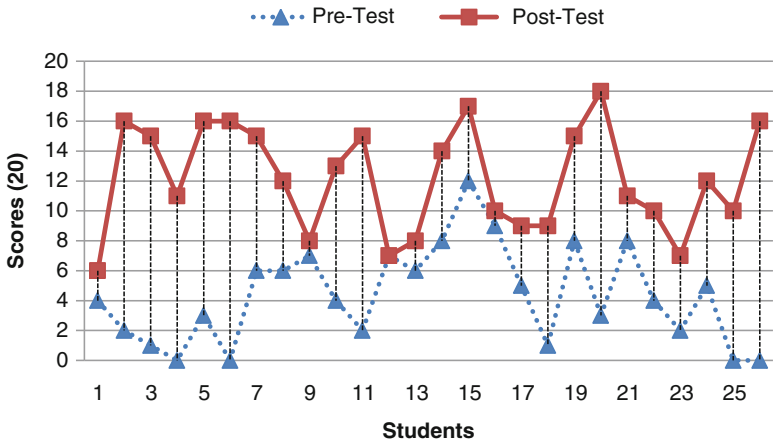


Fig. 12.11 TCAD++ pretest and posttest scores

Table 12.3 TCAD++ *t*-test

| <i>t</i> value for pretest and posttest | df | Statistical significance at 95 % confidence interval |
|---|----|--|
| 7.994 | 25 | 1.708 |

times and after school time. The TCAD tool has also provided a forum for social interaction as pupils played sometimes together with the farm games and other times competed against each other on who can build the biggest farm plantation. This study has encouraged hearing-impaired children to become independent learners and to also have fun in learning a new language.

Conclusions

This chapter describes the use of a system based on the concept of total communication philosophy to enhance the vocabulary acquisition and retention skills of hearing-impaired pupils. The total communication approach was shown to have allowed the structure of spoken language to be visually accessible to hearing-impaired learners. The TCAD system employed a variety of visual aids and animated features (e.g. lip reading, finger spelling sign and animation) to optimise the language skills of hearing-impaired pupils. It also reinforced the acquisition and retention of single words into vocabulary groups and stories and games through the utilisation of situated learning. These stories and games are set within the local context and culture of these primary school pupils. Though the developed system has focused on the teaching of English to Thai pupils, it can be easily adapted to any bilingual education programme.

The children made progress and enjoyed the animated e-dictionary platform and the learning of new vocabulary in situated learning environment. TCAD has also empowered the teachers with a tool to offer a rich and motivating environment evidenced by the pupils interacting with TCAD beyond school lessons. Future work will focus on extending TCAD functionalities to include blogs, visual media and social networking. To increase collaboration and communication among pupils further work involves the development of small group exercises which can be accessible by tablet PCs following the recent Thai Government initiative to provide primary school students with tablets to support their learning.

Acknowledgements The authors acknowledge the financial support of the Office of Higher Education Commission, Ministry of Education, Thailand, and the European Erasmus-Mundus Sustainable e-Tourism project 2010–2014.

References

- Babbidge H (1965) Education of the deaf. A report to the Secretary of Health, Education, and Welfare by his Advisory Committee on the education of the deaf. Ref. no. 0-765-119. Government Printing Office, Washington DC
- Barker LC (2003) Computer assisted vocabulary acquisition: the CSLU vocabulary tutor in oral deaf education. *J Deaf Stud Deaf Educ* 8(2):187–198
- Ben Ari M (2004) Situated learning in computer science education. *Comput Sci Educ* 14(2): 85–100
- Blamey PJ (2003) Development of spoken language by deaf children. In: Marschark M, Spencer P (eds) *Oxford handbook of deaf studies, language, and education*. Oxford University Press, pp 232–246
- Bosseler A, Massaro DW (2003) Development and evaluation of a computer-animated tutor for vocabulary and language learning in children with autism. *J Autism Dev Disord* 33(6): 653–672
- Cole EB, Flexer C (2007) *Children with hearing loss: developing listening and talking birth to six*. Plural Publishing, San Diego
- Collins A (1988) Cognitive apprenticeship and instructional technology. Technical report no 6899. BBN Labs Inc, Cambridge
- Davis FB (1944) Fundamental factors of comprehension in reading. *Psychometrika* 9:185–197
- Department for Education and Science (1968) *The education of deaf children: the possible place of finger spelling and signing*. HMSO, London
- Gersten R, Carnine D, Woodward J (1987) Direct instruction research: the third decade. *Remedial Spec Educ* 8(6):48–56
- Greeno JG (1998) The situativity of knowing, learning, and research. *Am Psychol* 53(1):5–26
- Hermans D, Knoors H, Ormel E, Verhoeven L (2008) Modeling reading vocabulary learning in deaf children in bilingual education programs. *J Deaf Stud Deaf Educ* 13(2):55–174
- Lave J (1988) *Cognition in practice: mind, mathematics, and culture in everyday life*. Cambridge University Press, Cambridge
- Lou MW (1988) The history of language use in the education of the deaf in the United States. In: Strong M (ed) *Language, learning and deafness*. Cambridge University Press, New York, pp 75–98
- McLellan H (1995) *Situated learning perspectives*. Educational Technology, Englewood Cliffs, NJ
- Mondala L, Doehler SP (2004) *Mod Lang J* 88(iv):501–518
- Moore DF (1978) *Educating the deaf: psychology, principles, and practice*. Houghton Mifflin, Boston

- Moore DF (2001) *Educating the deaf: psychology, principles, and practices*, 5th edn. Houghton-Mifflin, Boston
- Moore D, Sweet C (1990) Relationships of English grammar and communicative fluency to reading in deaf adolescents. *Exceptionality* 1:97–106
- Paul PV (2009) *Language and deafness*, 4th edn. Jones and Bartlett, Boston
- Salomon G (1996) Unorthodox thoughts on the nature and mission of contemporary educational psychology. *Educ Psychol Rev* 8(4):397–417
- Schick B, Moeller MP (1992) What is learnable in manually coded English sign systems? *Appl Psycholinguist* 3:313–340
- Schmitt N (2000) *Vocabulary in language teaching*. Cambridge University Press, Cambridge
- Vernon M, Andrews J (1990) *Psychology of deafness*. Longman, New York
- Wicha S, Sharp B, Sureephong P, Chakpitak N, Atkins AS (2012) An animated dictionary for hearing impaired students in Thailand. *Journal of Research in Special Educational Needs* 12 (4): 234–244
- Yusoff RCM, Zaman HB, Ahmad A (2010) Design a situated learning environment using mixed reality technology—a case study. *World Acad Sci Eng Technol* 71:887–892
- Zynga (2009) Zynga's FarmVille becomes largest and fastest growing social game ever. <https://zynga.com/>. Accessed 10 Oct 2009

Chapter 13

A Hybrid Approach to Automated Rating of Foreign Language Proficiency Using Oral Test Responses

Homayoon Beigi

Abstract This study was conducted to improve the automatic rating of oral test responses collected through Language Testing International's (LTI) Oral Proficiency Interviews using a Computer (OPIC). In OPIC tests, a computer automatically asks questions from the candidate and the responses of the candidate are recorded and consequently rated. This study has been performed on English OPIC tests. Although, no specific knowledge of the English language has been used for this phase of research and the results may be readily extended to tests in other languages. Preliminary results are quite promising, considering the utilization of the crude *Verbosity* feature.

Introduction

This study was conducted to improve the automatic rating of oral test responses collected through Language Testing International's (LTI) Oral Proficiency Interviews using a Computer (OPIC).¹ This research is a continuation of the preliminary work reported at the end of 2008 in a technical report (Beigi 2008) by Recognition Technologies, Inc. In an OPIC test, a computer automatically asks questions from the candidate and the responses of the candidate are recorded and consequently

¹These tests have been designed by the American Council on the Teaching of Foreign Languages (ACTFL) (ACTFL 2012).

H. Beigi (✉)

Recognition Technologies, Inc., 3616 Edgehill Road
Yorktown Heights, NY 10598, USA

Recognition Technologies, Inc., President Adjunct Professor in the Departments of Computer Science and Mechanical Engineering at Columbia University
e-mail: beigi@RecognitionTechnologies.com

rated. The results, reported here, stem from the application of the proposed rating technique on OPIc tests performed in English. However, no specific knowledge of the English language has been used in the algorithms presented for this phase of research. Therefore, the results may be readily extended to tests in other languages. Preliminary results are quite promising, considering the utilization of the crude *Verbosity* feature (Beigi 2008). *Verbosity* is a function of the quantity of the speech which has been uttered in response to a test question. It uses no information about the content of the response.

The main objective of this research is to best mimic the rating style of human raters using an automated process. One of the goals of the project is to be able to increase the granularity of the ratings. Specifically, it is desired to be able to break down the IM rating to three subcategories (IM Low, IM Mid, and IM High). The IM (Intermediate-Mid) rating encompasses a large portion of the population being tested. ACTFL defines these rating levels regularly in a guideline it provides its members (ACTFL 2012). To be able to attain this goal, the current rating style is learned by a statistical algorithm. Due to the continuous nature of the ratings returned by the statistical model, based on the a-posteriori probabilities returned by the model, one may increase the granularity of the rating to finer increments. This will produce further granularity which may not be reproduced by human raters in great ease!

