

HUMAN  
COGNITIVE  
PROCESSING

## Analological Modeling

Edited by  
Royal Skousen  
Deryle Lonsdale  
Dilworth B. Parkinson

# Analogical Modeling

HUMAN COGNITIVE PROCESSING is a forum for interdisciplinary research on the nature and organization of the cognitive systems and processes involved in speaking and understanding natural language (including sign language), and their relationship to other domains of human cognition, including general conceptual or knowledge systems and processes (the language and thought issue), and other perceptual or behavioral systems such as vision and non-verbal behavior (e.g. gesture). 'Cognition' should be taken broadly, not only including the domain of rationality, but also dimensions such as emotion and the unconscious. The series is open to any type of approach to the above questions (methodologically and theoretically) and to research from any discipline, including (but not restricted to) different branches of psychology, artificial intelligence and computer science, cognitive anthropology, linguistics, philosophy and neuroscience. It takes a special interest in research crossing the boundaries of these disciplines.

### **Editors**

Marcelo Dascal, *Tel Aviv University*

Raymond W. Gibbs, *University of California at Santa Cruz*

Jan Nuyts, *University of Antwerp*

### *Editorial address*

Jan Nuyts, University of Antwerp, Dept. of Linguistics (GER),  
Universiteitsplein 1, B 2610 Wilrijk, Belgium.

E-mail: [nuyts@uia.ua.ac.be](mailto:nuyts@uia.ua.ac.be)

### **Editorial Advisory Board**

Melissa Bowerman, *Nijmegen*; Wallace Chafe, *Santa Barbara, CA*;

Philip R. Cohen, *Portland, OR*; Antonio Damasio, *Iowa City, IA*;

Morton Ann Gernsbacher, *Madison, WI*; David McNeill, *Chicago, IL*;

Eric Pederson, *Eugene, OR*; François Recanati, *Paris*;

Sally Rice, *Edmonton, Alberta*; Benny Shanon, *Jerusalem*;

Lokendra Shastri, *Berkeley, CA*; Dan Slobin, *Berkeley, CA*;

Paul Thagard, *Waterloo, Ontario*

### **Volume 10**

Analogical Modeling: An exemplar-based approach to language

Edited by Royal Skousen, Deryle Lonsdale and Dilworth B. Parkinson

# Analogical Modeling

An exemplar-based approach  
to language

*Edited by*

Royal Skousen

Deryle Lonsdale

Dilworth B. Parkinson

Brigham Young University, Provo, Utah

**John Benjamins Publishing Company**  
Amsterdam/Philadelphia





™ The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

## Library of Congress Cataloging-in-Publication Data

Analogical Modeling : An exemplar-based approach to language / edited by Royal Skousen, Deryle Lonsdale and Dilworth B. Parkinson.

p. cm. (Human Cognitive Processing, ISSN 1387-6724 ; v. 10)

“Much of the Language, held at Brigham Young University on 22-24 March 2000” p. 8. Includes bibliographical references and indexes.

1. Analogy (Linguistics) 2. Psycholinguistics. I. Skousen, Royal. II. Lonsdale, Deryle. III. Parkinson, Dilworth B., 1951- IV. Conference on Analogical Modeling of the Language (2000: Brigham Young University) V. Series.

P299.A48 A53 2002

417'.7-dc21

2002028283

ISBN 90 272 2362 9 (Eur.) / 1 58811 302 7 (US) (Hb; alk. paper)

© 2002 – John Benjamins B.V.

No part of this book may be reproduced in any form, by print, photoprint, microfilm, or any other means, without written permission from the publisher.

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands  
John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

# Table of contents

List of contributors	IX
Introduction	1
<i>Royal Skousen</i>	
<b>I. The basics of Analogical Modeling</b>	
CHAPTER 1	
An overview of Analogical Modeling	11
<i>Royal Skousen</i>	
CHAPTER 2	
Issues in Analogical Modeling	27
<i>Royal Skousen</i>	
<b>II. Psycholinguistic evidence for Analogical Modeling</b>	
CHAPTER 3	
Skousen's analogical approach as an exemplar-based model of categorization	51
<i>Steve Chandler</i>	
<b>III. Applications to specific languages</b>	
CHAPTER 4	
Applying Analogical Modeling to the German plural	109
<i>Douglas J. Wulf</i>	
CHAPTER 5	
Testing Analogical Modeling: The /k/~/Ø alternation in Turkish	123
<i>C. Anton Rytting</i>	

#### IV. Comparing Analogical Modeling with TiMBL

##### CHAPTER 6

A comparison of two analogical models: Tilburg Memory-Based Learner versus Analogical Modeling 141

*David Eddington*

##### CHAPTER 7

A comparison of Analogical Modeling to Memory-Based Language Processing 157

*Walter Daelemans*

##### CHAPTER 8

Analogical hierarchy: Exemplar-based modeling of linkers in Dutch noun-noun compounds 181

*Andrea Krott, Robert Schreuder, and R. Harald Baayen*

#### V. Extending Analogical Modeling

##### CHAPTER 9

Expanding  $k$ -NN analogy with instance families 209

*Antal van den Bosch*

##### CHAPTER 10

Version spaces, neural networks, and Analogical Modeling 225

*Mike Mudrow*

##### CHAPTER 11

Exemplar-driven analogy in Optimality Theory 265

*James Myers*

##### CHAPTER 12

The hope for analogous categories 301

*Christer Johansson*

**VI. Quantum computing and the exponential explosion**

## CHAPTER 13

Analogical Modeling and quantum computing 319

*Royal Skousen***VII. Appendix**

## CHAPTER 14

Data files for Analogical Modeling 349

*Deryle Lonsdale*

## CHAPTER 15

Running the Perl/C version of the Analogical Modeling program 365

*Dilworth B. Parkinson*

## CHAPTER 16

Implementing the Analogical Modeling algorithm 385

*Theron Stanford*

Index 411



## List of contributors

R. Harald Baayen  
baayen@mpi.nl  
Associate Professor of  
General Linguistics  
University of Nijmegen  
The Netherlands

Steve Chandler  
chandler@uidaho.edu  
Associate Professor of English  
University of Idaho  
Moscow, Idaho, USA

Walter Daelemans  
walter.daelemans@uia.ua.ac.be  
Professor of  
Computational Linguistics  
University of Antwerp, Belgium

David Eddington  
davee@unm.edu  
Associate Professor of  
Spanish Linguistics  
University of New Mexico  
Albuquerque, New Mexico, USA

Christer Johansson  
christer.johansson@lili.uib.no  
Associate Professor of  
Computational Linguistics  
University of Bergen, Norway

Andrea Krott  
akrott@ualberta.ca  
Department of Linguistics  
University of Alberta  
Edmonton, Alberta, Canada

Deryle Lonsdale  
lonz@byu.edu  
Department of Linguistics  
and English Language  
Brigham Young University  
Provo, Utah, USA

Mike Mudrow  
mudrow@cc.usu.edu  
Department of Languages  
and Philosophy  
Utah State University  
Logan, Utah, USA

James Myers  
Lngmyers@ccunix.ccu.edu.tw  
Graduate Institute of Linguistics  
National Chung Cheng University  
Taiwan, Republic of China

Dilworth B. Parkinson  
dilworth\_parkinson@byu.edu  
Professor of Arabic  
Brigham Young University  
Provo, Utah, USA

C. Anton Rytting  
rytting.1@osu.edu  
Department of Linguistics  
Ohio State University  
Columbus, Ohio, USA

Robert Schreuder  
Rob.Schreuder@mpi.nl  
Professor of Psycholinguistics  
University of Nijmegen  
The Netherlands

Royal Skousen  
royal\_skousen@byu.edu  
Professor of Linguistics  
and English Language  
Brigham Young University  
Provo, Utah, USA

Theron Stanford  
shixilun@yahoo.com  
Department of Asian and  
Near Eastern Languages  
Brigham Young University  
Provo, Utah, USA

Antal van den Bosch  
Antal.vdnBosch@kub.nl  
Faculty of Arts  
Tilburg University  
The Netherlands

Douglas J. Wulf  
wulf@u.washington.edu  
Department of Linguistics  
University of Washington  
Seattle, Washington, USA

# Introduction

Royal Skousen

## Competing models of language description

Much of the debate in language description involves choosing between alternative methods of describing language. A number of completely different systems for predicting language behavior have been developed in the past two decades.

The original, traditional approach (actually centuries old) has been to devise specific rules. In such a declarative approach, rules are used to divide up the contextual space into distinct conditions and then to specify the corresponding behavior for each of those conditions. Such a traditional approach seems unavoidable for telling someone else how something behaves. Although rules may not actually be used by speakers in producing their own language, we still use rules to tell others how language works (even when describing the results of analogical modeling). In general, rule approaches form the basis for most descriptive systems, such as expert systems for doing medical diagnoses. The idea behind such medical systems is to specify for a given set of symptoms a corresponding disease or dysfunction.

In contrast to declarative, rule-based approaches, there are two different kinds of procedural, non-rule systems that have been developed in the last part of the twentieth century: neural networks and exemplar-based systems. These procedural approaches have no explicit statement of regularities. They use examples to set up a system of connections between possible predictive variables or they directly use the examples themselves to predict behavior. A procedural approach typically predicts behavior for only a given item. Generally speaking, directly accessible information about what variables lead to the prediction is not available – thus, the non-declarative nature of these approaches.

Neural networks had their beginnings in the middle of the twentieth century, but were then ignored for several decades until researchers realized that output nodes did not have to be directly predicted from input nodes. Hidden levels of nodes could be used to represent various levels of activation and thus learn virtually any kind of relationship between variables, providing certain possibilities



(such as back propagation) were allowed between levels of nodes. Although many of the kinds of connections between nodes seem impossible from a neurological point of view, the “neurological temptation” has been strong since the 1980s when it was shown that neural networks (or connectionism) could make interesting and robust predictions about language behavior – in contrast to brittle (rigid yet fragile) rule approaches. Rumelhart and McClelland, in particular, argued for “parallel distributed processing”, with its special emphasis on the non-accessibility of exemplars. Exemplars would, of course, be used to train a system, but the specific exemplars were not directly stored. Instead, the neural network would discover various associations and disassociations between the variables of the exemplars. Some connections between combinations of variables would be activated, others deactivated, or even negatively activated. The notion that the system would be “distributed” was particularly emphasized. Basically, this meant that the variable relationships would be spread across the system, and thus the specific ability to recover the original exemplars would be impossible. Rumelhart and McClelland emphasized this property as a plus; the distributed characteristic meant that their system would be robust and could make predictions when the given input was degraded or incomplete.

An opposing procedural approach was also developed in the 1980s, one that directly stores exemplars and then accesses them to make predictions. Most exemplar-based or instance-based approaches that were developed made their predictions in terms of finding the nearest neighbor (or  $k$  neighbors) to a given input, with of course the possibility that the “nearest neighbor” could be identical to the given input. It was recognized early on that some neighbors were “nearer” to the given input than other neighbors (even though the number of differing variables from the given input might be the same). This recognition has led researchers to try to determine the significance of the variables in predicting the outcome and to use some weighting of the variables to adjust the nearness of the neighbors.

A different approach to exemplars was taken by analogical modeling, as described by Skousen in the late 1980s. His explicit theory of analogical modeling of language differs considerably from traditional analogical approaches to language. One major problem with traditional analogy is that it is not explicit. Furthermore, virtually any item can serve as the exemplar for predicting behavior, although in practice the attempt is to first look to nearest neighbors for the preferred analogical source. But if proximity fails, one can almost always find some item considerably different from the given item that can be used to analogically predict the desired outcome. In other words, if needed, virtually any occurrence can serve as the analogical source.

Analogical modeling, on the other hand, will allow occurrences further away from the given context to be used as the exemplar, but not just any occurrence. Instead, the occurrence must occur in what is called a homogeneous supracontext. The analogical source does not have to be a near neighbor. The probability of an

occurrence further away acting as the analogical model is usually less than that of a closer occurrence, but this probability is never zero (providing the occurrence is in a homogeneous supracontext).

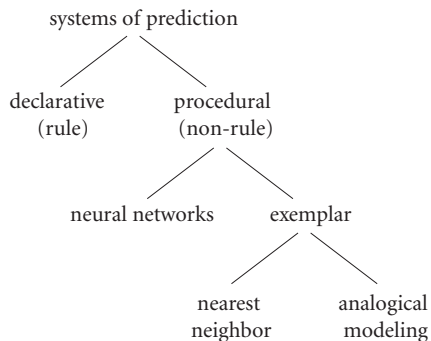
Further, the ability to use all the occurrences in homogeneous regions of the contextual space directly accounts for the gang effects we find when we describe either categorical or regular/exceptional behavior. In other words, we are able to predict “rule-governed” behavior (plus a little fuzziness) whenever the data behaves as if there is a rule.

Analogical modeling does not require us to determine in advance which variables are significant and the degree to which these variables determine the outcome (either alone or in various combinations). Nearest-neighbor approaches are like traditional analogical practice in that they try to predict behavior by using the most similar occurrences to the given context. But unless some additional information is added, the leakage across categorical boundaries and in regions close to exceptions will be too large.

As a result, nearest-neighbor approaches frequently try to correct for this excessive fuzziness by determining the overall importance of each variable. One can determine, for instance, the information gain (or other measures of reducing entropy) for each variable. Such added information requires a training period to determine this information, and in this regard makes these approaches like connectionism.

Analogical modeling, on the other hand, does not have a training stage except in the sense that one must obtain a database of occurrences. Predictions are made “on the fly”, and all variables are considered apriorily equal (with certain limitations due to restrictions on short-term memory). The significance of a variable is determined locally – that is, only with respect to the given context. Gang effects are related to the location of the given context and the amount of resulting homogeneity within the surrounding contextual space.

We may characterize these different types of language prediction by the following categorical representation:



Obviously, this decision tree provides a categorical (rule-based) description! Such seem unavoidable when we want to tell someone about some behavior.

And as we might expect, various combinations of these possibilities have been developed. For instance, some researchers have discussed the possibility of using neural networks to store exemplars – in other words, neural networks may need to be non-distributed, or at least we may need to allow for that possibility. Therefore the categorical boundaries between these differing systems of prediction may not be sharply defined.

Another, even more general combination has been promoted by Steven Pinker and his colleagues – namely, the dual-route approach to language description. Pinker argues that speakers of English use both rules and associative networks to predict the past tense in English: a syntactic-like rule to handle the regular past-tense form (add *d* to the verb stem), plus a lexical-based associative network involving memory of exemplars to predict irregular verbs (thus *sang* and *sung* are connected with *sing*). In other words, Pinker claims that a dual-route approach (involving both rules and non-rules) is used to predict language, in contrast to the main argument of the procedural approaches that all behavior can be predicted without rules (the single-route approach).

## Overview of this book

In addition to its tutorials (one in the first paper, three in the appendix), this book brings together contributions that reflect current research related to analogical modeling (AM). They are representative of the breadth and the detail with which analyses can be carried out within the AM paradigm as well as in comparing it to other approaches. These contributions should serve to initiate further research, stimulate thought about new applications, and encourage investigation into a wider variety of linguistic applications and wider array of languages.

### PART I. The basics of Analogical Modeling

We begin the book by covering the basics of analogical modeling (AM). Royal Skousen provides an overview of how AM works. He gives a succinct description of AM, then he goes through a simple tutorial of how AM would predict the pronunciation of the letter *c* in English (at the beginning of a word). The basic terminology and methods of AM are described in terms of this artificial example. Skousen also reviews some of the empirical evidence that supports analogical modeling and shows how AM differs from other exemplar-based approaches to language.

This first chapter is followed by one that discusses issues in analogical modeling. In that chapter, Royal Skousen goes over the main points of debate and discussion in current research on AM. The main issues are:

1. Is the principle of homogeneity actually necessary?
2. Should prediction be restricted to locally significant variables?
3. Should “unimportant” variables be maintained in predicting behavior?
4. What role does imperfect memory play in predicting behavior?
5. What counts as evidence for and against an analogical prediction?
6. Should all variables be given the same weight?
7. Should prediction be based on choosing an exemplar or a pointer to an exemplar?
8. How should the outcomes and variables in the dataset be categorized?
9. How do we deal with the exponential explosion that is inherent in analogical modeling?