Future phases of this research will be dealing with more substantial and qualitative features which utilize knowledge of the content of speech being uttered in response to the test questions (Beigi 2008). Some more discussion will be provided at the conclusion of this chapter.

OPIc

An OPIc test does not have any human tester associated with it. One may say that the computer is the tester. First, the candidate makes a self-assessment of his/her language proficiency. Then, a test is automatically created for the candidate by combining a collection of Novice, Intermediate, and Advanced prompts which are played back and the candidate is expected to respond to them. These responses are recorded and used to rate the candidate's proficiency. Different prompt categories require different lengths of response. The candidate is only allowed to produce a response which is limited in its length by some number of seconds, dictated by the test designers, corresponding to each prompt category.

Each OPIc exam consists of roughly 14 queries (prompts) that are picked from a large collection of stock questions. The questions are categorized into different levels of difficulty as well as the mental tasks that are required of the candidate in his/her response. Table 13.1 presents these categories as well as the maximum number of seconds allowed for the corresponding response and a quick remark about each category.

Table 13.1 Rating levels in an OPIc exam

| Prompt category | Level | Maximum duration of response (s) | Remark |
|-----------------|--------------|----------------------------------|---|
| nov1 | Novice | 30 | All Novice level questions |
| intrap | Intermediate | 90 | Role play |
| int2 | Intermediate | 60 | Describe an object or a place |
| int2n | Intermediate | 60 | A simpler version of int2 questions |
| int3 | Intermediate | 60 | Describe a process |
| int3q | Intermediate | 60 | Intermediate prompt to ask a question related to the intrp role-play |
| adv1 | Advanced | 120 | A past description |
| adv2 | Advanced | 120 | A past narration |
| adv3 | Advanced | 120 | Complication following the intrp role play |
| adv4 | Advanced | 120 | Description and narration following a story |
| adv5 | Advanced | 120 | Past description beyond the person (such as developments and changes) |
| adv6 | Advanced | 120 | Past description beyond the person (such as a current event) |

Depending on the self-proclaimed proficiency level of the candidate, a test is generated by combining a random set of questions, coming from the categories listed in Table 13.1. The number of questions from each category is dictated by the test level. For example, an Advanced test will have more advanced prompts in it, but it also includes some intermediate prompts. A Novice test will only have questions from the Novice and Intermediate categories. An Intermediate test will have more Intermediate questions (prompts) than an Advanced test would, but it will include less Advanced level prompts.

Audio Quality

The audio data was recorded using the μ -Law amplitude coding technique (G.711 1988) at a sampling rate of 8 kilo Hertz (kHz). The audio was then immediately converted to the High Efficiency-AAC Audio Format (**HE-AAC**) which is a very aggressive, lossy, and low-bit-rate audio compression technique (Beigi 2011). The compressed audio was uploaded to a server. In a limited number of tests, the audio was converted into MPEG-1 Audio Layer 3 (MP3) instead of HE-AAC. All audio responses, in turn, were converted back to Mu-Law 8-kHz audio and subsequently converted to a 16-bit linear Pulse Code Modulation (**PCM**) form which was used in the recognizer for obtaining the features described here.

Table 13.2 Rating levels in an OPIc exam

| Proficiency level | Acronym | Rating level |
|-------------------|---------|--------------|
| Novice Low | NL | 1 |
| Novice Mid | NM | 2 |
| Novice High | NM | 3 |
| Intermediate Low | IL | 4 |
| Intermediate Mid | IM | 5 |
| Intermediate High | IH | 6 |
| Advanced | A | 7 |

The Rating Process

OPIc tests are manually graded by human raters. There are 7 possible rating levels in the manual process. Table 13.2 shows the acronyms and numerical values used for the different rating levels. 1 corresponds to the least proficient group of speakers and 7 is associated with the highest level of proficiency.

Computed Features

Since the candidate responds to predefined prompts, his/her audio is not multiplexed with any other audio and is separately available for each response. Therefore, the *Verbosity* is computed by using the RecoMadeEasy® engine of Recognition Technologies, Inc. to extract segments where audio is present. The length in number of seconds of spoken audio constitutes *Verbosity*. To account for the length of each response (including pauses), the Verbosity feature is represented as a two dimensional vector which includes the length of spoken audio in seconds as the first dimension and the total length of the audio segment as the second dimension of the feature vector.

A rating process was trained and tested using the *Verbosity* feature. Let us assume that the feature for the t th response is denoted by $\mathbf{f}_t : 1 \mapsto \mathcal{R}^2$ and that the prompt associated with that response is denoted by l_i . Also, let r_k denote the k th rating level presented in Table 13.2. Theoretically, it is possible to describe any complex distribution by an infinite number of Gaussian distributions. However, in practice, this number may be made finite while obtaining a good approximation to the original complex distribution. If we assume that there exist a certain number of Gaussian Prototypes, the mixture of which describes the distribution of the features associated with responses to the i th prompt category, then the a-posteriori probability of the rating given a feature \mathbf{f}_t computed jointly for prompt l_i may be estimated by Eq. (13.1).

$$p(r_k|\mathbf{f}_t, l_i) = \sum_{j=1}^{N_i} p(r_k|g_j^i)P(g_j^i|\mathbf{f}_t, l_i) \quad (13.1)$$

where g_j^i is the j th Gaussian prototype for the i th prompt category and N_i is the number of Gaussian prototypes used to map the features associated with prompt level l_i . The a-posteriori probability of the Gaussian Cluster, g_i , given the feature \mathbf{f}_i is given by Eq. (13.2) (Duda Richard 1973).

$$P(g_j^i | \mathbf{f}_t, l_i) = \frac{p(\mathbf{f}_t | g_j^i) P(g_j^i)}{p(\mathbf{f}_t, l_i)} \quad (13.2)$$

where, $P(g_j^i) \forall j = 1, 2, \dots, N_i$ is the set of a-priori probabilities (priors) estimated by a clustering technique. $p(\mathbf{f}_t) \forall t = 1, 2, \dots$ are assumed to be 1 since at any instance, t , this represents the probability of occurrence of feature \mathbf{f}_t . Since the probability of occurrence of the feature vector, \mathbf{f}_t , has no bearing on the choice rating, r_k , this probability may be set to 1 at the time of the test. In other words, it is only important that,

$$P(g_j^i | \mathbf{f}_t, l_i) \propto p(\mathbf{f}_t | g_j^i) P(g_j^i) \quad (13.3)$$

$p(\mathbf{f}_t, l_i | g_j^i)$ is the joint likelihood of \mathbf{f}_t for prompt level l_i given Gaussian prototype g_j^i and may be computed using the equation for the *Normal* distribution,

$$p(\mathbf{f}_t, l_i | g_j^i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{f}_t - \mu_i)^T \Sigma_i^{-1} (\mathbf{f}_t - \mu_i) \right\} \quad (13.4)$$

$$\text{where } \begin{cases} \mathbf{f}_t, \mu_i \in \mathcal{R}^d \\ \Sigma_i : \mathcal{R}^d \mapsto \mathcal{R}^d \end{cases}$$

In (13.4), μ_i is the mean vector associated with the features of prompt level l_i , where,

$$\mu_i \triangleq \mathcal{E} \{ \mathbf{f}_t \} \triangleq \int_{-\infty}^{\infty} \mathbf{f}_t p(\mathbf{f}_t) d\mathbf{f}_t \quad (13.5)$$

The variance matrix of a multidimensional random variable is defined as,

$$\Sigma_i \triangleq \mathcal{E} \{ (\mathbf{f}_t - \mathcal{E} \{ \mathbf{f}_t \}) (\mathbf{f}_t - \mathcal{E} \{ \mathbf{f}_t \})^T \} \quad (13.6)$$

$$= \mathcal{E} \{ \mathbf{f}_t \mathbf{f}_t^T \} - \mu_i \mu_i^T \quad (13.7)$$

Training Process

In Eq. (13.1), the mixture coefficients, $p(r_k | g_j^i)$, are computed at the training stage using the *Expectation Maximization* algorithm and in conjunction with the training

data. Let us assume the total number of feature vectors associated with the rating, r_k , to be represented by T_{r_k} . Then, the joint likelihood of the feature vector, \mathbf{f}_t , associated with the training label, r_k , may be defined as,

$$\mathcal{L}_{t,k,j,i} = p(\mathbf{f}_t, r_k | g_j^i) \quad (13.8)$$

Therefore, the conditional probability of rating r_k with respect to the Gaussian prototype g_j^i , associated with prompt category, l_i is given by Eq. (13.9).

$$p(r_k | g_j^i) = \frac{\sum_{t,k,i} \mathcal{L}_{t,k,j,i}}{\sum_{j=1}^{N_i} \mathcal{L}_{t,k,j,i}} \quad (13.9)$$

The a-posteriori probability, $p(r_k | \mathbf{f}_t, l_i)$, is given by Eqs. (13.3) and (13.9), with the prior probabilities also computed at the training stage.

Consequently, the posterior likelihood for any rating given the selected feature (Verbosity) may be computed. These values will not generally add up to one and are considered to be likelihoods. A normalization is done to impose the summation of 1, rendering the computed values akin to a probability. Then, the rating with the highest likelihood is taken to be considered as the final rating for that response. An averaging or voting method may be used among the several responses in a test, to come up with the final rating for the test.

Results

Figure 13.1 shows the number of test responses used for each prompt category in obtaining the results. As it may be seen from the bar chart, there is a vast bias in the number of test responses available for each prompt category. This should somewhat affect the results. Figure 13.2 shows three graphs, summarizing the preliminary results obtained at the response level. Accuracy of the rating is reported separately for each prompt category. In total, 973,204 test responses were used in obtaining the rating results. Since the amount of data was limited, a round-robin approach for data conservation was used in selecting the training and test data sets. In this approach, the responses were split into 10 groups with balanced memberships from the different prompt categories. Then, 10 tests were designed, in each of which 9 segments were used for training and the disjoint remaining set was used for testing. A round-robin rotation allowed us to utilize the whole data for both training and testing while keeping the training and test data independent at all time.

The accuracy of reproducing the human ratings is shown to be slightly above 24% in average over all the 12 prompt levels of Table 13.1. Figure 13.2 also reveals two more statistics about the automatic rating performance. The first statistic is the graph labeled, “Accurate within 1.” This graph shows the accuracy of the automatic rating when compared with the human rating, if one allowed the results to be off by 1 level

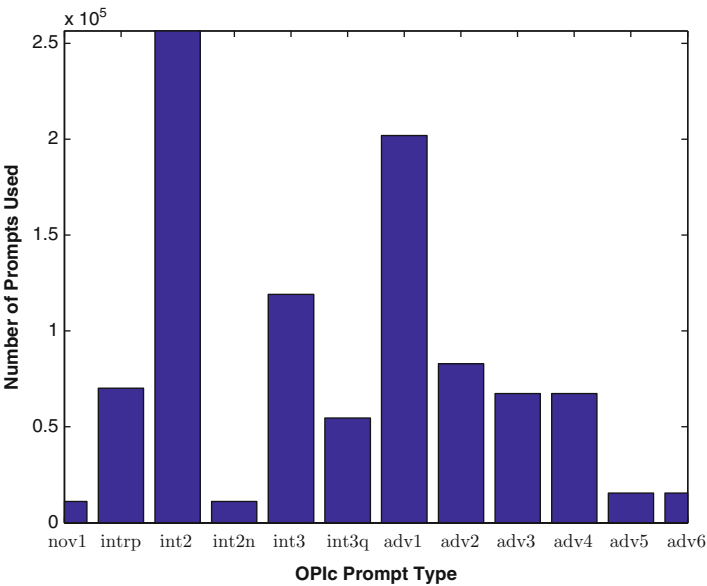


Fig. 13.1 Number of test samples used for the different prompt categories in obtaining accuracy results

within the possible 7 rating levels. On average, 66.33% of the ratings are correct when this looser requirement is adopted. For novice prompts this number is as high as 74.14% and it reaches its lowest performance for the most advanced prompts, hovering around 51%. The significance of this statistic becomes more clear when a test is rated both manually and through this algorithm. If the algorithm assigns a rating, higher by 1 level, compared to the human rating for candidate *A* and assigns a rating lower by 1 level, compared to the human rating given to candidate *B*, when the human ratings of *A* and *B* are the same, then, with about 66% confidence, candidate *A* is superior to candidate *B*. This may be extended to include 2 levels of relaxation, as reported in Fig. 13.2 with the graph labeled, “Accurate within 2”. In this case, the average confidence is about 91% for over all the different prompt levels. It can be as high as 98% for *int2n* level prompts and no less than about 84% for *int3q* prompt. Table 13.3 shows the detailed percentages associated with the results of Fig. 13.2.

Extension to the IM Delineation Analysis

In section “Results,” the methodology for reproducing a rating at the same granularity as that of human raters was reported. In reality, the ratings returned by the proposed system are produced in the form of a distribution. Using the provided

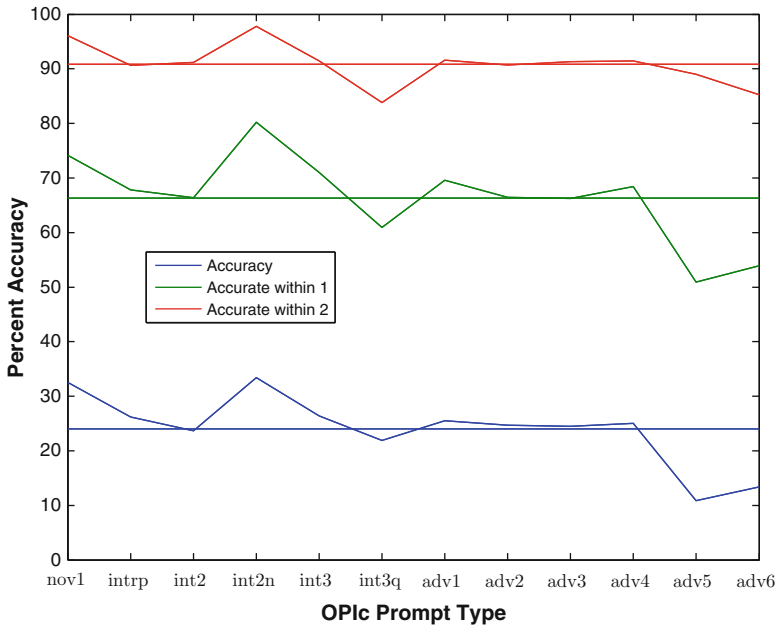


Fig. 13.2 OPIc test accuracy using the verbosity feature

Table 13.3 OPIc automatic rating performance for the different prompt categories

| Prompt category | Rating accuracy (%) | Accurate within 1 (%) | Accurate within 2 (%) |
|-----------------|---------------------|-----------------------|-----------------------|
| nov1 | 32.51 | 74.14 | 96.06 |
| intrp | 26.20 | 67.80 | 90.62 |
| int2 | 23.67 | 66.37 | 91.15 |
| int2n | 33.41 | 80.23 | 97.77 |
| int3 | 26.39 | 71.00 | 91.46 |
| int3q | 21.88 | 60.97 | 83.80 |
| adv1 | 25.48 | 69.59 | 91.59 |
| adv2 | 24.69 | 66.47 | 90.67 |
| adv3 | 24.47 | 66.24 | 91.34 |
| adv4 | 25.06 | 68.41 | 91.46 |
| adv5 | 10.87 | 50.89 | 88.97 |
| adv6 | 13.40 | 53.89 | 85.24 |

distribution, further rating resolution may be implemented. Figure 13.3 shows an example of the likelihood distribution returned by the automatic rating engine. As discussed in section “Results,” depending on the deviation of the automatic rating from the manual rating, a finer resolution may be attained. For example, in the rating level of interest, namely *IM* (rating 5), all the manual results are, by definition, rated *IM*. This means that if the automatic system rates the candidate higher, then the candidate may be considered an *IM High* (delineation level 3) and if a lower rating

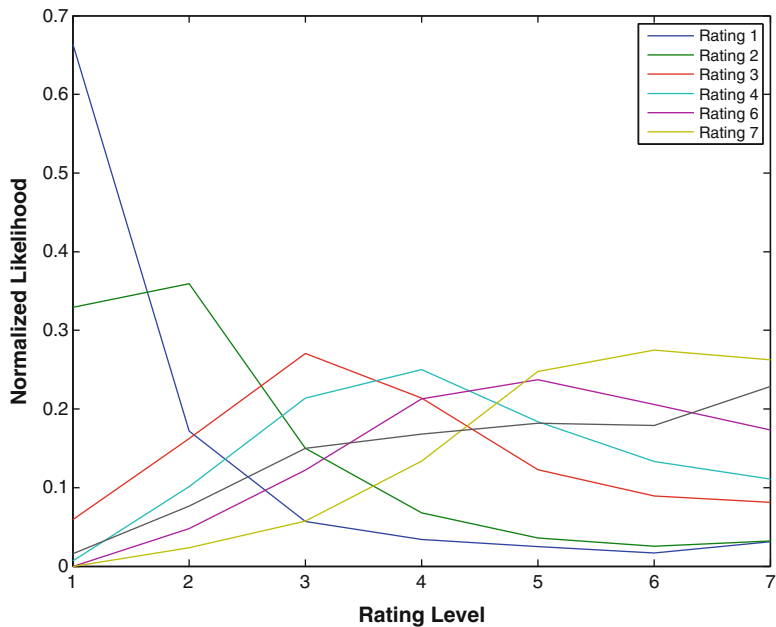


Fig. 13.3 Likelihood distribution for the rating levels of test samples associated with prompt category *int2* and ratings 1–7

is produced by the automatic system, then the candidate may be considered as an *IM Low* (delineation level 1). By looking at the probability levels returned by the automatic system, cutoff levels for delineation levels 1 and 3 may be obtained, thereby allowing for the determination of the range of level 2 delineation. The confidence in this rating is given by the last two columns of Table 13.3, depending on how much the automatic rating deviates from the manual rating. This translates to an average confidence of 66% if the discrepancy is at least one rating level and about 91% if the discrepancy is at least two levels.