In-depth discussion of these issues appears throughout the book.

## PART II. Psycholinguistic evidence for Analogical Modeling

In this section, Steve Chandler discusses the psycholinguistic evidence, both for and against various procedural approaches to language. He reviews numerous psycholinguistic experiments regarding language learning and considers the implications of these experiments for connectionist approaches (neural networks), prototype theory, nearest-neighbor approaches (instance- or exemplar-based), and analogical modeling.

## PART III. Applications to specific languages

In this section, we have two in-depth analogical analyses of morphology. First, Doug Wulf applies analogical modeling to the complex problem of plural formation in German. The complexity of this problem arises, in part, because there are two inter-related markers of plurality (umlauting and suffixation). Wulf also shows that by using only the most frequent examples, one can predict adult-like plural formation in German.

In the second paper, Anton Rytting uses analogical modeling to predict the /k/-Ø alternation in the Turkish nominal system. He concentrates on the ability of analogical modeling to predict the behavior of foreign loan words and the results of experiments with nonce words.

## PART IV. Comparing Analogical Modeling with TiMBL

In this section, we have three papers that compare analogical modeling with a specific exemplar-based version of nearest-neighbor prediction – namely, the Tilburg Memory-Based Learner (TiMBL), developed by Walter Daelemans and his colleagues at Tilburg University in the Netherlands. TiMBL (as well as a reference guide to it) is available on the internet at <<http://ilk.kub.nl/software.html>>.

In the first paper, David Eddington applies both AM and TiMBL to various problems in Spanish morphology (gender assignment, diminutive formation, and stress assignment) and generally finds that both approaches behave statistically the same in their ability to self-predict the behavior of each given dataset. However, in the case of stress assignment, AM is considerably more successful in predicting the direction of change – namely, the replacement of irregularly stressed words with the expected regular stress patterns of the language.

Walter Daelemans then compares AM with various versions of the TiMBL nearest-neighbor approach. The main goal of these comparisons is to determine which basic approach (TiMBL or AM) is better able to internally self-predict the behavior of the language data. In a number of cases, he is able to show that by using information gain, TiMBL is better able to self-predict the behavior. Of course, the ultimate question in the debate between AM and TiMBL is an empirical one: which approach best represents the dynamics of language behavior – that is, which one can best predict, for instance, the kinds of errors that children make in learning the language, or the dialectal and historical changes that have occurred in the language.

Finally in this section, Andrea Krott and her colleagues Rob Schreuder and Harald Baayen discuss compound formation in Dutch, once more testing whether AM or TiMBL is better at self-predicting which of three linkers (*-en-*, *-s-*, and *-Ø-*) is selected when forming nominal compounds in Dutch. They determine that both AM and TiMBL are very helpful in determining which variables correctly predict the choice of linker.

## PART V. Extending Analogical Modeling

In this section, four different papers discuss how analogical modeling might be related to other approaches to language description. In the first paper, Antal van den Bosch takes up the question of how a nearest-neighbor approach (such as TiMBL) might be expanded so that some aspects of AM's homogeneity might be available in nearest-neighbor predictions. Van den Bosch considers defining nearest neighbors in terms of instance families of same-class behavior rather than just atomistic examples.

Mike Mudrow compares AM with various versions of a standard instance-based learner (the IAC model of Grossberg) and a simple neural network model

(SimNet). He then applies the SimNet model to basically the same basic dataset that Skousen developed to predict the past tense in Finnish and finds that for probabilistic predictions SimNet appears to give more accurate transitions in probabilistic behavior. Mudrow also compares the predictions of AM and SimNet with respect to nominal compounding in Danish.

James Myers considers the possibility of doing optimality theory in terms of analogical modeling. Recent developments in optimality theory are beginning to recognize the need for exemplar-based predictions, so Myers investigates the problems of trying to reformulate optimality theory from an analogical perspective.

Finally, Christer Johansson considers the problem of categorization. All exemplar-based approaches to language description define datasets in terms of apriorily defined categories (for both variables and outcomes). Johansson looks into the possibility of letting the analogical system itself define the categories rather than simply assuming that the variables and outcomes are essentially innate or defined from the beginning.

## PART VI. Quantum computing and the exponential explosion

In this closing paper, Royal Skousen discusses the problem of the exponential explosion, an inherent problem in analogical modeling. The current algorithm for AM becomes inefficient in dealing with large numbers of variables; basically, each additional variable doubles both the memory and time requirements of the program. Skousen further argues that this problem of the exponential explosion can be found in all theories of language description, although this difficulty is often disguised by simply assuming that inter-relationships between variables can be ignored or accounted for otherwise.

Skousen proposes that the ultimate solution to this exponential problem of language description will involve quantum computing (QC). He first outlines a number of striking conceptual and mathematical similarities between AM and quantum mechanics, which imply that the exponential problem in AM may be treated tractably using QC.

Quantum computing requires reversibility, which means that in applying QC to language behavior we have to keep track of specific input and that this input must be fully recoverable at the end of computation. Using QC for language prediction thus implies that such a system will be exemplar-based, even if the exemplars are somehow used to derive rules or neural-like associations.

## Appendix

The appendix provides help in constructing datasets and running the computer program for analogical modeling. Deryle Lonsdale first describes how one goes about setting up an appropriate dataset for analogical prediction. He describes the kinds of variables that can be specified, as well as the associated outcomes assigned to those variables.

Dil Parkinson then shows how to run the Perl/C version of the analogical modeling program. In particular, he describes the various parameters that can be set in order to derive different analogical predictions.

And finally, Theron Stanford provides a description of how the current analogical modeling program works. This program can be downloaded from the internet at <http://humanities.byu.edu/am/>.

## Acknowledgments

Much of the original impetus for this book results from the Conference on Analogical Modeling of Language, held at Brigham Young University on 22–24 March 2000. The purpose of this conference was to bring together researchers in Royal Skousen's theory of analogical modeling as well as various other exemplar-based approaches to describing language. In virtually every instance, the papers in this book include subsequent work by the authors, but nonetheless the resulting papers are motivated by the discussions and presentations at the conference.

We, the editors, wish to gratefully acknowledge the support of Brigham Young University in putting on this conference, and especially the following entities for their generous financial and staff support: the Department of English, the Department of Linguistics and English Language, the Department of Asian and Near Eastern Languages, the College of Humanities, and the David M. Kennedy Center for International Studies. We also wish to thank Dan Jewell for his help in arranging the logistics of the conference.

Finally, we acknowledge the subsequent support of the Department of Asian and Near Eastern Languages with financial help in the editing of this book. We are especially grateful to Theron Stanford for his yeoman service in working with the editors to prepare the electronic text for typesetting by the John Benjamins Publishing Company.

PART I

## The basics of Analogical Modeling





## CHAPTER 1

# An overview of Analogical Modeling

Royal Skousen

### 1. The development of non-rule models to describe language

During the last two decades, as rule approaches have encountered difficulties in describing language behavior, several competing non-rule approaches to language have been developed. First was the development (or rejuvenation) of neural networks, more commonly known in linguistics as connectionism and best exemplified by the work of McClelland, Rumelhart, and the PDP Research Group (1986) in what they call “parallel distributed processing” (PDP). More recently, some researchers (such as David Aha and Walter Daelemans) have turned to exemplar-based systems (sometimes known as instance-based systems or “lazy learning”) to describe language behavior (Aha, Kibler, & Albert 1991; Daelemans, Gillis, & Durieux 1994). These exemplar-based learning systems involve hunting for the most similar instances (“nearest neighbors”) to predict language behavior. A more general theory of the exemplar-based approach is Royal Skousen’s analogical modeling of language, which permits (under well-defined conditions) even non-neighbors to affect language behavior.

These non-rule approaches have several advantages over the traditional rule approaches. First of all, they can be explicitly defined and are therefore testable. Second, they are procedurally defined – that is, they predict behavior for a given input, but do not declare any globally-defined rules. The problem of knowing how to learn and then use a general rule to predict specific behavior is avoided. Third, these non-rule approaches are robust in the sense that they can make predictions when the input is not “well-formed” or when “crucial” variables are missing. In general, boundaries between different behaviors (or outcomes) do not have to be precise; fuzzy boundaries and leakage across boundaries are in fact expected.

The fundamental works on analogical modeling (AM) are two books by Skousen. The first one, *Analogical Modeling of Language* (Skousen 1989), provides a complete, but basic, outline of the approach (Chapter 2) and then shows how the theory can be applied to derive various language properties (Chapter 3) as well

as deal with several theoretical language issues (Chapter 4). In Chapter 5, Skousen provides an in-depth analysis of past-tense formation in Finnish. In particular, he shows how analogical modeling, unlike traditional rule approaches, is able to describe the complex historical development of the Finnish past tense. The second book, *Analogy and Structure* (Skousen 1992), is a mathematical description of both rule-based and analogical approaches to describing behavior.

## 2. A succinct description of Analogical Modeling

Since analogical modeling is a procedural approach, predictions are always based on a dataset of occurrences. Each occurrence is specified in terms of a set of variables and an assigned outcome for that specific assignment of variables. A given set of variables can occur more than once in a dataset, as can the assigned outcome. (In fact, such repetition is normal.) For the purposes of discussion, we will assume that  $n$  variables are specified.

In order to make a prediction, we always do it in terms of a given context, where the variables are specified, but for which no outcome is given. Usually all  $n$  variables are specified in the given context, but this is not necessary. Our task is to predict the outcome for this given context in terms of the occurrences found in the dataset. For our purposes here, we will let  $m$  stand for the number of specified variables in the given context, where  $0 \leq m \leq n$ .

For each subset of variables defined by the given context, we determine which occurrences in the dataset occur with that subset. Each of these subsets of variables is called a supracontext. Given  $m$  variables in the given context, we have a total of  $2^m$  supracontexts.

Our problem is to determine the homogeneity (or its opposite, the heterogeneity) of each supracontext defined by the given context. Basically, a supracontext is homogenous if all its possible subcontexts behave identically. In predicting the outcome for the given context, we only apply information found in the homogeneous supracontexts. All heterogeneous supracontexts are ignored.

We determine whether a supracontext is homogeneous by using a nonlinear statistical procedure based on measuring the number of disagreements between different occurrences within the supracontext. We adopt a conceptually simple statistical procedure for determining the homogeneity of the supracontext – namely, if no subcontext of the supracontext increases the number of disagreements, the supracontext is homogeneous. Otherwise, the supracontext is heterogeneous. This measure ends up minimizing the number of disagreements in the supracontext.

Using this natural statistic, it is easy to show that there are only two types of homogeneous supracontexts for a given context: (1) the supracontext is determin-

istic; or (2) the supracontext is non-deterministic but there is no occurrence in the supracontext that is closer to the given context than any other occurrence in the supracontext.

These homogeneous supracontexts form what is called the analogical set. The final step is to randomly select one of the occurrences in the analogical set and make our prediction based on the outcome assigned to this occurrence. One alternative to random selection is to select the most frequent outcome. This method is referred to as selection by plurality.

### 3. A tutorial example of Analogical Modeling

In this introduction, analogical modeling will be described in terms of a simple example from English spelling. In this example our overall task will be to predict the pronunciation of the *c* letter in initial position in words of English.

In analogical modeling, predictions are always based on a dataset of occurrences. For our spelling example, we will make our predictions from the following (simplified) dataset:

outcome	variables	specification
k-c	a k e	cake
k-c	a l l	call
k-c	a n 0	can
k-c	a r 0	car
k-c	a t 0	cat
s-c	e l l	cell
s-c	e n t	cent
s-c	e r t	certain
č-c	h e c	check
k-c	l o s	close
k-c	l o u	cloud
č-c	h i n	chin
č-c	h u r	church
s-c	i r c	circle
s-c	i r c	circus
s-c	i t y	city
k-c	l a m	clam
k-c	l e a	clear
k-c	l o s	close
k-c	o a t	coat

k-c	o i n	coin
k-c	o l d	cold
k-c	o m e	come
k-c	o u n	count
k-c	o w 0	cow
k-c	r e a	cream
k-c	r o s	cross
k-c	r y 0	cry
k-c	u p 0	cup
k-c	u r e	cure
k-c	u t 0	cut
s-c	y c l	cycle
s-c	y c l	cyclone
s-c	y m b	cymbal

Occurrences in the dataset are specified in terms of a set of variables and an associated outcome for each specific assignment of variables. For instance, for the first occurrence listed (*cake*), the outcome is *k-c*, which means for this word that the *k* pronunciation is assigned to the initial letter *c*. Two other possible outcomes are listed in the dataset, *s-c* (as in *cell*) and *č-c* (as in *check*). Following the listing of the outcome, three variables are given – namely, the next three letters in the word (thus *a*, *k*, and *e* for *cake*). And finally, we specify the complete spelling of the word (that is, *cake*).

In this simple dataset, we restrict our variables to the first three letters after the initial *c*. If a word is short (such as *can*), we fill the empty variables with the null symbol 0 (a zero). For longer words, subsequent letters in the word are ignored. Thus for the word *certain*, the last three letters (*ain*) are left unspecified.

A given assignment of variables can occur more than once in a dataset, as can the associated outcome. In fact, such repetition is normal. In general, we will assume that  $n$  variables are specified. In the simple spelling example,  $n$  equals 3.

In order to make a prediction, we always do it in terms of a given context, where the  $n$  variables are specified, but for which no outcome is given. For instance, suppose we wish to predict the pronunciation of the initial *c* for the word *ceiling*. The given context will be the following three letters after the *c* – that is, *e*, *i*, and *l*.

For each subset of variables defined by the given context, we determine which occurrences in the dataset occur with that subset. Each of these subsets of variables is called a supracontext. Given  $n$  variables in the given context, we have a total of  $2^n$  supracontexts. Thus the number of supracontexts ( $2^n$ ) is an exponential function of the number of variables ( $n$ ).

For the given context *ceiling*, we have specified the three letters following the initial *c* as variables: *e*, *i*, and *l*. This gives us a total of  $2^3$  or 8 possible supracontexts:

supracontexts of *ceiling*

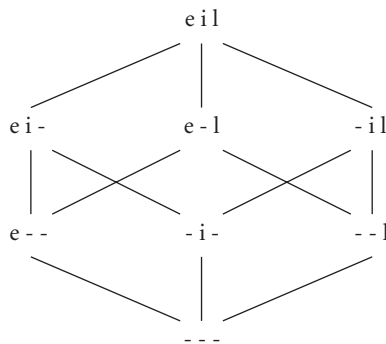
all three variables:	eil
two variables, one ignored:	ei-, e-l, -il
one variable, two ignored:	e--, -i-, --l
all three variables ignored:	---

For each of these supracontexts we determine which occurrences in the dataset occur in that supracontext:

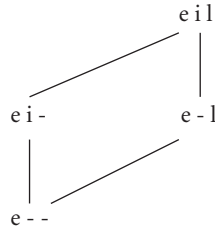
	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
eil	–	–	–	
ei-	–	–	–	
e-l	–	1	–	<i>cell</i>
-il	–	–	–	
e--	–	3	–	<i>cell, cent, certain</i>
-i-	1	–	1	<i>chin, coin</i>
--l	1	3	–	<i>call, cell, cycle, cyclone</i>
---	21	9	3	<the whole dataset>

For 3 of the 8 supracontexts, there are no occurrences from the dataset (*eil*, *ei-*, and *-il*). And for the most general supracontext (namely, when all three variables are ignored), we get every occurrence in the dataset.

Typically, the whole class of  $2^n$  supracontexts can be represented as a partially ordered lattice:



By following the connections upward in the lattice, we can determine the subcontexts for any particular supracontext in the lattice. Thus we have the following 4 subcontexts for the supracontext *e--*:



By definition, we count the supracontext itself as one of the subcontexts.

Our problem is to determine the homogeneity (or its opposite, the heterogeneity) of each supracontext defined by the given context. Basically, a supracontext is homogenous if all its possible subcontexts behave identically. In predicting the outcome for a given context, we use only the occurrences in homogeneous supracontexts. All heterogeneous supracontexts are ignored.

In analogical modeling, there can be two different types of homogeneous supracontexts for a given context: either (1) the supracontext is deterministic (only one outcome occurs), or (2) the supracontext is non-deterministic but all the occurrences occur within only one subcontext of the supracontext.

When we consider the supracontexts for our example involving *ceiling*, we note that there are two deterministic supracontexts, *e-l* and *e--*. The more general supracontext *e--* is homogeneous because it contains only examples of the *s-c* outcome. There can be no evidence that any subcontext of *e--* behaves any differently because the behavior of *e--* is deterministic. Similarly, the subcontext *e-l* also acts as a homogeneous supracontext since it too has only one kind of outcome (even if there is just one occurrence):

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
e i l	-	-	-	
e i -	-	-	-	
e - l	-	1	-	<i>cell</i>
e - -	-	3	-	<i>cell, cent, certain</i>

In addition, our example for *ceiling* has one non-deterministic homogeneous supracontext, *-i-*. For this supracontext, more than one outcome is found (thus its behavior is non-deterministic). Yet every subcontext of this supracontext is either empty or identical to the supracontext's behavior, so we can find no subcontext that behaves differently than the supracontext itself:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>
e i l	-	-	-
e i -	-	-	-

-il	-	-	-	
-i-	1	-	1	<i>chin, coin</i>

In this case, every subcontext of *-i-* (except the supracontext itself) is empty.

It is also possible that some of the subcontexts are identical to the supracontext, as in the following made-up example from a different dataset that lacks the occurrences for *chin* and *coin*, but instead has *chill* and *coil*:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
eil	-	-	-	
ei-	-	-	-	
-il	1	-	1	<i>chill, coil</i>
-i-	1	-	1	<i>chill, coil</i>

In this example, both non-deterministic supracontexts *-il* and *-i-* would be homogeneous.

Returning to the supracontexts of *ceiling* (and our example dataset), we see that two of the supracontexts are heterogeneous, *-l* and *---*. The supracontext *-l* is heterogeneous because its subcontext *e-l* behaves differently (having only the *s-c* outcome) while *-l* has both the *k-c* and *s-c* outcomes. We mark each heterogeneous supracontext with an  $\times$ , thus reminding us to exclude such when we come to predict the outcome for the given context:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
eil	-	-	-	
e-l	-	1	-	<i>cell</i>
-il	-	-	-	
$\times$ -l	1	3	-	<i>call, cell, cycle, cyclone</i>

In addition to *-l*, the general supracontext *---* is also heterogeneous because every occurring subcontext behaves differently than the supracontext *---*:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
eil	-	-	-	
ei-	-	-	-	
e-l	-	1	-	<i>cell</i>
-il	-	-	-	
e- -	-	3	-	<i>cell, cent, certain</i>
-i-	1	-	1	<i>chin, coin</i>
$\times$ -l	1	3	-	<i>call, cell, cycle, cyclone</i>
$\times$ ---	21	9	3	<the whole dataset>

It is easy to demonstrate that if any supracontext is heterogeneous, then whenever this supracontext acts as a subcontext in a more general supracontext, heterogeneous



ity will be implied. For instance, we have already seen that *-l* is heterogeneous. From this we may deduce that the more general supracontext *---* is also heterogeneous since *-l* is one of its subcontexts. We refer to this deductive kind of heterogeneity as inclusive heterogeneity. Thus the general supracontext *---* is inclusively heterogeneous.

Whenever a given context actually occurs, the predicted behavior is determined by the behavior of that given context. If the behavior of the occurring given context is deterministic, then every other deterministic supracontext will also be homogeneous, as in this example for predicting *century*:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
ent	–	1	–	<i>cent</i>
en-	–	1	–	<i>cent</i>
e-t	–	2	–	<i>cent, certain</i>
-nt	–	1	–	<i>cent</i>
e--	–	3	–	<i>cell, cent, certain</i>
× -n-	1	1	–	<i>can, cent</i>
× --t	1	2	–	<i>cent, certain, coat</i>
× ---	21	9	3	<the whole dataset>

On the other hand, if the occurring given context is non-deterministic, the only other homogeneous supracontexts would have to have the exact same number of occurrences as the given context, as in the following artificial example (not based on our example dataset):

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
elt	5	5	–	
× el-	10	10	–	
× e-t	5	12	–	
-lt	5	5	–	
× e--				
× -l-				
× --t				
× ---				

We note from this example that we get heterogeneity even if a supracontext (*el-*) has exactly the same proportions as a less frequent, closer supracontext (*elt*):

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
elt	5	5	–	
× el-	10	10	–	

Only when the frequencies are the same do we get homogeneity for the more general supracontext:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>
elt	5	5	–
-lt	5	5	–

Obviously, the normal use of statistical significance cannot be used to get these results. In standard statistics, there is no evidence that the behavior is different when the proportions are the same. We can see this when we consider a chi-squared analysis of possible  $2 \times 2$  arrays based on the occurrences from the preceding example. In the following, the symbol  $\alpha$  represents the probability of getting at least the specified value for  $\chi^2$ :

	<i>k-c</i>	<i>s-c</i>		
elt	5	5		10
	5	5		10
× el-	10	10		20

$\chi^2 = 0.00 \quad \alpha = 1.00$

When the proportions are exactly the same, the probability of getting  $\chi^2$  greater than or equal to zero is, of course, one. Only when the proportions are considerably different is this probability reduced:

	<i>k-c</i>	<i>s-c</i>		
elt	5	5		10
	8	2		10
× el-	13	7		20

$\chi^2 = 1.98 \quad \alpha = 0.15$

	<i>k-c</i>	<i>s-c</i>		
elt	5	5		10
	10	0		10
× el-	15	5		20

$\chi^2 = 6.67 \quad \alpha = 0.01$

Unlike traditional statistics, in analogical modeling the non-deterministic supracontext *el-* is heterogeneous in each of these three cases since the frequency of the supracontext *el-* is always greater than the frequency of the occurring given context *elt*.

Ultimately, whether a supracontext is homogeneous or heterogeneous is determined by using a nonlinear statistical procedure based on measuring the number of disagreements between different occurrences within the supracontext. To do this we connect all the occurrences within a supracontext to each other by means of a system of pointers. For each pointer from one occurrence to another, we indicate

whether the pointer points to a different outcome (a disagreement) or to the same outcome (an agreement). We adopt a conceptually simple statistical procedure for determining the homogeneity of the supracontext – namely, if no subcontext of the supracontext increases the number of disagreements, the supracontext is homogeneous. Otherwise, the supracontext is heterogeneous. This measure ends up minimizing the number of disagreements (that is, the number of pointers to differing outcomes) in the supracontext. It turns out that this statistic is based on a quadratic measure of information with its reasonable restriction that language speakers get only a single chance to guess the correct outcome. This quadratic measure is unlike Shannon’s logarithmic measure of uncertainty, which is based on the idea that speakers get an unlimited number of chances to guess the correct outcome. In analogical modeling, homogeneity is defined in terms of minimizing quadratic uncertainty. This single principle accounts for all the specific cases of homogeneity and heterogeneity, even the non-deterministic ones with equal proportions but different frequencies.

The statistical procedure of minimizing the number of disagreements is also the most powerful statistical test possible. However, by introducing the notion of imperfect memory, this test can be made equivalent to standard statistical procedures, especially when the probability of remembering a given occurrence is one-half. This kind of statistic is referred to as a natural statistic since it is psychologically plausible and avoids any direct consideration of probability distributions, yet has the ability to predict stochastic behavior as if the underlying probability distribution is known. On the basis of this natural statistic, it can be deduced that there are only the two types of homogeneous supracontexts – either deterministic ones or non-deterministic ones with occurrences restricted to a single subcontext.

The homogeneous supracontexts form what is called the analogical set. The final step in analogical prediction is to randomly select one of the occurrences in the analogical set and make our prediction based on the outcome assigned to this occurrence. Theoretically this selection can be done in two different ways: (1) randomly select one of the occurrences found in any of the homogeneous supracontexts; or (2) randomly select one of the pointers pointing to an occurrence in any of the homogeneous supracontexts. In the first case, the probability of selecting a particular occurrence is based on its frequency of occurrence within the homogeneous supracontexts. In the second case, the probability of selecting a particular occurrence is based on the square of its frequency of occurrence within the homogeneous supracontexts. This squaring of the frequency is the result of using a system of pointers (equivalent to the quadratic measure of uncertainty) to select an occurrence.

There is an alternative to random selection. Instead of randomly choosing one of the occurrences in the analogical set, one can examine the overall chances for each outcome under random selection but then select the most frequent outcome.

This method is referred to as selection by plurality and is used to maximize gain (or minimize loss).

Returning to our original example for predicting the pronunciation of the initial *c* in *ceiling*, we get the following results in the analogical set:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	
eil	–	–	–	
ei-	–	–	–	
e-l	–	1	–	<i>cell</i>
-il	–	–	–	
e- -	–	3	–	<i>cell, cent, certain</i>
-i-	1	–	1	<i>chin, coin</i>
× - -l	1	3	–	
× - - -	21	9	3	

Note that the two heterogeneous supracontexts (- -l and - - -) are excluded; they are each marked with an *x* and their occurrences are not listed since they do not occur as exemplars.

We can predict the outcome by selecting either occurrences (linearly) or pointers (quadratically):

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	linear	squared
eil	–	–	–		
ei-	–	–	–		
e-l	–	1	–	0 1 0	0 1 0
-il	–	–	–		
e- -	–	3	–	0 3 0	0 9 0
-i-	1	–	1	1 0 1	2 0 2
× - -l	1	3	–		
× - - -	21	9	3		
	Totals			1 4 1	2 10 2

Ultimately, the prediction can be made using either random selection or selection by plurality:

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>
random selection	.17	.67	.17	.14	.71	.14
selection by plurality	0	1	0	0	1	0

As can be seen, the predicted outcome for *ceiling* always favors the *s* pronunciation for the initial *c*. The closest exemplar to the given *ceiling* is the word *cell* (which has the *s-c* outcome), yet other words are also found in the analogical set (such as *cent*

and *certain*, which predict the *s-c* outcome. In fact, a couple of other exemplars (*chin* and *coin*) predict the two other outcomes (*k-c* and *ċ-c*). We note that under random selection the *s-c* outcome occurs at least two-thirds of the time, no matter whether we randomly select one of the occurrences (the linear prediction) or one of the pointers (the squared prediction). Under selection by plurality, we get of course only the *s-c* outcome.

#### 4. Empirical validation of Analogical Modeling

Analogical modeling has been applied to a number of specific language problems. Derwing and Skousen (1994) have used analogical modeling to predict English past-tense formation, especially the kinds of errors found in children's speech. Derwing and Skousen first constructed a dataset of verbs based on the frequently occurring verbs in grade-school children's speech and writing. Initially they predicted the past-tense for verbs in terms of a dataset composed of only the 30 most frequent verbs (most of which were irregular verbs), then they continuously doubled the size of the dataset (from 30 to 60, to 122, to 244, to 488, and finally to 976). Derwing and Skousen discovered that when the dataset was small, the kinds of errors children typically make were predicted, but by the time the dataset reached the third doubling (at 244 verbs) stability had usually set in, and the expected adult forms (that is, the standard language forms) were predicted more than any other. For instance, the most common prediction for the verb *snow* was *snew* as long as the dataset had only 30 or 60 verbs, but with 122 verbs (after the second doubling) the prediction shifted to the regular *snowed* (with a 90% chance). With the full dataset of 976 verbs, the probability of predicting the regular *snowed* reached 99%. Similarly, *overflowed* was most commonly predicted for *overflow* until the third doubling (at 244 verbs), and *succame* for *succumb* (pronounced *succome*, of course) until the fourth doubling (at 488 verbs).

Analogical modeling (along with other exemplar-based systems and connectionism) has been criticized because it proposes a single-route approach to predicting the past-tense in English (see, for instance, Jaeger et al. 1996: 455–457, 477–478). Prasada and Pinker (1993) have argued, on the other hand, for a dual-route approach – that is, irregular verbs in English are processed differently than regular verbs. More specifically, they argue that irregular verbs are predicted in an analogical, lexically-based fashion, but that regular verbs are predicted by rule (namely, by syntactically adding some form of the regular past-tense ending *-ed*). Jaeger et al. 1996 further argued that there is information from neural activity in the brain for the dual-route approach. The main claim about analogical modeling in Jaeger et al. 1996 was that analogical modeling could not predict the processing time dif-

ferences between regular and irregular verbs, and between known and unknown verbs. In reply, Chandler and Skousen (1997) noted that in Section 16.1 of *Analogy and Structure* (under “Efficiency and Processing Time”), the correct processing times were in fact predicted.

Prasada and Pinker (1993) report on a study in which English speakers produced past tense forms for various nonce verbs. They found that a subject’s willingness to provide irregular past-tense forms was strongly related to the nonce verb’s phonological similarity to existing irregular verbs, but for nonce verbs similar to existing regular verbs, no such correlation was found. Prasada and Pinker took this basic difference in behavior as evidence that English speakers use a dual-route approach in forming the past-tense, especially since a single-route connectionist approach failed to predict the basic difference in behavior. But more recently, Eddington (2000a) has shown that just because a particular implementation of connectionism fails to make the right prediction does not mean that the single-route approach is wrong. To the contrary, both analogical modeling and Daelemans’ instance-based approach (each a single-route approach to describing English past-tense formation) correctly predict Prasada and Pinker’s experimental findings.

An important application of analogical modeling is found in Jones 1996. Here we see analogical modeling applied to automatic translation (between English and Spanish as well as English and Japanese). Most work done in analogical modeling has dealt with phonology, morphology, and orthography (the linguistic disciplines most closely connected to an objective reality), but here Jones shows how analogical modeling can be applied to syntax and semantics. He contrasts analogical modeling with both traditional rule approaches and connectionism (parallel distributed processing). In a variety of test cases, he finds analogical modeling more successful and less arbitrary than parallel distributed processing.

There have also been a number of applications to several non-English language problems in, for instance, the work of Eddington (Spanish stress assignment) and Douglas Wulf (German plural formation). Eddington’s work on Spanish (Eddington 2000b) has shown that analogical modeling can correctly predict stress placement for about 95% of the words, but in addition can regularly predict the stress for nonce words from experiments and for errors that children make. Wulf (1996) has found that analogical modeling is able to predict cases where an umlauting plural type has been extended from a frequent exceptional German plural to other less frequent words.

Daelemans, Gillis, and Durieux (1997) have done considerable work comparing analogical modeling with various instance-based approaches to language. They have discovered that under regular conditions, analogical modeling consistently outperforms their own instance-based approaches in predicting Dutch stress (see their Table 1.3). Only when they add various levels of noise to the system are they

able to get comparable results for analogical modeling and their instance-based approaches (see their Table 1.4), but their introduction of noise appears irrelevant to the larger issue of which approach best predicts Dutch stress.

Skousen's work on the Finnish past-tense has been able to capture the otherwise unaccountable behavior of certain verbs in Finnish. Of particular importance is his demonstration (Skousen 1995:223–226) that the verb *sorta*- 'oppress', under an analogical approach, takes the past-tense form *sorti*. According to every rule analysis found in the literature, verbs stems ending in *-rta* or *-rtä* should take *-si* in the past-tense. Yet speakers prefer *sorti*, not *sorsi* (although *sorsi* does occasionally occur). When we look at the analogical set for *sorta*- (a relatively infrequent verb), we discover that for this example only, verbs containing *o* as the first vowel (24 of them) almost completely overwhelm verbs ending in *-rta* or *-rtä* (only 5 of these). And each of these verbs with *o* produce the past-tense by replacing the final stem vowel *a* by *i* (thus giving *sorti*). This large group of *o*-vowel verbs just happens (from an historical point of view) to take this same outcome. Although there is another group of verbs that take the *si* outcome, its effect is minor. The resulting probability of analogically predicting *sorti* is 94.6%.