The next section will be concerned with the development of techniques for combining the 14 or so responses for each test to produce an overall rating for the candidate. In the process, a methodology is presented for the evaluation of the delineations associated with *IM*-rated candidates. It utilizes the likelihood distributions depicted by Fig. 13.3 in association with the manual rating (*IM*) to evaluate the delineation for the candidate. For instance, if the distribution leans more toward the higher rating, then an *IM High* (delineation level 3) is recommended. A confidence level is also presented to back the choice of recommended delineation.

Table 13.4 Performance of the OPlc automatic combined (test-level) rating

| Criterion | Accuracy (%) |
|-------------------|--------------|
| $ r - r_H < 0.5$ | 53.21 |
| $ r - r_H < 1.0$ | 85.81 |
| $ r - r_H < 2.0$ | 98.60 |
| $ r - r_H < 3.0$ | 99.51 |

Combination to Achieve Test-Level Rating

In this section, we will combine the automatic ratings from different responses in a test, to come up with a final test-level rating, r . Each test is made up of about 14 responses to questions which are chosen from the different *Prompt Categories* listed in Table 13.3. In the previous sections, we discussed the methodology for computing the automatic rating of each response in the test, conditioned on the prompt category. The resulting rating is a likelihood distribution associated with the different possible ratings. To come up with a final rating for each test, the mean distribution is computed over all the individual prompt responses in the test. The resulting distribution is used to assign a rating to the test. Once the mean distribution is computed, the rating is the mean value of the mean distribution. This rating, r , is a real number and can vary from 1 (*novice*) to 7 (*advanced*). If an integer rating is desired, r would have to be rounded to the closest integer.

In order to evaluate the performance of the automatic rating, each test has been evaluated by a human rater as well. This rating is denoted by r_H . Table 13.4 shows the accuracy of the automatic rating, r , when compared with the human rating, r_H . The first row of the table shows the accuracy for a rounded integer version of the rating which means that $|r - r_H| < 0.5$. The mean accuracy seems to be over 53%, which is considerably higher than the mean accuracy of about 22% associated with each prompt response.

Much higher accuracies are achieved if one loosens the performance criterion. For example, in row 2 of Table 13.4, about 86% accuracy is achieved if one allows the difference between the automatic rating and the human rating to be within one rating level.

A total of 5, 138 tests were rated using the automated system discussed in this report. Figure 13.4 shows the histogram of the deviation of the automatic rating from human ratings. The mean deviation is $\mu = -0.056$ and the standard deviation is $\sigma = 0.74$. Note that a deviation of up to 0.5 is considered a correct rating when integer rating is used. This amounts to the 53.21% accuracy reported in row 1 of Table 13.4.

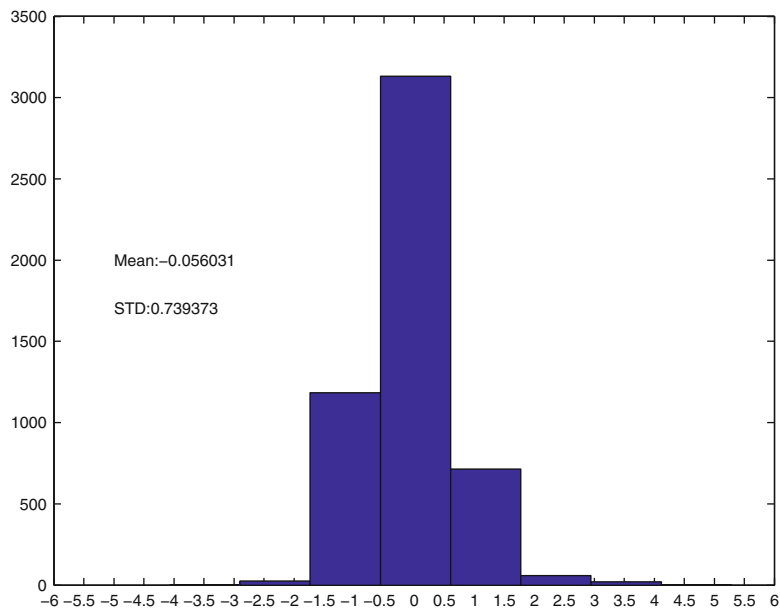


Fig. 13.4 Deviation statistics of the automatic ratings versus human rating

Conclusion

In the study covered in this report, it has been shown that we are able to duplicate the human rating of oral proficiency tests through a computer using an automated process with very crude features. The features, studied here, are only related to the amount of speech generated by candidates while responding to questions. Even using this crude information, over 53% accuracy is achieved in duplicating human rating results. In addition, if one allows for an error of less than one level within the 7 possible rating levels, over 86% accuracy is achieved. The distribution of the deviation from the human rating, reported in Fig. 13.4, shows a slight bias toward underrating ($\mu = -0.056$). This is also apparent from the slight skew of the distribution toward the negative deviation. The existence of most of the tests within the middle three bins of the distribution represents the sanity of this rating. Namely, it shows that there are not many surprises in the automatic rating results. This may also be observed by the fact that about 99% of all the tests fall within a deviation of up to 2 rating levels when compared to human rating.

One of the objectives of this study is to achieve better subrating of the Intermediate-Mid level tests. Since most candidates are categorized in this level, it is highly desirable to break up this category into smaller groups. In section “Extension to the IM Delineation Analysis,” it was noted that based on the combined results from the human rating and the automatic rating, a higher granularity may be achieved for further separation of the Intermediate-Mid level candidates into 3

subcategories. This is done by using the results from the automatic rating system to separate the candidates who have been human-rated into the Intermediate-Mid level. Since human raters are generally not able to rate candidates in a higher granularity than the 7 levels shown in Table 13.4, it is hard to assess the performance of the automatic subrating. However, based on the tight performance of the distribution (small standard deviation) shown in Fig. 13.4, it may be argued in a qualitative fashion that very reliable results may be obtained. Such combination techniques have been used in many mathematical applications such as fusion results shown in Viswanathan et al. (2000).

These results are quite promising, since the basic accuracy for each prompt response was shown to be below 23% in the preliminary results, but using a combination of the results from the roughly 14 prompt responses per test, the reported respectable accuracies of over 53% have been achieved. Based on the preliminary studies reported in Beigi (2008) and the results of this report, a considerably more accurate system is foreseen once linguistic information is utilized in the calculation of the automatic rating of the OPIc tests.

The author has been working on creating a language model based on ACTFL *Writing Proficiency Tests (WPT)* (ACTFL 2012). Also, work is in progress to produce the transcription of the discussed oral tests using a speaker-independent speech recognizer. The transcribed output will then be processed by the said language model, in order to produce ratings based on linguistic content. The results will then be combined with those discussed here, in order to produce a hybrid system for automated rating of foreign language proficiency. The language model is being developed based on WPTs to avoid further complications due to transcription errors. Most speech recognizers will not be able to produce better than 50% word-level accuracy on speech of nonnative speakers. Therefore, it is important to develop the language model independent of these transcription errors. Part of the practical problem at this moment is the number of WPTs which are available for developing this language model. In the absence of a large number of WPTs, massive amount of text from large literary corpora are being used to produce a target model. Although such target model would have to be modified to fit the statistics coming from nonnative language production. In addition, since written language is governed by somewhat different statistics compared to oral language, in most languages, the produced language model would have to be modified to reflect oral speech.

The methods discussed here are mostly language-independent, although relevant statistics would need to be generated for the language of interest to achieve greater accuracy. The path of the future work, discussed in the above paragraph, is far more language-dependent and requires a speech recognizer and a pertinent language model in the language of choice. The performance of the language model and rating by textual context is plagued by many practical uncertainties produced by the behavior of the speech recognizer and the inconsistencies among written and oral language as well as different language proficiency levels. In addition to statistical analysis on the textual context, it will definitely be necessary to devise methods for finding common errors among nonnative speakers of the language of choice.

Acknowledgements The author would like to thank the Center for Language Studies at the Brigham Young University, Language Testing International (LTI), and the American Council on the Teaching of Foreign Languages (ACTFL) for making this research possible.