More generally, a correct theory of language behavior needs to pass certain empirical tests (Skousen 1989:54–76). In cases of categorical behavior (such as the indefinite article *a/an* in English), there should be some leakage (or fuzziness) across categorical boundaries (such as *an* being replaced by *a*). Similarly, when we have a case of exceptional behavior in a field of regular behavior (such as the plural *oxen* in English), we should find that only when a given context gets very close to the exceptional item do we get a small probability of the given context behaving like the exception (such as the infrequent plurals *axen* for *ax* and *uxen* for the nonce *ux*). And finally, in empty space between two occurrences of different behavior, we should get transitional behavior as we move from one occurrence to the other.

A theory of language behavior is tested by considering what kinds of language changes it predicts. The ability to simply reproduce the outcomes for the occurrences in the dataset does not properly test a theory. Instead, we try to predict the outcome for given contexts that are not in the dataset, and then we check these predictions against the kinds of changes that have been observed, preferably changes that have been naturally observed. Such data for testing a theory can be found in children's language, historical change, dialect development, and performance errors. Experiments (involving for instance, nonce items) can be helpful if their results do not inordinately clash with naturally observed changes, but in general, artificial experiments always run the risk of contaminated results. Experiments can help us gather additional data, providing their results do not sharply contradict observations from actual language use.

## 5. Local versus global significance of variables

This explicit theory of analogical modeling differs considerably from traditional uses of analogy in language description. First of all, traditional analogy is definitely not explicit. Related to this problem is that almost any item can serve as the analogy for predicting behavior, although in practice the attempt is to always look to nearest neighbors for the preferred analogical source. But if this fails, one can almost always find some item, perhaps considerably different, that can be used to analogically predict the desired outcome. In other words, if needed, virtually any occurrence can serve as the analogical source.

Skousen's analogical modeling, on the other hand, will allow occurrences further away from the given context to be used as the exemplar, but not just any occurrence. Instead, the occurrence must be in a homogeneous supracontext. The analogical source does not have to be a near neighbor. The probability of an occurrence further away acting as the analogical model is nearly always less than a closer occurrence, but this probability is never zero (providing the occurrence is in a homogeneous supracontext).

Further, the ability to use all the occurrences in all the homogeneous supracontexts of the contextual space directly accounts for the gang effects we find when we describe either categorical or regular/exceptional behavior. In other words, we are able to predict "rule-governed" behavior (plus a little fuzziness) whenever the data behaves "regularly".

Analogical modeling does not require us to determine in advance which variables are significant and the degree to which these variables determine the outcome (either alone or in various combinations). Nearest-neighbor approaches are like traditional analogical practice in that they try to predict behavior by using the most similar occurrences to the given context. But unless some additional information is added, the leakage across categorical boundaries and in regions close to exceptions will be too large. As a result, nearest-neighbor approaches frequently try to correct for this excessive fuzziness by ranking the statistical significance of each variable. One can determine, as Daelemans, Gillis and Durieux have (1994:435–436), the information gain (or other measures of reducing entropy) for each variable. Such added information requires a training period to determine this information, and in this regard is like connectionism.

Analogical modeling, on the other hand, does not have a training stage except in the sense that one must have a dataset of occurrences. Predictions are made "on the fly", and all variables are considered apriorily equal (with certain limitations due to restrictions on short-term memory). The significance of a variable is determined locally – that is, only with regard to the given context. The extent of any gang effect is determined by the location of the given context and the amount of resulting homogeneity within the surrounding contextual space.



## References

- Aha, David W., Dennis Kibler, & Marc K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Chandler, Steve (1995). Non-declarative linguistics: Some neuropsychological perspectives. *Rivista di Linguistica*, 7, 233–247.
- Chandler, Steve, & Royal Skousen (1997). Analogical modeling and the English past tense: A reply to Jaeger et al. 1996. <<http://humanities.byu.edu/am/>>.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1994). The acquisition of stress: A data-oriented approach. *Computational Linguistics*, 20, 421–451.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1997). Skousen's analogical modeling algorithm: A comparison with lazy learning. In D. Jones & H. Somers (Eds.), *New Methods in Language Processing* (pp. 3–15). London: University College Press.
- Derwing, Bruce, & Royal Skousen (1994). Productivity and the English past tense: Testing Skousen's analogy model. In S. D. Lima, R. L. Corrigan, & G. K. Iverson, *The Reality of Linguistic Rules* (pp. 193–218). Amsterdam: John Benjamins.
- Eddington, David (2000a). Analogy and the dual-route model of morphology. *Lingua*, 110, 281–298.
- Eddington, David (2000b). Spanish stress assignment within analogical modeling of language. *Language*, 76, 92–109.
- Eggington, William G. (1995). Analogical modeling: A new horizon. *Rivista di Linguistica*, 7, 211–212.
- Jaeger, Jeri J., Alan H. Lockwood, David L. Kemmerer, Robert D. Van Valin Jr., Brian W. Murphy, & Hanif G. Khalak (1996). A positron emission tomographic study of regular and irregular verb morphology in English. *Language*, 72, 451–497.
- Jones, Daniel (1996). *Analogical natural language processing*. London: University College Press.
- McClelland, James L., David E. Rumelhart, & the PDP Research Group (1986). *Parallel distributed processing* (PDP), 2 volumes. Cambridge, MA: MIT Press.
- Prasada, Sandeep, & Steven Pinker (1993). Generalization of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8, 1–56.
- Robinson, Derek (1995). Index and analogy: A footnote to the theory of signs. *Rivista di Linguistica*, 7, 249–272.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Press.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Press.
- Skousen, Royal (1995). Analogy: A non-rule alternative to neural networks. *Rivista di Linguistica*, 7, 213–231.
- Skousen, Royal (1998). Natural statistics in language modeling. *Journal of Quantitative Linguistics*, 5, 246–255.
- Wulf, Doug (1996). An analogical approach to plural formation in German. In *Proceedings of the Twelfth Northwest Linguistics Conference. Working Papers in Linguistics*, 14 (pp. 239–254). Seattle: University of Washington.

## CHAPTER 2

# Issues in Analogical Modeling

Royal Skousen

### Homogeneity and Analogical Modeling

Analogical modeling is an exemplar-based system that usually includes the nearest neighbors in the analogical set, but typically the analogical set will also have exemplars that are not particularly near to the given context. One may ask whether there is much difference in prediction if we avoid looking for more distant examples. The tremendous advantage of ignoring homogeneity would be that we would no longer need to test the exponentially increasing number of supracontexts for homogeneity. In other words, do we really need to have homogeneity in analogical modeling?

Let us consider the analogical sets for a number of Finnish verbs for which the past tense is predicted from the present stem (as discussed in Chapter 5 of *Analogical Modeling of Language*, Skousen 1989: 101–136). In the examples we will be considering, there are three possible outcomes, represented as *V-i*, *a-oi*, and *tV-si*. In each case, we give the full analogical set, but then we identify the nearest neighbors and redo the prediction in terms of only these nearest neighbors.

In certain cases, we get the same basic prediction, whether or not homogeneity is invoked. For instance, if the nearest neighbors all have the same behavior, we get the same virtual prediction with or without homogeneity. For the relatively rare verb *raata-* ‘to toil’, we can theoretically get three different past-tense forms: *raati*, *raatoi*, *raasi* (corresponding respectively to the outcomes *V-i*, *a-oi*, and *tV-si*). But the nearest neighbors (*kaata-* ‘to overturn’ and *raasta-* ‘to grate’, each marked with an asterisk in the following listing) have the *a-oi* outcome, so the nearest-neighbor prediction is *a-oi* 100% of the time. On the other hand, given the full analogical set, *a-oi* is predicted 99.6% of the time. These predictions agree with the behavior of Finnish speakers, who consistently give *raatoi* as the past-tense form for *raata-*.

## GIVEN CONTEXT

RAVA0=0=TA *raata-* 'to toil'

## ANALOGICAL SET

outcome	variables	verb	pointers	%
a-oi	HAVA0=OSTA	haasta-	160	17.9
a-oi	KAVA0=0=TA	*kaata-	192	21.5
a-oi	PAVA0=OHTA	paahta-	160	17.9
a-oi	RAVA0=OSTA	*raasta-	216	24.2
tV-si	RIVESN0=Ta	rientä-	4	0.4
a-oi	SAVA0=OTTA	saatta-	160	17.9

## STATISTICAL SUMMARY

outcome	pointers	%
a-oi	888	99.6
tV-si	4	0.4
total frequency = 892 pointers		

On the other hand, if the nearest neighbors behave differently, then we get competition. For instance, for the infrequent verb *saarta-* 'to surround', there are two nearest neighbors, *saatta-* 'to accompany' and *siirtä-* 'to move'. The first one takes the *a-oi* outcome (namely, *saattoi* for the past tense), the second takes *tV-si* (thus, *siirsi* for the past tense). Analogical modeling, like a nearest neighbor approach, predicts both *saattoi* and *saarsi* fairly equally, although not exactly equally:

## GIVEN CONTEXT

SAVASR0=TA *saarta-* 'to surround'

## ANALOGICAL SET

outcome	variables	verb	pointers	%
a-oi	AA0=SL0=KA	alka-	16	1.2
a-oi	AA0=SN0=TA	anta-	28	2.1
a-oi	HAVA0=OSTA	haasta-	80	5.9
tV-si	HUVOSL0=TA	huolta-	56	4.1
a-oi	KAVA0=0=TA	kaata-	96	7.0
a-oi	KA0=SN0=TA	kanta-	28	2.1
a-oi	KA0=SROTTA	kartta-	26	1.9
tV-si	KIVESL0=Ta	kieltä-	48	3.5
tV-si	KIVESR0=Ta	kiertä-	90	6.6
tV-si	KUVUSL0=TA	kuulta-	56	4.1
tV-si	KaVaSN0=Ta	kääntä-	48	3.5

a-oi	MA0=SLOTTA	maltta-	18	1.3
tV-si	MU0=SR0=TA	murta-	12	0.9
tV-si	MYVoSN0=Ta	myöntä-	48	3.5
a-oi	PAVA0=OHTA	paahta-	80	5.9
tV-si	PIVISR0=Ta	piirtä-	90	6.6
tV-si	PYVoSR0=Ta	pyörtä-	90	6.6
a-oi	RAVA0=OSTA	raasta-	80	5.9
tV-si	RIVESN0=Ta	rientä-	48	3.5
a-oi	SAVA0=OTTA	*saatta-	108	7.9
a-oi	SA0=0=0=TA	sata-	12	0.9
tV-si	SIVISR0=Ta	*siirtä-	108	7.9
V-i	SOVU0=0=TA	souta-	2	0.1
tV-si	TYVoSN0=Ta	työntä-	48	3.5
tV-si	VaVaSN0=Ta	vääntä-	48	3.5

## STATISTICAL SUMMARY

outcome	pointers	%
V-i	2	0.1
a-oi	572	41.9
tV-si	790	57.9

total frequency = 1364 pointers

These predictions agree with the intuitions of native speakers. (See, for instance, under *saartaa* in the *Nyky-suomen Sanakirja* [Dictionary of Modern Finnish] (Sademiemi 1973), where both the past-tense forms *saartoi* and *saarsi* are listed as equally possible.)

A similar example occurs with the infrequent verb *kaarta-* 'to swerve'. Again there are two nearest neighbors competing with one another (*kaata-* 'to overturn' and *kiertä-* 'to wind'). Both analogical modeling and the nearest neighbor approach basically predict *kaartoi* and *kaarsi* as equally possible, again in agreement with speakers' predictions:

## GIVEN CONTEXT

KAVASR0=TA *kaarta-* 'to swerve'

## ANALOGICAL SET

outcome	variables	verb	pointers	%
a-oi	AA0=SL0=KA	alka-	16	1.0
a-oi	AA0=SN0=TA	anta-	28	1.7
a-oi	HAVA0=OSTA	haasta-	80	4.8
tV-si	HUVOSL0=TA	huolta-	56	3.4

a-oi	KAVA0=0=TA	*kaata-	178	10.7
a-oi	KAVI0=OHTA	kaihta-	34	2.1
a-oi	KAVI0=0=VA	kaiva-	30	1.8
a-oi	KA0=SN0=TA	kanta-	74	4.5
a-oi	KA0=SR0TTA	kartta-	70	4.2
a-oi	KA0=0=OSVA	kasva-	14	0.8
a-oi	KA0=0=OTTA	katta-	24	1.4
tV-si	KIVESL0=Ta	kieltä-	64	3.9
tV-si	KIVESR0=Ta	*kiertä-	118	7.1
tV-si	KUVUSL0=TA	kuulta-	76	4.6
tV-si	KaVaSN0=Ta	kääntä-	64	3.9
a-oi	MA0=SLOTTA	maltta-	18	1.1
tV-si	MU0=SR0=TA	murta-	12	0.7
tV-si	MYVoSN0=Ta	myöntä-	48	2.9
a-oi	PAVA0=OHTA	paahta-	80	4.8
tV-si	PIVISR0=Ta	piirtä-	90	5.4
tV-si	PYVoSR0=Ta	pyörtä-	90	5.4
a-oi	RAVA0=OSTA	raasta-	80	4.8
tV-si	RIVESN0=Ta	rientä-	48	2.9
a-oi	SAVA0=OTTA	saatta-	80	4.8
tV-si	SIVISR0=Ta	siirtä-	90	5.4
tV-si	TYVoSN0=Ta	työntä-	48	2.9
tV-si	VaVaSN0=Ta	vääntä-	48	2.9

## STATISTICAL SUMMARY

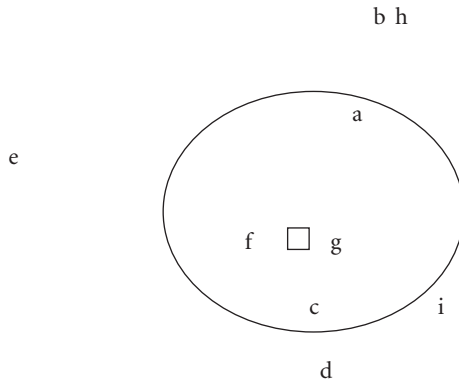
outcome	pointers	%
a-oi	806	48.6
tV-si	852	51.4

total frequency = 1658 pointers

From these examples it might seem reasonable to dispense with homogeneity as a necessary condition for predicting the behavior of a given context.

In its most primitive form, the nearest neighbor approach can be thought of as some kind of identification or recognition test. For each given context, we would first search for that given context in the dataset. If not found, we then assume that there is some error in the given context and thus look for an item in the dataset that most reasonably could be considered a mistaken variant of the given context. Thus *raata-* could be considered an error for either *raasta-* or *kaata-* (similarly, *saarta-* could be a mistake for *saatta-* or *siirtä-*).

In analogical modeling, this kind of simple recognition task is equivalent to treating each occurrence in the dataset as having its own distinguishing outcome. The result would be that every distinct occurrence in the dataset would have a different outcome. This would prevent any substantive use of homogeneity in predicting the outcome. No groups of distinct occurrences would ever be able to work together to produce a gang effect. The only occurrences in the analogical set would therefore be unobstructed. We can see this result in the following schematic, where the lower-case letters stand for non-repeating outcomes:



The square  $\square$  stands for the given context. The only unobstructed occurrences (encircled in the schematic) are *a*, *f*, *g*, and *c*. These occurrences include the nearest neighbors, plus any other neighbors with an unobstructed path from the given context. All other occurrences are further away (*b*, *h*, *e*, *i*, and *d*) and are obstructed, and are therefore prevented from being used as exemplars.

Nonetheless, neither this procedure of perceptual identification nor using only the nearest neighbors will always work. Consider the past-tense form for the infrequent Finnish verb *sorta-* ‘to oppress’. In this example, the nearest neighbor to *sorta-* is the verb *murta-* ‘to break’. This verb takes the *tV-si* outcome and thus predicts the *tV-si* outcome for *sorta-* (that is, *sorsiti*). However, speakers prefer the *V-i* outcome for *sorta-* (that is, *sortiti*). Interestingly, the analogical set for *sorta-* definitely predicts the *V-i* outcome, despite the fact that *murta-* (marked with an asterisk) is the nearest neighbor:

## GIVEN CONTEXT

SO0=SR0=TA *sorta-* 'to oppress'

## ANALOGICAL SET

outcome	variables	verb	pointers	%
V-i	HO0=0=OHTA	hohta-	111	5.1
V-i	HOVI0=0=TA	hoita-	94	4.3
tV-si	HUVOSL0=TA	huolta-	3	0.1
V-i	JO0=0=OHTA	johta-	111	5.1
V-i	JOVU0=OSTA	jousta-	76	3.5
tV-si	KIVSR0=Ta	kiertä-	20	0.9
V-i	KOVI0=OTTA	koitta-	76	3.5
V-i	KO0=0=0=KE	koke-	49	2.2
V-i	KO0=0=OSKE	koske-	37	1.7
V-i	KO0=0=OSTA	kosta-	111	5.1
tV-si	KUVUSL0=TA	kuulta-	3	0.1
V-i	LOVI0=OSTA	loista-	76	3.5
tV-si	MU0=SR0=TA	*murta-	37	1.7
V-i	NO0=0=OSTA	nosta-	111	5.1
V-i	NOVU0=0=SE	nouse-	32	1.5
V-i	NOVU0=0=TA	nouta-	94	4.3
V-i	OO0=0=0=LE	ole-	49	2.2
V-i	OO0=0=OSTA	osta-	111	5.1
V-i	OO0=0=OTTA	otta-	111	5.1
tV-si	PIVISR0=Ta	piirtä-	20	0.9
V-i	POVI0=OSTA	poista-	76	3.5
V-i	PO0=SL0=KE	polke-	55	2.5
V-i	PO0=SL0=OTTA	poltta-	121	5.6
V-i	PO0=0=0=TE	pote-	80	3.7
tV-si	PYVoSR0=Ta	pyörtä-	20	0.9
tV-si	SIVISR0=Ta	siirtä-	26	1.2
V-i	SOVI0=OTTA	soitta-	92	4.2
V-i	SO0=0=OTKE	sotke-	46	2.1
V-i	SOVU0=0=TA	souta-	117	5.4
V-i	SUVI0=OSTA	suista-	7	0.3
V-i	SU0=0=0=LA	sula-	17	0.8
V-i	SU0=SL0=KE	sulke-	13	0.6
V-i	SU0=0=0=RE	sure-	9	0.4
V-i	Sa0=SR0=KE	särke-	17	0.8

V-i	TOVI0=OSTA	toista-	76	3.5
V-i	VOVI0=OTTA	voitta-	76	3.5

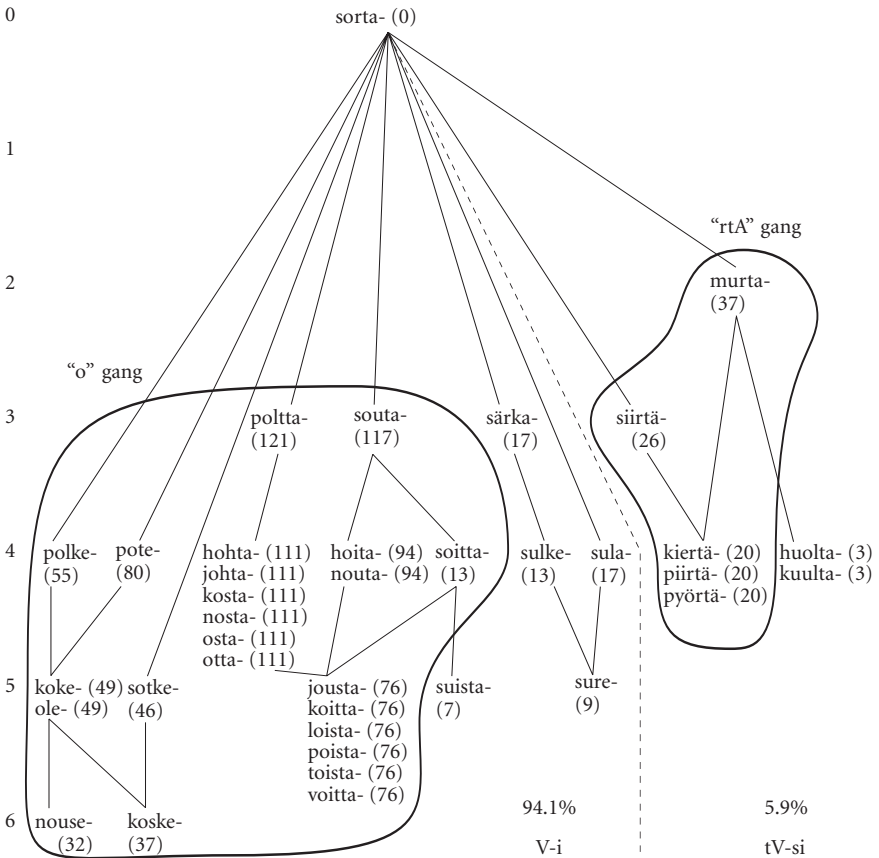
STATISTICAL SUMMARY

outcome pointers %

V-i	2051	94.1
tV-si	129	5.9

total frequency = 2180 pointers

When we look at the entire analogical set organized according to distance from the given context *sorta-*, we discover that there is a huge gang of *o* verbs that take the *V-i* outcome. Here distance refers to the number of variables in disagreement between *sorta-* (the given context) and any particular verb in the dataset.





Each of the 24 verbs in this large homogeneous space have *o* as the main vowel (*polttä-* ‘to burn’, *souta-* ‘to row’, *osta-* ‘to buy’, *hoita-* ‘to take care of’, and so on). There is, to be sure, a competing but smaller gang of five verbs that end in *rta* or *rtä* (including the nearest neighbor *murta-* ‘to kill’). But the much larger *o* gang overwhelms the *rta/rtä* gang. Because of the strong gang effect for the *o* vowel, there are many more pointers to any particular verb in the *o* gang than to virtually any other verb elsewhere in the analogical set. For instance, the nearest neighbor *murta-* (at a distance of only two from the given context, *sorta-*) is found in the weak *rta/rtä* gang and is accessed by 37 pointers, the same number of pointers pointing to *koske-* ‘to touch’ (which is at a much further distance of six from *sorta-* and is at the furthest reaches of the *o* gang).

### Local versus global significance of variables

Analogical modeling of the Finnish past tense has been able to capture the otherwise unexplained behavior of certain verbs in Finnish. Of particular importance is the verb *sorta-* ‘to oppress’, which takes the past-tense form *sorti*. According to every rule analysis found in the literature, verbs stems ending in *-rta* or *-rtä* should take *-si* in the past tense. Yet speakers prefer *sorti*, not *sorsi*.

In the original dataset of Finnish verbs, there are 24 verbs that have the *o* vowel. In *Analogical Modeling of Language*, Skousen (1989: 114–124) correctly predicted the regularity of the past tense for all 24 of these verbs as well as for two infrequent verbs with the *o* vowel – namely, *sorta-* ‘to oppress’ and *jouta-* ‘to have time’. By examining the analogical sets for these 26 verbs, we can determine what fraction of the verbs in each analogical set is made up of *o* verbs:

1. for 20 of these *o* verbs, only 24 out of 117 different verbs in the analogical set are *o* verbs;
2. in one case (*pote-* ‘to be sick’), we get 24 out of 70 verbs;
3. in only five cases do we get a dominance of the *o* verbs:

<i>hoita-</i> ‘to tend’	24 out of 27 verbs
* <i>jouta-</i> ‘to have time’	24 out of 27 verbs
<i>nouta-</i> ‘to fetch’	24 out of 30 verbs
<i>souta-</i> ‘to row’	24 out of 34 verbs
* <i>sorta-</i> ‘to oppress’	24 out of 36 verbs

Two of the last five verbs (*jouta-* and *sorta-*, each marked with an asterisk) are not in the dataset. The three occurring verbs are not determined solely by the *o* vowel since they each have six variables in common:

Variable 2	o	the first vowel is <i>o</i>
Variable 3	V	the initial vowel is long
Variable 5	Ø	the vowel is not followed by a syllable-final sonorant
Variable 7	Ø	the vowel is not followed by a syllable-final obstruent
Variable 9	t	the second syllable begins with a <i>t</i>
Variable 10	a	the stem ends with an <i>a</i> vowel

So for three verbs in the dataset (*hoita-*, *nouta-*, and *souta-*), we have a cluster of six co-occurring variables that predicts that the final stem vowel will be replaced by *i* to form the past tense. In other words, Variable 2 alone (the one that represents the *o* vowel) does not predict the outcome. So in trying to predict the past tense for the only three verbs where the *o* variable might prove significant, this variable cannot be separated out from five other variables.

The *o* vowel is an good example of a locally significant variable in the Finnish past-tense database. In predicting the past-tense form for all verbs except one, this *o* variable is not crucial, no matter how frequent the verb is. It only turns out to be crucial for the relatively infrequent verb *sorta-* ‘to oppress’, a verb which is not in the dataset. In other words, the crucialness of this variable for *sorta-* cannot be learned from predicting the past tense of other verbs. This variable only becomes crucial when the analogical system is asked to predict the past tense for *sorta-*. In an analogical approach, the significance of the *o* variable is locally determined, not globally. The occurrences in the dataset carry the information necessary to make predictions, but the significance of a particular variable cannot be determined independently of the occurrences themselves. Nearest-neighbor approaches, when they rely on measuring information gain, can never obtain sufficient gain for this *o* vowel to be able to predict *sorti*. It seems that only a local approach will correctly predict *sorti* as the preferred past tense.

### Including or excluding “unimportant” variables

The *sorta-* example in Finnish implies that (1) virtually any variable can potentially affect linguistic behavior, and (2) a variable may be locally significant, but not globally. A variable (or combination of variables) is globally significant if we can directly determine the statistical significance of that variable (or combination of variables) from the occurrences in the dataset. Local significance, on the other hand, means that the statistical significance of a variable (or combination of variables) only shows up when trying to predict the outcome for a given context. And usually the given context has an unusual combination of variables, uncharacteristic of combinations found in the dataset itself.

The computer implementation of analogical modeling places a limit on the number of variables for specifying occurrences and given contexts. Because of the exponential explosion of analogical modeling, not every possible variable can be included. In Section 3.1 of *Analogical Modeling of Language* (Skousen 1989:51–54), the possibilities were restricted by using principles of proximity and lexical distinguishability, and by limiting phonetic representations to individual sounds and basic syllable structure. Nonetheless, a number of “unimportant” variables were always specified for each dataset, and this decision proved to be highly significant. If the *o* vowel had not been fully specified for the verbs in the Finnish dataset, the analogical prediction for *sorta-* would have been incorrect.

Theoretically, one could use analogical modeling to determine which variables are globally significant. But this would defeat the real purpose of analogical modeling, which is to predict actual language behavior. Consider, for instance, the indefinite article *a/an* in English. In the dataset discussed in Section 3.2 of *Analogical Modeling of Language* (Skousen 1989:54–59), there are 136 occurrences of *a*, each followed by a word beginning with a consonant, and 28 occurrences of *an*, each followed by a vowel-initial word. There are no exceptions in the dataset to the “standard” rule. If we analyzed the dataset globally, we would discover that the *a/an* outcome was entirely predictable in terms of a single variable – namely, the syllabic nature of the first sound in the following word. Given this result, we could argue for excluding every other variable from the dataset.

But if we did this, then severe difficulties in predicting actual language behavior would arise. First of all, we would be unable to predict the one-way leakage that we find in children’s language, adult errors, and dialect development – namely, the prevalent tendency to replace *an* by *a*, but not *a* by *an* (thus “a apple”, but not “an boy”). The only reason we are able to predict the appropriate fuzziness of *a/an* usage is because the dataset contains other variables besides the “correct” or “crucial” one. If only that one variable were given, then we would always “correctly” predict *an* for vowel-initial words and *a* for consonant-initial words, yet speakers do not behave this “correctly”.

A more serious problem would arise in cases requiring robustness. If the first segment of the following word were, for instance, obscured by noise, we would be unable to predict anything since our dataset would only have information about the first segment. Actual speakers can deal with defective input, which means that we must specify more than just the significant variables if we want speakers to deal with cases where the significant variables might be either missing or distorted.

## Imperfect memory and its effects on analogical predication

Daelemans, van den Bosch, and Zavrel (1999) argue that with nearest-neighbor approaches, predictions are worse if the data is “mined” in advance – that is, if variables are reduced and “bad” (or “exceptional”) examples are removed. Such systems tend to collapse or become degraded when memory losses occur. On the other hand, memory loss is important in analogical modeling, especially since imperfect memory results in statistically acceptable predictions (and reduces the extraordinary statistical power of the approach). For instance, randomly throwing out about half the data leads to standard statistical results (described in Skousen 1998). In analogical modeling, statistically significant results are retained under conditions of imperfect memory. In fact, a statistically significant result is one that holds when at least half the data is forgotten. The reason that analogical modeling can get away with substantial memory loss is because this approach considers much larger parts of the contextual space, whereas nearest-neighbor approaches tend to fail when memory is imperfect.

In analogical modeling, given sufficiently large amounts of data, stability sets in, with the result that adding more examples in the dataset will have little effect on predicting behavior. Imperfect memory also shows how less frequent exceptions tend to be removed from a language, but frequent exceptions are kept. This agrees with what Bloomfield observed many years ago about historical change (1933:408–410).

## System self-prediction versus dynamic language usage

One important task in analogical modeling is to define exactly what we are trying to predict. One common task used in exemplar-based systems is to determine how much of the dataset is self-predicting – that is, for each occurrence in the dataset, we determine how the rest of the dataset would predict the outcome for that occurrence. In analogical modeling, we would make the testset equal to the dataset, but we would exclude the given context for each item to be tested. For each tested occurrence in the dataset, we would then compare the predicted outcome with the actual outcome listed in the dataset. The overall percentage of correctness then could be interpreted as a measure of accuracy for the exemplar-based system being used. The goal, it would seem, in system self-predictability is to maximize the percentage of correctness.

Despite its attractiveness, self-predictability is misguided. In actual fact, all this approach is doing is measuring the degree of regularity within the dataset. In Section 3.3 of *Analogical Modeling of Language* (Skousen 1989:60–71), this very

method is discussed (and is referred to as “excluding the given context from the dataset”). The importance of this measure is that it describes how much of the system is regular (that is, predictable). If the predicted outcome agrees with the actual outcome, the occurrence is regular. Otherwise, it is exceptional (according to one of two definitions of exceptionality). For each case of disagreement, system self-predictability is simply stating that the outcome for this item must be remembered. In cases of agreement, the outcome can be forgotten since surrounding exemplars predict the same (correct) outcome.

For instance, suppose we are trying to predict the plural in English. If the correct plural for the noun *ax* is forgotten, the analogical system (even with a rather small number of the most frequent exemplars) will predict *axes*, which is the correct plural. On the other hand, if the plural for the noun *ox* is forgotten, the correct plural *oxen* will never be predicted because all surrounding exemplars take the regular plural ending. In this case, *axes* will be predicted rather than *oxen*. All this is, of course, obvious. But we do not penalize the analogical system because it misses in the case of *ox*. The reason it misses is because the standard language itself fails to use the regular *axes*. It is not reasonable to view the missing of *oxen* as somehow a failure in analogical modeling.

The real power of analogical modeling is that it predicts the appropriate kinds of fuzziness in the contextual space. For instance, if the plural for *ax* is forgotten, *axes* will be predicted most of the time, but not always. There is a small probability that *axen* will be predicted – and of course this leakage towards the *-en* plural is because the exceptional plural *oxen* for *ox* is near *ax*. The real advantage of analogical modeling is in its ability to predict this minor leakage. The word *ax* is surrounded by similarly-behaving regular plurals, and the resulting gang effect dilutes the influence of the isolated, exceptional *ox*. Even when we are close to an exceptional item, the regular behavior will still dominate.