References

- ACTFL (2012) American Council on the Teaching of Foreign Languages (ACTFL) guidelines
- Beigi H (2008) Whether computer analyses can predict human ratings of speaking proficiency. Recognition Technologies, Inc. Technical Report No.: RTI-20081205-01
- Beigi H (2011) Fundamentals of speaker recognition. Springer, New York. ISBN: 978-0-387-77591-3, <http://www.FundamentalsOfSpeakerRecognition.org>
- Duda Richard O, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York. ISBN: 0-471-22361-1
- G.711 (1988) Pulse Code Modulation (PCM) of voice frequencies. ITU-T recommendation. URL www.itu.int/rec/T-REC-G.711/e
- Viswanathan M, Beigi HS, Maali F (2000) Information access using speech, speaker and face recognition. In: IEEE international conference on multimedia and expo (ICME2000)

Chapter 14

Multilingual Systems, Translation Technology and Their Impact on the Translator's Profession

Carlos S.C. Teixeira

Abstract Starting from an overview of multilingual systems, I point out the usefulness of machine translation in some translation contexts, especially in dynamic environments. Then I describe how electronic tools can be useful for the human translator and how machine translation can be integrated into translation memory systems in the translation workflow. After this general panorama, I enumerate a series of studies in the field of translation studies that deal with translation process research and investigate the interaction between human translators and those technologies. I note that the main aspects being investigated are productivity, quality and effort. I also mention how the new technologies might affect the translation market and the activity of translation professionals. I conclude by indicating some areas for future research, including tool usability and job satisfaction.

Introduction

Multilingual Systems

For the purposes of the current chapter, a multilingual system is defined as any dynamic system that can support a variety of languages or locales.¹ This definition excludes static multilingual entities such as a bilingual poetry book or tourist

¹In the context of computer software, a “locale” can be defined as “a collection of standard settings, rules and data specific to a language and geographical region” (Esselink 2000, 471).

C.S.C. Teixeira (✉)
Intercultural Studies Group, Universitat Rovira i Virgili,
Avda. Catalunya 35, Tarragona, 43002 Spain
e-mail: carlostm@linguanativa.com.br

brochure. A multilingual system can be available in a variety of media, but it is most commonly found in electronic devices such as computers, cell phones, smartphones, game consoles or household appliances. The applications (programs) on these devices usually have components for language input, processing and output. On the input side, they can comprise a keyboard or a keypad (either physical or on-screen), a drawing board (for handwriting) or a microphone (for voice input). The language-processing component can involve speech recognition, handwriting recognition, transliteration, translation and speech synthesis. The output component typically comprises screens (for text display) or loudspeakers (for voice output).

Static Versus Dynamic Translations

Translation enters this as part of the processing component, which is the core mechanism that allows the system to understand/deliver messages in different languages. The translation element can again be of two kinds: static or dynamic.

The static translation element comprises “stored” language fragments that remain within the system waiting to be used according to the user’s preferences. This is the case of user-interface strings. For example, when you choose a different display language for a computer application in the Options menu, the texts to be displayed in the new language are already stored in the program. They do not need to be generated in real time, as they were created previously, by human translators, machine translation or a combination of both.

The dynamic translation element is present in real-time translation, which is where machine translation plays a key role.

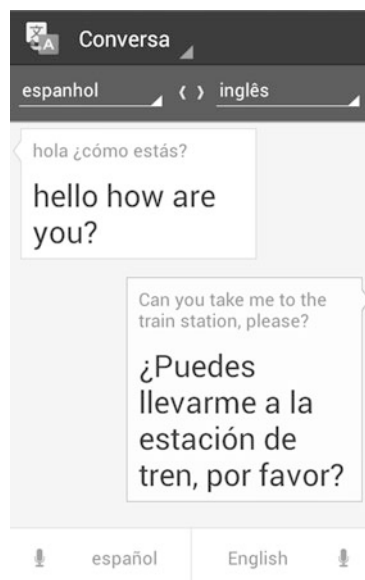
The Relevance of Machine Translation

Let us take automatically translated chats as an example: If you add a “bot”² as one of your contacts in Google Talk, you can get all your chat messages instantly translated. If you add one of your regular contacts to this “conversation”, you get an instantly translated chat, where the “bot” acts as your translator. In both cases, the resulting translated text is not previously stored within the application, but has to be generated somehow as you type.

In the realm of mobile devices, some smartphone applications can now offer “automated consecutive interpreting”. This allows the user to speak in one language and then hear the corresponding translation. The application also offers the original

²A “bot” is a virtual contact that will reply to each of your chat messages by translating it into a chosen language. For more details, see <http://support.google.com/talk/bin/answer.py?hl=en-US&answer=172257&topic=1190&ctx=topic>.

Fig. 14.1 Conversation mode of the Google Translate app for mobile platforms



text in written form, so users can make sure that what they said was understood correctly (Fig. 14.1). Internally, what happens is a sequence of speech recognition, transcription, machine translation and speech synthesis.

Although the results are not always accurate, it is easy to imagine situations where such a tool could be useful: when one is in a country where people speak a language one does not understand, or in any low-risk situation where an interpreter is not available.

Other examples of multilingual applications supported by machine translation include automatic subtitling (as available on YouTube for English-language videos), real-time translation of Web sites (such as in the standard configuration of the Chrome browser) and the option to translate posts and comments in Facebook (through Microsoft Bing).

Recent Developments in Machine Translation

As can be seen from the above examples, machine translation is thus the only option for providing translated content to the user in the context of dynamic multilingual systems. This explains the increasing demand for MT integration in several applications and for quality improvements in its underlying technology, which is being matched by intense research and development efforts in both industry and academia.

Several IT giants such as Google,³ Microsoft⁴ and IBM⁵ have large-scale research projects on natural language processing and machine translation. In another front, the open-source initiative Moses,⁶ a machine-translation engine builder, has served as the basis for several commercial and non-commercial projects, such as LetsMT,⁷ EuroMatrixPlus⁸ and Stilven.⁹ Other initiatives include PLoTO¹⁰ and Apertium,¹¹ to name a few.¹²

In addition to providing real-time translation in dynamic multilingual systems, machine translation also seems to be the strongest trend for addressing the volume explosion of (static) translatable content (Romaine and Richardson 2009; European Commission 2009), considering the potential increase of translation output through the use of MT (Plitt and Masselot 2010).

Current Trends

Technology developments such as faster Internet connections and cloud computing have fostered the emergence of collaborative scenarios¹³ that were non-existent a decade ago. In addition to this, the integration of machine translation into collaborative platforms has made it possible for people that are not proficient in a foreign language and are not traditionally trained to work as translators to start translating—often on a voluntary basis. This is typically the case with video subtitling on the Internet, also known as fansubbing.

A similar trend, called crowdsourcing, community translation or volunteer translation,¹⁴ relies on users to localize Web sites, online forums and the like, as has been the case with Facebook and Twitter Web pages or with TED subtitles.

Testimonials from industry also corroborate the potential of combining machine translation with collaborative platforms for crowdsourcing translation, both on a commercial or a volunteer basis (see Cavalitto 2012).

³<http://research.google.com/about.html>.

⁴<http://research.microsoft.com/apps/dp/areas.aspx?a=47189>.

⁵<http://www.research.ibm.com/compsci/spotlight/nlp/>.

⁶<http://www.statmt.org/monos>.

⁷<http://www.letsmt.eu>.

⁸<http://www.euromatrixplus.net>.

⁹<http://project.cgm.unive.it/stilven.en.html>.

¹⁰<http://www.pluto-patenttranslation.eu>.

¹¹<http://www.apertium.org>.

¹²While MT research in the United States tends to be done within (big) companies, in Europe it usually takes place in the form of industry–academia consortia, most often as government-funded research.

¹³On collaborative translation, see O'Brien (2011).

¹⁴See Pym (2011) for a discussion about these terms.

Potential Impacts of Technology on the Translator Profession

A question that has been around for a long time and is still the subject of dispute in several translators' forums is whether MT is a potential *replacement* for human translation in "traditional" scenarios. A survey conducted by Piróth (2011) with 160 translators from the IAPTI, ATA and ProZ.com actually reflects the mixed feelings regarding the use of MT among translators (see also Hartmann 2010).

Since translators are not available in all situations and there are alternative ways of having language translated in its various forms (static text, dynamic texts, images, speech, etc.), what will happen with the translator proper? What changes should be expected by a professional translator (i.e. someone that makes their living mainly from their translation work)? In other words, what's in it for translators?

What is certain is that language tools in general, and machine translation in particular, can be powerful aids not only for the general user and for the translation buyers but also for the language professionals, including translators.

Language Tools and Translation Memory Systems

Language Tools

Any applications that support the understanding, processing or rendering of linguistic material might be called *language tools*, ranging from text editors to dictionaries, from optical character recognition (OCR) to speech transcription software.

Although all those tools can also benefit translators directly or indirectly, there is a subset of language tools that are especially targeted at translators, generally included under the umbrella name of computer-aided translation (CAT)¹⁵ tools. These include translation memory systems and glossary managers, but also document aligners, term extractors and several kinds of validators and checkers.¹⁶

Translation Memory Systems

Translation memory (TM) systems have been in use for about 20 years as a means of increasing productivity and consistency over time. Basically, a TM comprises a database of source segments (sentences) paired with their corresponding target

¹⁵Pym (2011: 77) argues against the use of this term. Others prefer Translation Environment Tool (TEt), a term coined by Jost Zetzsche.

¹⁶For a comprehensive, albeit somehow outdated, categorization of electronic translation tools, see Austermühl (2001: 8–17).

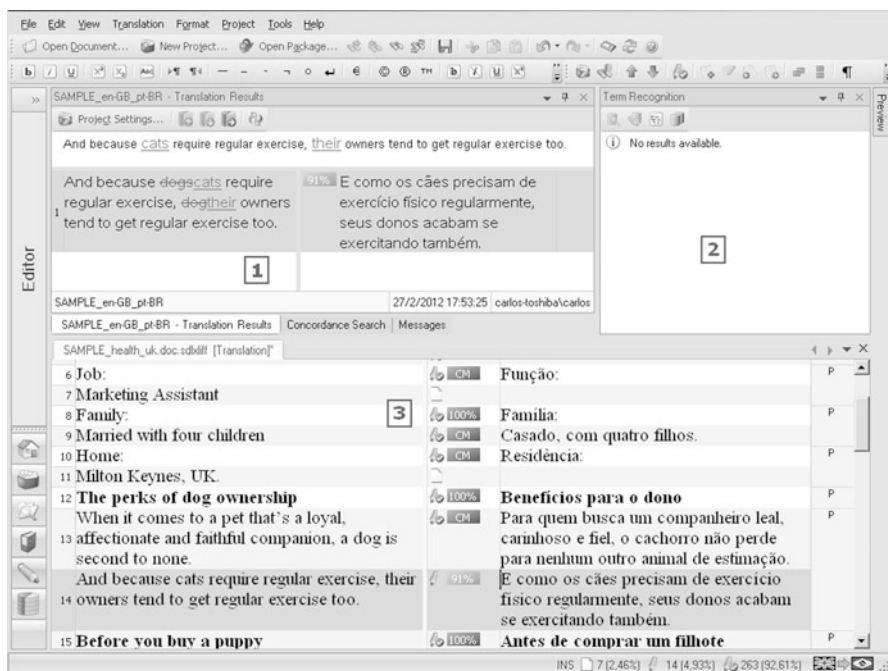


Fig. 14.2 Typical interface of a translation memory system

segments, created by either manual translation or document alignment. After several translation rounds, the volume of this database gets large enough to allow for the leverage of repeated segments or similar segments.

These tools offer translators in real time the suggested translations for a particular segment. An *exact match* happens when the segment being translated is identical to a source segment in the database. When the segment being translated is similar to a source segment in the database, but not identical, there is a *fuzzy match*. There may be one or several matches of each kind. In order to help translators make any necessary edits in the suggested translation, for each segment being translated the tool displays the level of similarity (as a percentage) between the source text in the current segment and the source text in the database. Additionally, for each suggested translation, it highlights the individual differences (additions, deletions, modifications) (Fig. 14.2). In sum, for each segment we may thus have exact matches, fuzzy matches within a reasonable range (usually 75–99%) and no matches.¹⁷

¹⁷For a more detailed definition of TM databases; an explanation of how analyse, concordance and matching functions work; potential of productivity and quality gains, etc., see Webb (1998).

The Convergence of TM and MT

In traditional TM systems, when no suitable match was found in the memory translators had to translate from scratch or to type on top of the source text (to take advantage of unchangeable elements between the source and the target text, such as code, tags or proper names). No use was made of machine translation and in fact, until recently TM and MT were seen as opposing approaches to using technology in translation.¹⁸

With the quality improvements in statistical and hybrid machine translation in the last 5–10 years, mainstream TM systems started to incorporate MT as a way to populate target segments when no translation match is available in the database. Translators then have an additional source of suggestions (MT engines) besides those coming from the translation memory.

In a parallel development, translation done on top of raw machine-translation output—generally referred to as post-editing (Krings 2001; Allen 2003, 2005; Pym 2011), had already been in use for some time, but without resorting to translation memories. Now some post-editing environments, as is the case with the Google Translator Toolkit (Fig. 14.3), also offer an option to include translation memories as a reference, making MT-assisted TM and TM-assisted MT come closer together.



Fig. 14.3 Example of environment for post-editing machine translation

¹⁸The last statement is especially true among translation professionals. Historically, translation memory systems are an offspring of machine translation developments in the 1980s.

Research in the Field

In this scenario of integration between TM and MT, new questions are being asked by translators, translation agencies, translation clients and tool suppliers, including the following: What is the increase in productivity from the resulting integration? How is quality affected? What changes happen in the tasks performed by language professionals?

Translation Process Research

Historically, the focus of study of translation research had moved from the source text to the target text, in a myriad of product-centred approaches. Recent years have seen the rise of process-centred approaches, i.e. studies that focus on what happens *while* translation takes place, i.e. *while* translators translate, in an attempt to address the new questions.

Crucial for these studies are the strategies and technologies that allow researchers to collect real-time data from the actual translation process, which today typically takes place on a computer. Alongside more traditional data-gathering methods such as think-aloud protocols (TAPs) (Ericsson and Simon 1998; Jakobsen 2003), other methods such as keystroke logging (Jakobsen 2002, 2006), eye-tracking (O'Brien 2006, 2009) and screen recording allow us to identify where attention is being placed and even to measure cognitive load, e.g. through pupil dilation (see O'Brien 2006; Shreve and Angelone 2010). Several studies also combine those methods (see Alves 2003; Dimitrova 2005; Carl et al. 2011) for better confirmation of results.

Translation Technology Research

Machine translation has traditionally been a subject of study mainly within the computational linguistics community, possibly due to the technicalities involved in its conception. Translation scholars are more interested in knowing what those systems can offer than in the meanders of the underlying technology. Translation memory systems, on the other hand, for their immediate applicability and simpler conception, have become a frequent subject within translation circles.

Tool manufacturers and the translation industry in general have published extensively on different types of tools, while academia has tried to keep up with the developments in a more conceptual way (cf. Pym 2012a for the different kinds of contributions that can be expected from either community).

MT Research

Although the idea of automatic translation can be traced back to as early as the seventeenth century, machine translation in its more contemporary form dates back to the first half of the twentieth century and has undergone several stages of development, with the current trend being towards data-driven statistical machine translation (SMT) and hybrid approaches (combining SMT with rule-based MT).¹⁹

As stated before, MT in itself was not a topic in translation research, but revising and post-editing have deserved serious attention, e.g. in Krings (2001), Guerra Martínez (2003) and Mossop (2001, 2007). Several studies have tried to determine whether there is an actual increase in speed while post-editing machine-translated segments when compared to translating from scratch.

Allen (2003, 2005) conducted several studies on machine translation post-editing with specific tools and provides some guidelines for improving its results. Lee and Liao (2011: 142) “suggest various benefits for the use of MT, such as facilitating source text comprehension and reducing translation errors.” Nevertheless, the authors’ productivity assumptions that MT can save much time “from needing to type out words” (Lee and Liao 2011: 141), although plausible, deserve further empirical testing. A recent study with seven participants working in the English-to-Danish language combination (Carl et al. 2011) actually points in the opposite direction. García (2010) compares time and quality between translating “entirely from the source text” versus “editing machine translation” and also finds that “time differences were not significant”, although “the machine translation seeded passages were more favourably assessed” (2010: 7).

TM Research

Much more recent than machine translation, CAT tools appeared first in the form of terminology management systems in the mid-1980s and then as translation memory (TM) systems in the early 1990s.

The first academic studies to deal with TM systems appeared before the turn of the millennium.²⁰ Later on, Dragsted (2004) investigated how the forced (usually sentence based) segmentation in TM systems reflects and interferes with cognitive segmentation. Her study raises an important debate on an aspect of translation tools, also applicable to MT, that is usually taken for granted but can have an impact on all the metrics relevant for the field (including quality).

¹⁹For an introduction to machine translation and its history, see Hutchins and Somers (1992). For a more recent overview of MT developments, see Way (2009).

²⁰Webb (1998) presents a comprehensive study of TM systems and summarises the state of affairs at the end of the 1990s, with a prophetic foresight for the following decade. For a very recent history and overview, with future prospects, see Zetzsche (2012).

More recently, Christensen and Schjoldager (2010) provide a good general overview of research on translation memories and suggest some paths for future research, namely, “on how translators interact with TM technology and on how it influences translators’ cognitive processes” (2010: 99). Yamada (2011a) investigates how the type of content (“free translation” vs. “literal translation”) in a translation memory affects translation speed, and concludes that literal translations are more advantageous for higher fuzzy-match categories. Christensen (2011) focuses on studies that deal with “mental processes” in the interaction between translators and TM tools.

TM/MT Research

As far as the integration between TM and MT is concerned, O’Brien (2006) compares the performance of TM vs. MT when translators work in a “traditional” TM system that integrates MT suggestions. One of her findings is that “cognitive load [and processing speed] for machine translation matches is close to fuzzy matches of between 80 and 90% value” (2006: 185). For fuzzy matches above 90%, including exact matches, TM processing is faster and requires a lighter cognitive load, whereas the opposite happens for fuzzy matches below 80%.

Guerberof (2009)²¹ analyses time and quality when editing TM suggestions vs. MT suggestions in a post-editing environment (a TM was used during the pre-translation phase, reproducing an actual scenario that can be found in industry). Her “findings suggest that translators have higher productivity and quality when using machine-translated output than when processing fuzzy matches [at any percentage level] from translation memories” (2009: 11).