So just giving a percentage of system self-prediction is not particularly insightful. What we are really interested in is whether or not errors in system self-prediction might actually reflect the kinds of errors children make while learning their native language, or how the language has evolved historically or dialectally. Perhaps we can find evidence for these errors in adult speech, in new words entering the language, or in experiments involving nonce items. The kind of evidence we are looking for should represent the dynamics and variation of actual language usage.

In the case of the Finnish past tense (discussed in Chapter 5 of *Analogical Modeling of Language*, Skousen 1989:101–136), system self-prediction was actually very high. But the interesting results were the cases where the system failed to self-predict correctly (see Section 5.6, Skousen 1989:119–124):

(1) *one infrequent archaic verb*

VERB	FREQUENCY	OUTCOME	
		PREDICTED	ACTUAL
<i>virikka</i> - ‘to utter’	2	<i>virkki</i>	<i>virkkoi</i>

This verb is of quite low frequency, although it did end up in the dataset. It is only used in archaic-sounding poetic expressions and is readily recognized as coming from the Finnish national epic, the Kalevala. Historically, the past tense is *virkkoi* (the outcome *a-oi*), but when the past-tense outcome is not remembered, the analogical system predicts *virkki*. This regularizing past-tense form (using the outcome *V-i*) is strongly favored (95.4%). In support of this prediction, we note that the Kalevala itself sometimes uses *virkki*, thus showing the analogical tendency to replace the historical *virkkoi* (for instance, line 102 of Poem 2; see Lönnrot 1964:8).

(2) *five highly frequent, exceptional verbs*

VERB	FREQUENCY	OUTCOME	
		PREDICTED	ACTUAL
<i>taita</i> - ‘to know how’	125	<i>taitoi</i>	<i>taisi</i>
<i>tietä</i> - ‘to know’	101	<i>tieti</i>	<i>tiesi</i>
<i>pyytä</i> - ‘to request’	63	<i>pyyti</i>	<i>pyysi</i>
<i>löytä</i> - ‘to find’	57	<i>löyti</i>	<i>löysi</i>
<i>huuta</i> - ‘to shout’	29	<i>huuti</i>	<i>huusi</i>

Dialectally, the last four verbs are found with the *V-i* outcome, thus the errors in system self-prediction are not really errors, but reflect actual language tendencies. Moreover, the role of frequency here shows the long-recognized tendency for exceptions in a language to be highly frequent. Low frequent exceptions are eliminated over time, which is precisely what the theory of analogical modeling predicts. In fact, in Section 5.5 of *Analogical Modeling of Language* (Skousen 1989:114–118), we discover that all the low frequency verbs (ones that did not make the dataset) are correctly predicted by the system, including the unexpectedly correct prediction of *sorti* for *sorta*- ‘to oppress’. And in Section 5.6 (which included frequent verbs in the language), it is only the most frequent verbs that are exceptional (Skousen 1989:119–124). Most of the verbs in the dataset itself are regular, which is what we historically expect.

## Variable weighting

Nearest neighbor approaches frequently add a weighting factor to each variable so that closeness to the given context (that is, similarity) is determined in terms of

the more significant variables. This kind of information is determined in advance of prediction. One motivation for this procedure is to avoid an exponential explosion of possibilities. It is assumed too difficult to determine the possible effect of each combination of variables in predicting the outcome. As a consequence, some function of the individual variables is proposed to predict the combined effect of a group of variables.

In analogical modeling, in principle at least, all variables are given the same weight. In determining the analogical set for a given context, homogeneity and the resulting gang effects ultimately account for the statistical significance of any particular combination of variables.

Some researchers in analogical modeling (such as David Eddington – see his article in this volume) have experimented with giving extra weight to particular variables (by repeating the variable a number of times in the data specification). This may be helpful from a heuristic point of view, although it cannot be correct in principle, at least for variables of the same type.

But for cases where the variable specification may involve completely different kinds of variables, the following question thus arises: Should phonological, semantic, social, and syntactic variables all be equally weighted with respect to each other? It seems quite appropriate that within the same kind of variable, weighting could be the same for each variable. Thus far in analogical modeling, there has been little mixing of classes. For instance, in *Analogical Modeling of Language*, the variables used to predict the Arabic terms of address (Section 4.5, Skousen 1989:97–100) are all social variables (gender, age, social class, and familiarity). In predicting the Finnish past tense (Chapter 5, Skousen 1989: 101–136), there are only phonological variables (specifying phonemes and basic syllabic structure). What would happen if we had a dataset with specifications involving both a social variable (such as gender) and a phonological variable (whether the initial segment is a vowel or a consonant)? Should both these variables from different classes be equally weighted? It seems unlikely that they would be, but how to compare their weight seems unclear. Following Eddington, one could use analogical modeling as a kind of discovery procedure to see which multiples of variable classes would predict the best results. Ultimately, such an approach seems problematic.

Another difficulty with weighting deals with the zeros that show up in the data representations. Some of these zeros are redundantly so. For instance, if we specify say a third syllable, we could represent the non-occurrence of that syllable by specifying the vowel as a zero. Surrounding consonants that could occur if the vowel existed would also be zeros, but redundantly. Typically, in analogical modeling, such redundant zeros have been represented by equal signs. In running the computer program, one can choose to ignore such redundant zeros, and in fact nearly all applications of analogical modeling thus far have followed this choice.

(Non-redundant zeros, on the other hand, have always been treated as regular variables.)

Recent research in analogical modeling has suggested there may be cases where redundant zeros should perhaps be treated the same as non-redundant zeros (that is, as actual variables). One place where this may be crucial is when we compare words with a differing number of syllables. Consider, for instance, the two Finnish verbs *kasta-* ‘to baptize’ and *tarkasta-* ‘to examine’. We first line up the two verbs from the end of each word:

= 0 = k a s t a  
t a r k a s t a

The difference between the two is that there is an antepenultimate syllable for the longer one, but none for the two-syllable one. In terms of variable specification, we can mark the shorter one as having a non-redundant zero for its antepenultimate vowel and two redundant zeros for the possible onset and coda for that syllable (that is, as ‘=0=kasta’). Now the question here is whether the difference between these two words is just one variable. If we ignore the redundant zeros (the equal signs), we only have one difference. But if we treat the redundant zeros as actual zeros, we get three differences:

difference of one	difference of three
= 0 = k a s t a	0 0 0 k a s t a
t a r k a s t a	t a r k a s t a

It seems quite reasonable that we should somehow count the whole syllable difference when comparing words with a differing number of syllables. This would mean that in making predictions, there would be more distance between such words, which seems more reasonable than always assigning a uniform difference of one.

We can see this difference quite clearly when we compare the verb *soitta-* ‘to ring’ with three longer verbs: *osoitta-* ‘to show’, *tasoitta-* ‘to level’, and *taksoitta-* ‘to assess’. If we ignore the redundant zeros, we consistently get a difference of one between *soitta-* and each of the three-syllable verbs. But if we count all the zeros, we get a sequence of verbs that move further and further away from *soitta-*, which seems intuitively correct:

given verb ( <i>soitta-</i> )	redundant zeros =0=soitta	no redundant zeros 000soitta
number of differences:		
<i>osoitta-</i> 0o0soitta	1	1
<i>tasoitta-</i> ta0soitta	1	2
<i>taksoitta-</i> taksoitta	1	3



These results suggest that in comparing words with a differing number of syllables, redundant zeros may need to be counted as real zeros.

### Effects of parameter specification

One important aspect of analogical modeling is that adjusting parameters and conditions doesn't make much difference in the resulting predictions. This is quite different from neural networks, where there are so many parameters and conditions to manipulate that almost any result can be obtained. One wonders if there is any limit to what can be described when so many possibilities are available. Lately, this same problem seems to be afflicting nearest neighbor systems, especially given all the different ways of measuring the global significance of each variable (and thus determining closeness).

Recent work in analogical modeling, on the other hand, suggests that in analogical modeling it is difficult to manipulate parameters to get different predictions. This is actually a desired result. Consider, for instance, whether random selection is done by choosing either an occurrence or a pointer (see Skousen 1992:8–9). The first choice provides a linear-based prediction, the second a quadratic one. Yet when either method is used in analogical modeling, we get the same basic results except that under linearity we get a minor increase in fuzziness at category boundaries and around exceptional occurrences.

We also get the same basic results when we consider the conditions under which a given outcome can be applied to a given context. This problem first arose in trying to predict the past tense for Finnish verbs. In *Analogical Modeling of Language* (Skousen 1989:101–104), the three possible past-tense outcomes were narrowly restricted by including a number of conditions:

- outcome *V-i*: replace the stem-final vowel by *i*  
*additional conditions*: none
- outcome *a-oi*: replace the stem-final *a* vowel by *oi*  
*additional conditions*: the first vowel is unround  
(*i*, *e*, or *a*)
- outcome *tV-si*: replace the sequence of *t* and the stem-final non-high  
unround vowel (*e*, *a*, or *ä*) by *si*  
*additional conditions*: the segment preceding the  
*t* is either a vowel or a sonorant (that is, not an  
obstruent)

Further, these added conditions had been assumed in all rule analyses of the Finnish past tense.

But these added conditions are not part of the actual alternation (which replaces one sound – or a sequence of sounds – by another). So one obvious extension of applicability would be to ignore these additional conditions and allow an outcome to apply only whenever a given verb stem meets the conditions specified by the actual alternation:

- outcome *V-i*: replace the stem-final vowel by *i*
- outcome *a-oi*: replace the stem-final *a* vowel by *oi*
- outcome *tV-si*: replace the sequence of *t* and the stem-final non-high unround vowel (*e, a, or ä*) by *si*

The argument for relaxing the conditions is that the analogical model itself should be able to figure out the additional conditions since they occur in the verbs listed in the dataset.

But one can even go further and let every outcome apply no matter what the stem ends in:

- outcome *V-i*: replace the stem-final vowel by *i*
- outcome *a-oi*: replace the stem-final vowel by *oi*
- outcome *tV-si*: replace the stem-final sequence of consonant and vowel by *si*

The argument here is that the analogical model itself should be able to figure out the alternation itself.

Applying these different conditions on outcome applicability, the results were essentially the same. The only difference in prediction (using selection by plurality) occurred in a handful of cases of nearly equal probability between competing outcomes.

In other words, analogical modeling doesn't provide many opportunities for varying parameters and conditions. We get the same basic results no matter whether we randomly select from the analogical set by occurrence or by pointer – and no matter what the degree to which we restrict the conditions on outcome applicability. The only real way to affect the results is in the dataset itself, by what occurrences we put in the dataset and how we specify the variables for those occurrences. And specifying the dataset is fundamentally a linguistic issue. Thus analogical modeling is a strong theory and is definitely risky. It is not easily salvaged if it substantially fails to predict the right behavior.

### Categorizing the outcomes

In predicting the past tense in English (see Derwing & Skousen 1994), the following issue comes up: Should the regular past tense be considered a single outcome or

three different outcomes. From a concrete point of view, we could argue that there are three different regular past-tense forms in English: *-d*, *-t*, and *-Ed* (where *E* stands for the schwa vowel). The distribution of these forms is determined by the final sound of the present-tense stem:

if the final sound is an alveolar stop (*t* or *d*), add *-Ed*:

*-Ed*    *paint*    *painted*

otherwise, if the final sound is voiceless (*p, f, θ, s, š, č, k*), add *-t*:

*-t*      *laugh*    *laughed*

otherwise, add *-d*:

*-d*      *use*        *used*

On the other hand, one could take an abstract approach and assume that there is only one past-tense form (namely *d*) and that the other two pronunciations are predicted phonologically: when preceded by an alveolar stop, schwa is inserted before the *d* suffix; otherwise, *d* is devoiced to *t* when preceded by a voiceless consonant.

A similar underlying assumption of categorization is involved when trying to predict three past-tense outcomes for Finnish: *V-i*, *tV-si*, and *a-oi*. Two of these outcomes involve the class symbol *V* (for vowel), yet in actual fact the morphological alternations themselves involve specific vowels, and not every possible vowel:

<i>V-i</i>	<i>e-i</i>	<i>tule-</i>	<i>tuli</i>
	<i>ä-i</i>	<i>pitä-</i>	<i>piti</i>
	<i>a-i</i>	<i>muista-</i>	<i>muisti</i>
<i>tV-si</i>	<i>te-si</i>	<i>tunte-</i>	<i>tunsi</i>
	<i>tä-si</i>	<i>tietä-</i>	<i>tiesi</i>
	<i>ta-si</i>	<i>taita-</i>	<i>täisi</i>
<i>a-oi</i>	<i>a-oi</i>	<i>autta-</i>	<i>auttoi</i>

So using specific vowels instead of *V*, we could break up *V-i* into three separate outcomes (*e-i*, *ä-i*, and *a-i*); similarly for *tV-si*, we would have *te-si*, *tä-si*, and *ta-si*. Using such a system with more outcomes, gang effects in the contextual space would be dramatically reduced because similar outcomes would now be considered different outcomes, thus reducing the amount of homogeneity in the contextual space.

These problems in categorization consistently show up in analogical modeling (and every theory of language, for that matter). Whenever we specify the outcomes for a dataset, we are making decisions about categories. If we combine a number of specific outcomes into a more general one, the chances are greatly increased

that our predictions will be less fuzzy and involve considerably stronger gang effects. One wonders then if there isn't some way we can let the analogical system itself determine the outcomes (which, after all, is a problem in categorization). This question is now beginning to be considered (see Christer Johansson's article in this volume).

## The exponential explosion

Analogical modeling, as is well-known, tests all combinations of variables. If there are  $n$  variables, we get  $2^n$  combinations. Basically, increasing the given context by one variable doubles the memory requirements as well as the running time (see Section 6.1 of *Analogical Modeling of Language*, Skousen 1989:137–139). This exponential explosion is much like the folk story about the peasant who got his prince to give him a penny on the first day of the month and agree to double the amount for each subsequent day of the month. After about half the month was over, the prince suddenly realized that his little agreement was soon going to bankrupt him. (On the last day of a 31-day month the prince would have to pay out  $2^{30} = 1,073,741,824$  pennies, and the total payment for the whole month would be  $2^{31} - 1 = 2,147,483,647$  pennies, over \$21 million, given a hundred pennies to the dollar.)

Our research group has tried several different approaches to dealing with the exponential explosion. One constant approach has been to fine-tune the computer program, based on the original Pascal program given in Appendix 3 of *Analogical Modeling of Language* (Skousen 1989:191–204). The original program could handle only about 12 variables, but more recent improvements (including a major rewriting of the program in Perl) allow us to run over 20 variables. Still, the exponential effects are there and are ultimately unavoidable.

A few years ago, our research group considered a revised algorithm that does not keep track of every possible combination of variables, but instead stores information about certain crucial heterogeneous supracontexts which define the boundary between homogeneity and heterogeneity in the multi-dimensional contextual space. Still, the exponential explosion occurred, sometimes extraordinarily so, especially if there were supracontexts involving non-deterministic behavior. But one helpful result was that the exponential explosion occurred only in time, not in memory.

This last result suggests that perhaps parallel processing might be applied to this revised algorithm in order to reduce the intractable exponential explosion to a tractable solution which could perform linearly in time and memory. In Section 6.1 of *Analogical Modeling of Language* (Skousen 1989:137–139), it was shown that

applying parallel processing to the original algorithm would reduce the program to a linear function in time, but the memory requirements would only be reduced by a factor on the order of  $1/\sqrt{n}$ . Since the revised algorithm already reduces the memory requirements to a linear order, parallel processing would only need to reduce the running time to linearity.

## Quantum Analogical Modeling

Since 1999, Skousen has been working on a completely different approach to dealing with the exponential explosion – namely, by re-interpreting analogical modeling in terms of quantum computing. Quantum computing operates on the principles of quantum mechanics and can, in theory, simultaneously keep track of an exponential number of states (such as  $2^n$  supracontexts) in terms of  $n$  quantum variables (called qubits, short for quantum bits). With such a quantum approach, we can potentially reduce intractable exponential problems to tractable ones. This has been demonstrated by Peter Shor’s 1994 discovery that quantum computing can reduce the running time for finding the two primes of a long integer (used in coding and decoding messages) from an intractable exponential processing time (when run on classical computers) to a tractable polynomial one – if only one had a quantum computer to run it on. Although there is no practical hardware implementation yet of a quantum computer, Shor showed in principle there was at least one important case of exponentiality that could be overcome using quantum simultaneity. This result strongly suggests that perhaps the solution to the exponential explosion in analogical modeling is not to avoid it or try to circumvent it, but rather to use the inherent parallelism of quantum computing to directly account for the exponentiality.

The evidence from language behavior continues to support the requirement that all possible combinations of an unlimited number of variables need to be handled, yet within linear time. Moreover, examples of local predictability (such as *sorta-* in Finnish) show that speakers do not determine in advance which combinations of variables are significant. These decisions are always made “on the fly” and for a specific given context. The exponential explosion is obvious in analogical modeling (and from the beginning has been recognized as inherent).

Other procedural approaches such as neural networks and nearest neighbor systems attempt to avoid the exponential explosion by trying to determine (often indirectly) the “most significant” variables and thus limit the number of possibilities that must be considered. For neural networks and the more sophisticated nearest neighbor approaches, the attempt to reduce exponentiality occurs in a “training stage”. This task is a global one and is inherently exponential since ultimately there

is no effective limit on the number of variables that must be considered and no principled way to account for any given combination of variables from acting in a statistically distinct way.

Most recently, working within a model of quantum computing, Skousen has been able to develop a quantum-based algorithm that deals directly with the exponential explosion and significantly re-interprets the basic approach to determining the analogical set and selecting an appropriate supracontext for predicting its behavior (see his article on quantum analogical modeling in this volume). The original description of analogical modeling discovered in the early 1980s (and described in *Analogy and Structure*, Skousen 1992, as well as *Analogical Modeling of Language*, Skousen 1989) was designed to account for language behavior, yet the fundamental equivalence between quantum computing and analogical modeling (unknown to Skousen at the time) was already present in the theory of analogical modeling and perhaps explains why the fundamental theory itself has remained unchanged since its first explication.

Theron Stanford, a member of the analogical modeling research group, has rewritten the basic analogical modeling program to take advantage of the classical subroutines and procedures in the quantum-computational algorithm. The new program, running (of course) on a classical, non-quantum computer, demonstrates substantial improvements in determining the analogical set, although the exponential explosion still shows up since the classical computer cannot directly account for quantum simultaneity.

Current work in procedural linguistic approaches has emphasized what one might call the “neurological temptation” – that is, the desire to remake every language problem into one involving neurons (or connections that have the appearance of neurons). Analogical modeling, on the other hand, argues for what one might call the “quantum temptation”.

## References

- Bloomfield, Leonard (1933). *Language*. New York: Holt, Rinehart & Winston.
- Daelemans, Walter, Antal van den Bosch, & Jakub Zavrel (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 34, 11–43.
- Derwing, Bruce, & Royal Skousen (1994). Productivity and the English past tense: Testing Skousen’s analogy model. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 193–218). Amsterdam: John Benjamins.
- Lönnrot, Elias (1964). *Kalvela*. Helsinki: Suomalaisen Kirjallisuuden Seura.
- Sadeniemi, Matti (Ed.). (1973). *Nykysuomen sanakirja*. Porvoo, Finland: Werner Söderström.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.

Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.

Skousen, Royal (1998). Natural statistics in language modeling. *Journal of Quantitative Linguistics*, 5, 246–255.

PART II

**Psycholinguistic evidence for  
Analogical Modeling**





## Skousen's analogical approach as an exemplar-based model of categorization

Steve Chandler

This paper continues the effort begun in three earlier papers (Chandler 1993, 1994, 1998) to locate Skousen's analogical model of language (AM) (Skousen 1989, 1992) in a larger theoretical and empirical framework within experimental cognitive psychology. Inasmuch as possible, I shall not repeat the contents of those earlier papers, except to correct some errors in them. Instead, I try here to expand upon those works by fixing Skousen's analogical model much more explicitly and precisely into a broader framework of category representation and associated cognitive processes. My goal is to show that the analogical approach provides us with a powerful, unifying framework for understanding how our brains construct and use "concepts" or "categories" from our memories for specific experiences, both linguistic and nonlinguistic.

The overall plan of this paper is first to describe how analogical modeling functions as a general theory of categorization, second to describe in some detail the contributions of that framework to our understanding of certain key cognitive processes and behaviors, and finally to conclude with an analogical modeling account for data said to argue strongly for a symbolic-rule based component for English past-tense verb morphology. This paper consists of a broad literature review and synthesis interspersed with brief simulations demonstrating the ability of the analogical approach to model data which have proven problematic for other approaches.

### 1. Preliminary considerations

Cognitive psychologists often single out categorization, or classification, as perhaps the most fundamental cognitive act. Over the past quarter century at least four major kinds of categorization theories (now essentially reduced to three) have competed with one another for widespread acceptance, each associated with its

own school of major proponents. Curiously, however, with at least two notable exceptions to be discussed below, two of those schools of categorization have come to be studied, tested, and debated most intensely within the field of linguistics, while the other two have competed almost exclusively within cognitive psychology. I try to illustrate this curious split in Figure 1.

Admittedly, Figure 1 greatly oversimplifies the situation in both linguistics and psychology. For each of the four types of models represented in Figure 1, there are numerous variations not shown, and for virtually every possible pairing of the four classes of models shown there, there are hybrid models which have been proposed, tested, and discussed. However, following Broadbent's (1987) suggestion, I find it more useful heuristically here to compare classes of theories sharing certain key defining characteristics than to become enmeshed in the details distinguishing among the alternatives within one class of theories. The one exception to this is my more detailed comparison of Skousen's AM with other exemplar-based theories. Thus, Figure 1 serves more as a schematic representation for the points I want to make about the study of categorization in linguistics and psychology than it does an effort to characterize the full range and variety of theories.

Perhaps the single most curious fact about the two academic disciplines represented in Figure 1 is how rigidly they partition into the two disciplines the types of cognitive models that they are willing to consider seriously. The bottom line in the figure represents the two major exceptions to this alignment of disciplines and theoretical frameworks, which I discuss below. Thus, within linguistics Steven Pinker and his colleagues couch the debate almost exclusively in terms of symbolic-rule systems versus connectionist models (e.g., Pinker & Prince 1988; Prasada & Pinker 1993). On the few occasions that proponents of the symbolic-rule school do allude to an exemplar-based model such as Skousen's AM, they lump it together with connectionist models and dismiss it out of hand with the same arguments used against the connectionist models. The connectionist linguists, on the other hand, virtually never consider exemplar-based alternatives to their own approaches either. I do not know why connectionist linguists do not consider exemplar-based models as alternatives to their own approaches, but I surmise that it is at least in part because they have accepted McClelland and Rumelhart's claim (1986a: 199ff.) that their PDP connectionist models can account for the instance-based effects said to motivate exemplar-based models as an alternative model of categorization.

What makes the attitude of the linguists toward instance-based models particularly curious, indeed even ironic, is the situation represented on the other side of Figure 1. For almost 30 years now, one of the core debates in psychology has been exactly the issue of whether exemplar-based approaches (or instance-based, I will use the terms interchangeably)<sup>1</sup> or schema-abstraction models of prototype theory (including both rule schemas and connectionist representations) better account for the experimental data on concept learning and categorization. The proponents

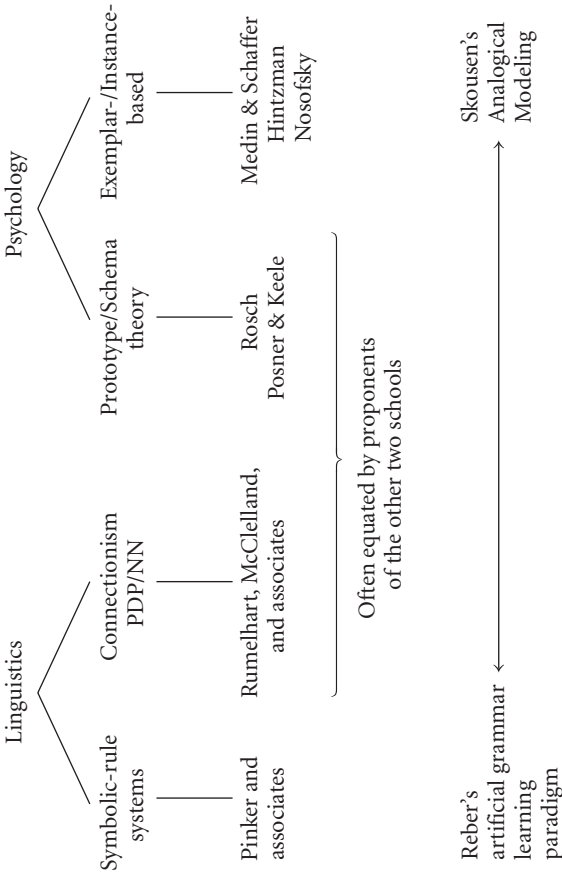


Figure 1.

of these two schools of thought certainly see important theoretical and empirical differences between prototype models and exemplar-based models of cognitive processes. I have described elsewhere why highly respected observers of the debate (e.g., Hintzman 1990; Ashby 1992; Estes 1994; Shanks & St. John 1994; Shanks 1995) all conclude that the exemplar-based models are better supported empirically by the experimental data than are the rule or schema abstraction models – better supported but not unequivocally supported by the data. Shanks (1995), for example, despite an apparent preference for exemplar-based models, cites evidence from artificial grammar learning (AGL) studies which he, and others, see as implicating some sort of rule or schema abstraction process, possibly as part of a dual representational system analogous to the dual system of verb morphology representation posited by Pinker and his associates. Because of its similarity to the linguistic debate and indeed because it has been cited in support of that debate (cf. Ullman, Corkin, Coppola, Hickock, Growdon, Koroschetz, & Pinker 1997), I will examine this issue in some detail in Section 3. Nevertheless, Estes (1994: 51) characterized as “curious” the fact that the prototype abstraction models, now generally embodied in connectionist form in the cognitive psychology literature, continue to be by far the “most visible” variety of categorization theory in the literature although “exemplar-similarity models” have been much more successful empirically over the previous decade.

With the possible exception of certain researchers within the AGL paradigm, whose work I discuss in Section 3.1, cognitive psychologists today do not generally consider explicitly the kinds of symbolic-rule systems assumed by the linguists on the left side of Figure 1. Apparently, most cognitive psychologists see such rule-based systems as having gone the way of the information processing models inspired by computer programming analogies and popular in the 1970s (e.g., Klatzky 1975; Miller & Johnson-Laird 1976; Lindsay & Norman 1977). In the discussion which follows, we will see incidental references to rule-based cognition, but what those researchers call rule-based behavior is nothing like the unconscious grammatical systems posited by most theoretical linguists. Instead, the psychologists appear to mean the strategic, conscious application of algorithmic-like procedures to the solving of some specific problem. Although I will not consider this kind of rule-based processing further, it is worth noting that there are models based on experimental findings which suggest that even such rule-based operations may rely ultimately on the step-wise application of exemplar-based knowledge in order to operate and may be replaced by exemplar-based data-bases in memory as people become more experienced at solving the relevant problems (Logan 1988; Nosofsky & Palmeri 1997).

Another curious fact represented in Figure 1 – and important for the discussion that follows – is that both many linguists and many psychologists lump connectionist models and prototype models (à la Posner & Keele 1968; Rosch 1973)

into the same theoretical category. What is curious about this is that there are both other linguists and other psychologists who identify closely with connectionist approaches to modeling language and who insist that connectionist models are not prototype models, but as is so often the case in such disagreements, the difference appears to lie mostly in the different meanings different researchers assign to the term *prototype*. Estes (1994) identified at least four significantly different working definitions of “prototype” commonly used in the cognitive psychology literature:

1. a cover term for the “prototype effects” described by Rosch (1973) and others as summarized below;
2. a hypothesized mental representation summarized and schematicized from perceptual experiences (e.g., Posner & Keele 1968);
3. the “central tendency” of a category of exemplars; it may or may not have its own structural existence in the brain; it could be just an “effect” of unknown origin, a prototype effect;
4. a “focal member”, “a highly representative exemplar of a category” which can stand for the entire category as a default interpretation.

In all fairness, it probably is not appropriate to equate the most familiar connectionist models with Definition (4). Such connectionist models do not normally store or retain exemplars (that is, individual representations of input stimuli), McClelland and Rumelhart's apparent claim (1986a: 199ff.) to the contrary notwithstanding (see Section 2.2.1). Definition (2) also may not be a fair characterization of connectionist models. It depends on what one means by “mental representation” of a category. If it is a unique, or mostly unique, ensemble of neurons dedicated to representing a particular concept, then most connectionist models clearly are not prototype models of that sort. On the other hand, if one takes the pattern of weighted connections among units of a trained connectionist network to be in some sense equivalent to our mental representations of experiences, then such models are fairly called prototype models. Finally, Definitions (1) and (3), central tendency representations that give rise to the prototype effects (described just below), seem properly applicable to connectionist models, and, indeed, McClelland and Rumelhart (1986a: 182ff.) say so quite clearly and explicitly.

The bottom line in Figure 1 represents the final curiosity that I want to comment on, the two major exceptions to the alignment of theories and disciplines just described and, therefore, major focal points for comparing the adequacy of all four classes of theories. Reber (1967), a psychologist, has long sought to demonstrate that people can and do abstract schematicized representations from experiences. His artificial grammar learning paradigm, motivated originally by theoretical linguistic rules, has become a major focal point in psychology for the debate over rule-learning versus instance-based theories. I examine this debate too in some detail below. Curiously, in experimental psychology, as in linguistics, the debate that

began as rule-based systems (specifically finite-state grammars) versus instance-based alternatives has also evolved so as to include prototype abstraction in connectionist networks versus instance-based approaches (e.g., McClelland & Rumelhart 1986a; Dienes 1992). Thus, in the AGL debate, we see all four types of models listed in Figure 1 represented. The other crossover shown in the bottom line of Figure 1 is, of course, Skousen (1989, 1992) and his analogical approach to modeling language.

## 2. Prototype effects, instance effects and models of categorization

All models of categorization use some measure of “similarity”, either explicitly or implicitly, to predict behavior on concept learning and categorization tasks, yet each does so in a different way. It is precisely those differences which allow us to evaluate those alternative models by comparing their predictions of behavior with actually observed behavior. In the case of models of categorization, there is a large and diverse literature confirming many times over a body of robust experimental effects having to do with the basic characteristics of human categorizations. Any adequate model of categorization will have to account for those effects as well as sundry other effects seen in such closely related tasks as recognition, recall, and learning. I will review those key effects before turning to a more detailed comparison of how the different classes of models shown in Figure 1 try to address them.

First, with respect to the basic characteristics of naturally occurring categories – by which I mean categories not created artificially by rigid definition or through the application of formal operations of logic or mathematics, but instead categories arising spontaneously from our everyday kinds of perceptual experiences – all show the prototype effects demonstrated in such work as Posner & Keele 1968 and summarized in Rosch 1973. These effects include the result that collectively the members of a given category will show a “graded internal structure” in that some members will seem (that is, be adjudged) “more typical” of the category than will other members. Categories will show implicational relationships among the characteristic features of its members. In other words, the presence of some one or more identifiable features, such as scales and gills, will predict with some specifiable probability the expected presence of other features such as fins and float bladders. Finally, categories show “fuzzy boundaries” – that is, for the least typical members of a category, it may not always be clear (that is, we will judge less reliably) whether they are members of the given category or members of a different category. Indeed, many natural categories are apparently nonlinearly separable. This means that the categories actually overlap in their features and that one or more members of one category may be more like the members of some other category in some

respect than they are like the members of their own category. As a consequence, there is no simple, obvious way to separate the two categories definitively by simply comparing general features. One may simply have to list some exceptions by name. The categories of irregular past-tense forms in English illustrate nonlinear separability well. In standard English *bring* looks and sounds much more like *sing* or *swing* (which also overlap) than it does like the *-ought* verb class to which it really belongs. Nonlinear separability appears to be closely related to what Skousen calls heterogeneity in supracontexts and will become an important basis for comparing the empirical adequacy of the various models discussed below.