Key Distinctions

Only a couple of the published studies that we are aware of seem to take into account a particular aspect that distinguishes translation memory systems from post-editing environments: TM systems show translators the “provenance” and the “relevance” of the translation suggestions coming from the memory, whereas most environments for post-editing MT display the “best translation suggestion possible” without any indication of its origin or degree of confidence. The presence or the absence of provenance information might influence translators’ performance and the failure to make this distinction might explain the discrepant results of some studies that compare translation times when (post-)editing MT and TM suggestions (such as O’Brien 2006 vs. Guerberof 2009).

By comparing MT-assisted translating versus TM-assisted post-editing in the same tool, Teixeira (2011) corroborates the “provenance” assumption and indicates

²¹ See also Guerberof (2012) for a more recent version of this study.

that the information displayed about each suggested translation (whether it comes from MT, TM, at which match percentage) has an impact on performance. Although quality was not affected, the amount of editing (an indirect measure of effort) was higher when provenance information was not available. Furthermore, knowledge of provenance seems to affect speed diversely depending on the type of translation suggestion.

Another study with a broader scope goes into the details of which pieces of information are more relevant for the translator (Morado Vázquez and Torres del Rey 2011). The initial experiment reported on in this study indicates that “some metadata elements are more often taken into consideration [by translators] than others” and that “the usability of the metadata provided has a lot to do with the way the tool presents it or works on it” (emphasis added).

An additional distinction that fails to be made in several studies is whether translators are given the option of a second rendering (self-revising phase), as this can drastically affect the final quality of the product. Just in the same way as professional translators may tend to spend more time revising than novice translators do (Jakobsen 2002; Dimitrova 2005),²² it is plausible to assume that translators can work differently during the drafting and revising phases when post-editing raw MT than when editing TM matches.

All in all, the benefits of TM or MT seem to depend hugely on several factors, including the text type and size/quality of the memory. Many of the productivity gains reported are directly due to these variables, and not due to the TM or the MT system itself. This is why it is extremely important to consider all variables when analyzing and comparing the results of different studies.

The Main Variables

In the studies mentioned above, there are mainly three variables being investigated: time (speed/productivity), effort and quality, and the latter continues to pose the most difficult problems to translation research. Several studies resort to human reviewers to assess translation quality based on predefined criteria (e.g. the LISA grid) and rely on individual subjectivity. Others use metrics originally intended for raw machine translation evaluation (BLEU, NIST, etc.).

While still relying on subjective criteria, Huang (2011) stresses the need to shift from the concepts of “good” and “bad” translation to a distinction between “acceptable” and “unacceptable”, and suggests a statistical approach. An alternative method that has also been used as an industry-relevant measure of quality is revision time, i.e. the longer it takes a professional reviewer to edit a translation up to delivery

²²Quite a few studies now suggest that with TM/MT translators adopt a “revise-as-you-go” approach (see Yamada 2011b; Martín-Mor 2011).

standards, the lower the quality of the translation. Although revision time provides a concrete measurement (words revised per hour), it still relies on the reviewers' subjectivity and specific usability or acceptability expectations.

Fields for Future Research

Quality

The importance of translation quality assessment is justified by the need to have a standard of comparison across studies: if we want to compare different translation environments or setups, we need to have a palpable measurement of the output quality. If we consider that most studies that investigate translation processes have to deal with only a limited number of subjects, language combinations, machine translation engines, (post-)editing environments, etc., the need for replication and comparisons becomes even more evident in order to allow for more general conclusions.

MT-Related Questions

Tool developers might also be interested in making their tools more productive. One potential improvement, which goes hand in hand with the evolution of MT algorithms themselves, would be to have translation tools display the *degree of confidence* of a given MT suggestion, in the same way current TM systems display the degree of fuzziness of a given TM suggestion.²³

However, several questions related to machine translation remain to be answered: How do translators perceive MT when actually working with it? How is MT being integrated in the translation workflow in the real world (by freelance translators, by LSPs, by clients)? How can MT be made more useful to translators (hypotheses: degrees of confidence, configuration options in the tools)? How can MT interact with existing translation memories for specific projects? These questions call for a greater interaction between the MT research and development community, translation tool manufacturers, translators, translation companies and translation researchers.

²³This brings about some interesting topics for discussion: Would there be a real difference between editing TM suggestions and MT suggestions in such an environment? To what extent would it make sense to differentiate between human-assisted machine translation (HAMT) and machine-assisted human translation (MAHT)?

Sociological Aspects

In addition to topics such as productivity, effort and quality that have dominated the scene so far, research in the field needs to also address some of the social aspects of technology and the expected changes in the profession—more translations for free, by non-professionals, source-language knowledge becoming less important.

One can envisage some necessary changes for the translation professional, such as the need to keep up with the latest tools and to learn post-editing techniques. Translators will probably also need to work in collaboration with area experts, final users and clients, who will also be able to edit the “moving text”²⁴ throughout the translation workflow.²⁵

Job Satisfaction

A related human aspect that needs to be addressed is how translation technology affects job satisfaction. Although it is believed that the integration of MT and TM may improve job satisfaction for translation professionals, the few existing studies on the topic²⁶ are generally inconclusive. Future studies will need to look into potential effects of the integration of MT and TM on job satisfaction, considering factors such as process workflows, tool interfaces, translation modes and available metadata to translators.

I believe that job satisfaction might involve at least two components: (a) pleasure in performing the translation activity, and (b) receiving fair payment for this activity. Of course, “pleasure” and “fair” are perceived on a personal basis, but one could assume that performing repetitive tasks or tasks for which one is not well prepared, well suited or simply does not like can hardly be pleasant. Likewise, decreasing wages (calculated as \$/h, not \$/text) cannot be considered fair remuneration. It is a question of who reaps the benefits of increased productivity. If these assumptions are correct, one could argue that (a) translation tools should ideally help eliminate repetitive tasks, and (b) the resulting payment schemes should try to compensate for the more intellectual nature of the remaining tasks.

²⁴Here I borrow the title of Anthony Pym’s (2004) book.

²⁵For a further discussion on this topic, see Pym (2012a, b).

²⁶For example, Wallis (2006) compares job satisfaction in two different translation modes (pre-translation vs. interactive). Her study involves only TM, but a similar comparison could be done to include MT.

Ergonomics and Usability

Ergonomics and usability are additional relevant aspects of the human–technology interaction that deserve more attention. Drawing on a survey among 874 translation professionals from 54 countries, Lagoudaki suggests, “many of the existing commercial TM systems are technology-driven applications (e.g. with an abundance of useless features and a complex, impractical and difficult to learn user interface), rather than user-driven applications” (2006: 4). Although several years have passed since her survey, this seems to still be the case.

Some aspects of the translator’s working interface have begun to call researchers’ attention (see Teixeira 2011; Morado Vázquez and Torres del Rey 2011, for the relevance of information displayed on screen to translators), but in this era of touch screens and voice commands, how could translators’ tools be made more intuitive and ergonomic, hence more productive? Is dictation software about to experience the same boom in development as machine translation has in the last decade?

Training

Training may be understood in two different ways: MT-engine training (tuning) and translator training. These are both exciting fields of research, as some studies indicate that the way forward to increase productivity and quality (and consequently to reduce effort and improve job satisfaction?) is to train MT systems (Plitt and Masselot 2010) and to train translators to use MT output (Pym 2009, 2012b; García 2010).

Acknowledgements I would like to thank Anthony Pym, Amy Neustein, David Orrego-Carmona and Esmaeil Haddadian Moghaddam for revising earlier versions of this manuscript. I would also like to acknowledge the funding to my doctoral research, provided through the European Commission’s TIME Marie Curie fellowship (FP7-PEOPLE-2010-ITN-263954).

References

- Allen J (2003) Post-editing. In: Somers H (ed) *Computers and translation: a translator’s guide*. John Benjamins, Amsterdam and Philadelphia, pp 297–317
- Allen J (2005) What is post-editing? *Translation Automation Newsletter* 4:3–5
- Alves F (ed) (2003) *Triangulating translation: perspectives in process oriented research*. John Benjamins, Amsterdam and Philadelphia
- Austermühl F (2001) *Electronic tools for translators*. St Jerome, Manchester
- Carl M, Dragsted B, Elming J, Hardt D, Jakobsen AL (2011) The process of post-editing: a pilot study. In: *Proceedings of the 8th international NLPSC workshop—special theme: human-machine interaction in translation*. Copenhagen studies in language 41. Samfundslitteratur, Copenhagen, pp 131–142. <http://www.mt-archive.info/NLPSC-2011-Carl-1.pdf>
- Cavalitto E (2012) The translation center behind translators without borders. *MultiLingual*: 31–34

- Christensen TP (2011) Studies on the mental processes in translation memory-assisted translation—the state of the art. *trans-kom* 4(2):137–160
- Christensen TP, Schjoldager A (2010) Translation-memory (TM) research: what do we know and how do we know it? *Hermes: J Lang Commun Stud*: 89–101
- Dimitrova BE (2005) Expertise and explication in the translation process. John Benjamins, Amsterdam
- Dragsted B (2004) Segmentation in translation and translation memory systems: an empirical investigation of cognitive segmentation and effects of integrating a TM system into the translation process. Doctoral thesis, Copenhagen Business School, Samfundslitteratur
- Eriksen KA, Simon HA (1998) How to study thinking in everyday life: contrasting think-aloud protocols with descriptions and explanations of thinking. *Mind Cult Activ* 5(3):178–186
- Esselink B (2000) A practical guide to localization. John Benjamins, Amsterdam and Philadelphia
- European Commission, Directorate-General for Translation (2009) Study on the size of the language industry in the EU. http://bookshop.europa.eu/is-bin/INTERSHOP.enfinity/WFS/EU-Bookshop-Site/en_GB/-/EUR/ViewPublication-Start?PublicationKey=HC8009985
- García I (2010) Is machine translation ready yet? *Target* 22(1):7–21
- Guerberof A (2009) Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus* 7(1):11–21
- Guerberof A (2012) Productivity and quality in MT post-editing. <http://thebigwave.it/birdeye/productivity-and-quality-in-mt-post-editing-by-ana-guerberof>
- Guerra Martínez L (2003) Human translation versus machine translation and full post-editing of raw machine translation output. Master's dissertation, Dublin City University, Dublin
- Hartmann N (2010) Real voices: what translators do and why we need to keep doing it. AMTA conference presentation. http://amta2010.amtaweb.org/AMTA/papers/1-01-Hartmann_paper.pdf
- Huang HJ (2011) Intermediality and human vs. machine translation. *CLCWeb: comparative literature and culture* 13:3. In: Tötösy de Zepetnek S, Asunción López-Varela Azcárate, Haun Saussy, Jan Mieszkowski (eds) Thematic issue new perspectives on material culture and intermedial practice. <http://docs.lib.purdue.edu/clcweb/vol13/iss3/10>
- Hutchins WJ, Somers HL (1992) An introduction to machine translation. Academic, London
- Jakobsen AL (2002) Translation drafting by professional translators and by translation students. In: Empirical translation studies: process and product. *Copenhagen studies in language* 27, pp 191–204
- Jakobsen AL (2003) Effects of think aloud on translation speed, revision, and segmentation. In: Alves F (ed) *Triangulating translation. Perspectives in process oriented research*. John Benjamins, Amsterdam and Philadelphia, pp 69–95
- Jakobsen AL (2006) Research methods in translation—Translog. In: Sullivan KPH, Lindgren E (eds) *Computer keystroke logging and writing: methods and applications*. Elsevier, Oxford, pp 95–105
- Krings HP (2001) In: Koby GS et al (eds) *Repairing texts. Empirical investigations of machine translation post-editing processes*. Kent, OH, Kent State UP
- Lagoudaki E (2006) Translation memory systems: enlightening users' perceptive. Translation memories survey 2006. Imperial College London. http://isg.urv.es/library/papers/TM_Survey_2006.pdf
- Lee J, Liao P (2011) A comparative study of human translation and machine translation with post-editing. *Compilat Translat Rev* 4(2):105–149
- Martín-Mor A (2011) La interferència lingüística en entorns de Traducció Assistida per Ordenador. Doctoral thesis, Universitat Autònoma de Barcelona, Bellaterra
- Morado Vázquez L, Torres del Rey J (2011) The relevance of metadata during the localisation process—an experiment. Paper presented at the international T3L conference: tradumatica, translation technologies & localisation, Universitat Autònoma de Barcelona, Spain, 21–22 June, (unpaginated)
- Mossop B (2001) *Revising and editing for translators*. St Jerome, Manchester

- Mossop B (2007) Empirical studies of revision: what we know and need to know. *J Spec Trans* 8 (unpaginated)
- O'Brien S (2006) Eye-tracking and translation memory matches. *Perspectives: Studies in Translatology* 14(3):185–205
- O'Brien S (2009) Eye tracking in translation-process research: methodological challenges and solutions. In: *Copenhagen studies in language* 38. Samfundslitteratur, Copenhagen, pp 251–266
- O'Brien S (2011) Collaborative translation. In: Gambier Y, Van Doorslaer L (eds) *Handbook of translation studies*, vol 2. John Benjamins, Amsterdam and Philadelphia, pp 17–20
- Piróth A (2011) Translation automation survey among translators. <http://www.pirothattila.com/APiroth-MT-Survey.pdf>
- Plitt M, Masselot F (2010) A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics* 93:7–16
- Pym A (2004) *The moving text: localization, translation, and distribution*. John Benjamins, Amsterdam and Philadelphia
- Pym A (2009) Using process studies in translator training: self-discovery through lousy experiments. In: Göpferich S, Alves F, Mees IM (eds) *Methodology, technology and innovation in translation process research*. Samfundslitteratur, Copenhagen, pp 135–156
- Pym A (2011) Translation research terms. A tentative glossary for moments of perplexity and dispute. In: Pym A (ed) *Translation research projects 3*. Intercultural Studies Group, Tarragona, pp 75–110
- Pym A (2012a) Democratizing translation technologies—the role of humanistic research. In: Cannavina V, Fellet A (eds) *Language and translation automation conference*. Rome, The Big Wave, pp 14–29
- Pym A (2012b) Translation skill-sets in a machine-translation age. Paper presented to the 16th international symposium on interpretation and translation teaching. Graduate institute of cross-cultural studies, Fu Jen Catholic University, Taiwan, January 14–15, 2012. <http://usuaris.tinet.cat/apym/on-line/training/2012.competence-pym.pdf>
- Romaine M, Richardson J (2009) State of the translation industry: smarter, more casual. myGengo, Tokyo. <http://mygengo.com/report/download/2009/>
- Shreve GM, Angelone E (2010) *Translation and cognition*. John Benjamins, Amsterdam & Philadelphia
- Teixeira CSC (2011) Knowledge of provenance and its effects on translation performance in an integrated TM/MT environment. In: *Proceedings of the 8th international NLPCS workshop—special theme: human-machine interaction in translation*. Copenhagen studies in language 41. Samfundslitteratur, Copenhagen, pp 107–118. <http://www.mt-archive.info/NLPCS-2011-Teixeira.pdf>
- Wallis J (2006) Interactive translation vs pre-translation in the context of translation memory systems: investigating the effects of translation method on productivity, quality and translator satisfaction. MA thesis, Ottawa
- Way A (2009) A critique of statistical machine translation. *Linguistica Antverpiensia* 8/2009. Special issue on Evaluation of Translation Technology: 17–41
- Webb LE (1998) *Advantages and disadvantages of translation memory*. MA thesis. Monterey: Monterey Institute of International Studies.
- Yamada M (2011a) The effect of translation memory databases on productivity. In: Pym A (ed) *Translation research projects 3*. Intercultural Studies Group, Tarragona, pp 63–73. http://isg.urv.es/publicity/isg/publications/trp_3-2011/yamada.pdf
- Yamada M (2011b) *Revising text: an empirical investigation of revision and the effects of integrating a TM and MT system into the translation process*. Doctoral thesis, Rikkyo University
- Zetzsche J (2012) Translation technology comes full circle. *MultiLingual* April 2012: 50–51

Editors Biographies

Amy Neustein, Ph.D., is Editor-in-Chief of the *International Journal of Speech Technology* and Series Editor of the *Springer Briefs in Speech Technology*. She has recently coedited (with Judith A. Markowitz) *Mobile Speech and Advanced Natural Language Solutions* (Springer 2013). She has also authored/edited books in the humanities, which have been cited in the *Chronicles of Higher Education*, as well as two prior Springer books in speech technology: *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics* and *Forensic Speaker Recognition: Law Enforcement and Counter-Terrorism*. She is a member of the visiting faculty at the National Judicial College since 1985, and a member of MIR (Machine-Intelligence Research) Labs since 2010. She is the recipient of several distinguished awards: pro Humanitate Literary Award; Information Technology: New Generations (Medical Informatics) Award; the Los Angeles County Supervisor Humanitarian Award; and the Woman of Valor: Lifetime Achievement Award. She is the CEO and Founder of Linguistic Technology Systems, located in Fort Lee, New Jersey.

Judith A. Markowitz, Ph.D., has been a leading analyst and thought leader in the speech-processing industry for over 25 years. She has recently coedited (with Amy Neustein) *Mobile Speech and Advanced Natural Language Solutions* (Springer 2013). Her prior technical publications include two books (*Lexical-Semantic Relations* and *Using Speech Recognition*), newsletters, reports, and numerous articles. She has been an invited expert to the World Wide Web Consortium (W3C)'s Voice Browser Working Group. She has served on two American National Standards Institute (ANSI) committees and was lead editor of INCITS 456, an ANSI standard for voice biometrics. In 2003, *Speech Technology Magazine* named her one of the top ten leaders in speech; in 2006 she was elevated to IEEE Senior Member status. In addition to her work in speech processing, she has recently served as the national president of the Lambda Literary Foundation. She is the president of J. Markowitz, Consultants, located in Chicago, Illinois.