Lakoff (1987), Comrie (1989), Croft (1991), and Taylor (1995), among others, have surveyed a wide range of descriptive linguistic categories and have argued that all of the demonstrated (or at least posited) linguistic categories known to them show the same prototype effects characteristic of all other naturally occurring categories known to researchers. (I have added to their lists in my 1994 paper.) This issue is important for evaluating the theories represented in Figure 1. The symbolic models do not accommodate these prototype effects without considerable additional theoretical interpretation and processing apparatus. Such models partition items deterministically into linearly-separable categories of behavior. One might argue that language is special in that it is underlain by a competence grammar that is different in kind from other mental faculties and that in such a competence grammar the abstract, symbolic category markers operate deterministically without any discernable evidence of internal category structure or content. A noun is a noun. A verb is a verb, or, at least, a regular verb is a regular verb is a regular verb. The problem with this competence view of rules and category symbols is that ultimately it is empirically wrong. Linguistic categories look and behave like all other natural categories.<sup>2</sup>

Thus, within a competence model either the prototype effects summarized by Lakoff, Taylor, and others show up only in the operations of certain components of the language system, or they arise from some kind of unspecified "noise" in the performance embodiment of the competence system. For example, it might be argued that they show up only when the contentless category symbols of competence grammar become instantiated in acts of performance. In the past few years, some proponents of competence models embodied as symbolic-rule systems have opted for a dual-system interpretation, sometimes linking it to Squire's (1992) neuropsychological hypothesis of declarative versus procedural memory systems (cf. Ullman et al. 1997; Jaeger, Lockwood, Kemmerer, Van Valin, Murphy, & Khalak 1996; Hagiwara, Sugioka, Ito, Kawamura, & Shiota 1999). In Sections 3.1 and 4 we will see independent reasons to question these claims.



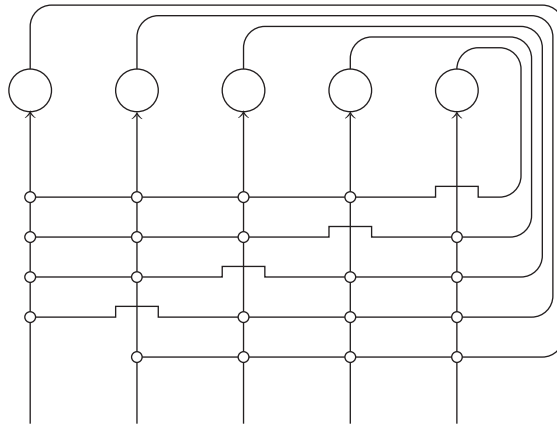
## 2.1 Connectionist models of categorization

The new generation of connectionist models – those inspired by the parallel distributed processing systems of Rumelhart and McClelland and their colleagues (Rumelhart & McClelland 1986a; McClelland & Rumelhart 1986b) – have been designed explicitly to model prototype effects more accurately and in intuitively more satisfying ways than rule-based systems do (McClelland & Rumelhart 1986a). Despite the many criticisms leveled against the connectionist simulations of the English past-tense verb forms by the proponents of symbolic-rule systems (e.g., Pinker & Prince 1988, 1994; Prasada & Pinker 1993), ultimately those critics were compelled by the evidence to adopt something like a connectionist approach to account for just those verb forms which do not fit readily into a competence grammar. Thus, the heart of the debate in linguistics, which I will return to in the last section of this paper, has become now whether connectionist models alone are adequate for accommodating both regular and irregular verb morphology or whether those processes motivate the dual-route model championed by Pinker and his associates.

The criticisms leveled against the connectionist models by proponents of the symbolic-rule systems and others need not be rehashed here in any comprehensive way (see Pinker & Prince 1988, 1994; Chandler 1993, 1994). In general, those criticisms fall into two groups: relatively minor objections to a particular implementation and demonstration, many of which have been redressed in subsequent studies (e.g., failing to use meaning to distinguish between homophonous verbs such as *lie* versus *lie*) and much more serious objections to the framework itself (e.g., objections to back-propagation as a learning procedure or to the vulnerability of such systems to catastrophic interference or to their failure to model different kinds of probabilistic responses accurately). There are, however, several systemic characteristics of connectionist models as models of categorization which are worth reviewing in preparation for the comparisons and evaluations to be presented below.

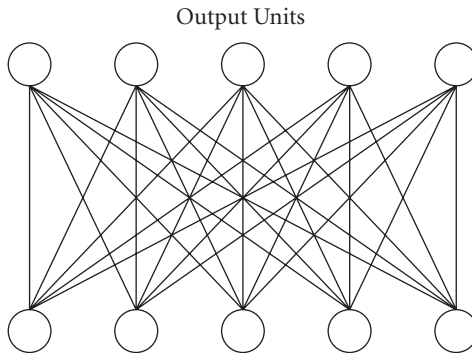
In their PDP volumes, McClelland and Rumelhart actually describe two different types of connectionist models, pattern associators and autoassociators (Rumelhart & McClelland 1986a; McClelland & Rumelhart 1986b), commonly represented as shown in Figure 2.<sup>3</sup>

It is important to distinguish these two types of networks because their behaviors are different in several crucial respects, leading to different strengths and weaknesses in modeling different kinds of cognitive operations. The pattern associator networks (or feedforward networks, as they are also called) learn to associate different input patterns with different output patterns. In their simplest form, every input unit is connected to every output unit. Thus, they are used when one wants to model the association of patterns of similar input stimuli to particular



Input/Output Units

Autoassociator



Input Units

Pattern Associator

Figure 2.

output representations such as alternative category labels or alternative past-tense forms. These are the types of connectionist systems that have been used to model past-tense verb form acquisition and use. In the autoassociator, on the other hand, instead of feeding the input forward to a second set of units, each unit in a common group of units feeds back to every other unit in the network (but not to itself). Thus, the network learns, through exposure to many stimuli, to associate the fea-

tures of those input stimuli into a common representation for them, namely their central tendencies and deviations from them. After training, the output (stabilized levels of activations among the units) is a result of an interaction of the input feature values with the central tendencies (and deviations from them) stored in the network. Autoassociators are used to model prototype effects and have been used to model the artificial grammar learning tasks to be described below.

As noted above, connectionist systems were developed in large part to offer an intuitively more satisfying model of certain prototype effects (McClelland & Rumelhart 1986a), specifically the typicality effects, the graded internal structure, the fuzzy boundaries among concepts, and the implicational relationships among stimulus features. Thus, the pattern associator used by Rumelhart and McClelland (1986b) to model past-tense verb forms does model graded category membership well and fuzzy category boundaries, but such systems can only model linearly separable (that is discrete, non-overlapping) categories.<sup>4</sup> Recall that in nonlinearly separable categories, a member of one category might actually be more like the prototype of another category than it is like the prototype of its own category. Yet, as Shanks (1995) points out, not only do humans routinely learn nonlinearly separable categories, but under some common circumstances, we actually learn them faster than we do some similar but linearly separable categorizations (e.g., Medin & Schwanenflugal 1981).

McClelland and Rumelhart (1986a) note, and subsequent simulations have confirmed (e.g., Plunkett & Marchman 1991), that pattern associators with so-called “hidden units”, namely additional layers of units between the input and output layers, can learn nonlinearly separable categorizations. In fact, as McClelland and Rumelhart (1986a:210) acknowledge, “given enough hidden units”, pattern associators can learn any arbitrary pairing between input and output, but the systems can do so only by using externally triggered back-propagation as the learning mechanism (Shanks 1995). Unfortunately, this sort of back-propagation is not a neurologically plausible model of learning (Crick 1989; Edelman 1987). Nevertheless, Kruschke (1992) developed a hybrid connectionist-exemplar model, *ALCOVE*, by assigning a hidden exemplar-representing unit (a type, not a token) to each different stimulus item included in the training set (one meaning of the term “exemplar”), but he still used back-propagation as the learning mechanism for adjusting connection weights. Thus, even though *ALCOVE* does model nonlinear separability well and overcomes catastrophic interference, it still suffers from the theoretical and empirical problems inherent with back propagation.

Besides not corresponding to any known neurological process, back propagation also encounters several other empirical problems as a learning procedure due to the delta rule used to implement it. The delta rule is the formula used to “strengthen” (increase) the connection between units that have both been activated on a given training trial and to “weaken” (decrease) the connection between

an activated unit and an inactivated unit on a given training trial. This latter feature is usually called “decay”, and that is one of the theoretical problems with the delta rule. As Shanks (1995) noted in passing, there is virtually no empirical justification for the notion that people forget or weaken a memory trace once it has been laid down (see also Estes 1994). Certainly memory traces are not intentionally weakened as part of a learning process. On the contrary, the evidence suggests much more strongly that what we call “forgetting” is better characterized as a problem of memory access rather than one of actually forgetting something once learned (see the discussion of proactive memory interference below). Shanks concluded that any learning model that relies on the notion of memory decay to make its outcomes come out right is inherently suspect.

The delta rule creates different kinds of empirical problems for pattern associators and autoassociators. For example, Shanks (1995) described several sets of findings that he considers problematic for certain instance-based accounts of associative learning (specifically, Hintzman's multiple trace model and Nosofsky's generalized context model). Three of those findings – perceptual learning, latent inhibition, and conditional cue extinction – seem to me to be even more problematic for the delta rule and connectionist models of learning while not at all problematic for Skousen's analogical model.

Perceptual learning refers to the incidental learning about background or contextually coincidental features not having any evident significance for the concept being learned. In a pattern associator, the nonsignificant features would be as likely to occur with outcome A as with outcome B. Therefore, the decay function would keep the value of those connections very low, having no real information value. Meanwhile, in the autoassociators, features occurring incidentally with input exemplars will take on an exaggerated value in the resulting representation. Both results turn out to be wrong when compared with experimental data. McLaren, Kaye, and Mackintosh (1989) found that occasional, coincidental exposures to co-occurring stimuli may enhance the subsequent associative pairing of those stimuli even if their co-occurrence has not been significant or consistent in the past. Thus, a pattern associator that reduces their value to chance underestimates their significance. Conversely, McLaren et al. (1989) also described what they labeled as “latent inhibition”, the second finding that is problematic for connectionist models. If a stimulus is presented on its own with no consequences and no association with an outcome, later pairing of this stimulus with an outcome on subsequent presentations will actually retard its association with that outcome when compared with the learning rate for a new stimulus-plus-outcome pairing.

The third experimental finding that Shanks considered problematic for those instance-based models that he considered, but which I believe actually to be more problematic for connectionist models, is the phenomenon of “cue extinction”. Simply put, cue extinction refers to the empirical observation that if a subject has

learned to pair a given cue with a given outcome, then presenting that cue on subsequent trials with no associated outcome will first reduce the probability of responding to that cue with the previously associated outcome and will lead eventually to the apparent “extinction” of the association. The delta rule would appear to model this accurately. However, Shanks finds “curious” the fact that so-called “conditional inhibitors” do not extinguish in the same way. For example, if a rat first learns that a flash of light signals that a shock is to follow and that a light-plus-a-tone signals that no shock will follow, simply sounding the tone alone with no associated outcome will neither extinguish nor even reduce the effectiveness of the light-plus-tone cue as a signal that a shock will not follow. The delta rule predicts that the effectiveness of the tone as a cue ought to be reduced, as do the instance-based models considered by Shanks. As I demonstrate below, Skousen’s analogical model appears to account for all three phenomena (perceptual learning, latent inhibition, and extinction) easily and naturally within a single mechanism.

Two other important issues with respect to the adequacy of connectionist systems as learning models are the number of training exposures required to associate a pattern with a given outcome and then the number of subsequent trials required to learn a contrasting pattern or to change a previously established response pattern. Connectionist modelers using back propagation rely on hundreds or sometimes even thousands of trials, each causing a tiny, incremental change in the connection weights to “train up” the network. Yet, as Estes (1994) reminds us, real people can often learn simple concepts in one or two trials, and they can begin to reflect the objective probability of a variable outcome in as few as ten trials. Moreover, subjects learn a new, different response for an established categorization-plus-response equally quickly given many previous exemplars (tokens) of the old category response or only a few. Theios and Muise (1977), for example, reported a reading latency study in which they compared reading times for real words to reading times for matched pseudo-spellings (e.g., *grene* versus *green*). As one might expect, they found that the real-word spellings were read aloud about 20 msec. faster than were the pseudo-spelling homophones. Almost as an afterthought, though, Theios and Muise had the same subjects repeat the task with the same stimuli. Again as one would expect, there was a learning effect. Their subjects increased their average speed by about 26 msec. in reading the pseudo-spellings aloud, which made their latencies for these spellings just as fast, and sometimes even faster than the latencies had been for reading the real words aloud on the first pass. Back propagation cannot replicate learning performances such as these, especially as the subjects were given no explicit feedback as to which readings may or may not have been “correct”.

The final issue I will discuss about connectionist models of categorization also provides a transition into a discussion of exemplar-based models. McClelland and Rumelhart (1986a: 189ff.) claimed that their models can represent in the same net-

work at the same time both representations of prototypes as well as coexisting representations of “particular, repeated exemplars”. The example they describe is a network that has been trained on repeated exposures to a variety of dogs – all associated with just the category label “dog” – and repeated exposures to two specific dogs, one named “Rover” and the other named “Fido”. In time the network associates each of the names with specific subpatterns of “dogs”, i.e., specific characteristics. However, as described in the next section, this is not what the proponents of the exemplar-based models mean by exemplar-based or instance-based representations. They are not talking about a mini-prototype created from repeated exposures to a given individual (a type representation) in which the individual encounters (tokens) are somehow amalgamated into a generalized schematic memory for the individual while the details of the separate encounters are discarded or lost. What we have is not a category-to-exemplar relationship but a category-to-subcategory relationship.<sup>5</sup> The reason this distinction is important will become clear in a moment when we examine some of the characteristics of exemplars that McClelland and Rumelhart's model cannot account for – at least not without adding an actual exemplar model to their existing connectionist model.

McClelland and Rumelhart (1986a:200ff.) also describe their simulation of several instance-based effects that Whittlesea (1983) had observed in a series of experiments on the rapid perception of letter strings. The strings represented systematic distortions (letter changes) from a “prototype” letter string, some distortions leaving a string “close” (more similar) to the prototype and some “further” (less similar) from it. The subjects first trained on the category exemplars, then Whittlesea tested their abilities to read and copy the strings accurately when they were presented very briefly on a tachistoscope. He was interested in how much string similarity to the prototype versus similarity to a more distant exemplar would help string perception. He found that both helped about equally well. McClelland and Rumelhart did indeed replicate Whittlesea's basic findings, after a fashion. They trained an autoassociator network on a set of strings from a single category (one used in Whittlesea's study) by giving their network a set of exemplars exhibiting systematic distortions of that letter-string prototype. The autoassociator, of course, developed a central-tendencies representation for those exemplars that were close to the prototype, but the training set also consisted of enough units representing enough features that it was also able to create localized representations for the outlying exemplars of the training set. Thus, when McClelland and Rumelhart tested the system with new exemplars that were closer to the prototype than to the outliers, it activated the units associated with the prototype more strongly, and when they presented it with a test item that was more similar to one of the outlying members of the training set than it was to the prototype, it activated those units more strongly than it did the prototype.

While McClelland and Rumelhart did replicate Whittlesea's results for the outlier exemplars and for the prototype, these are not the sorts of instance effects that have motivated and sustained work on exemplar-based models for over two decades. The connectionist model did not represent competing prototypes within the same system, as did Whittlesea (see Whittlesea 1987). Therefore, it did not address the key issues of instance-based categorization. For example, it did not have to decide whether a given exemplar was a member of category A or category B, although it is possible that it could have learned to do so so long as the two categories were linearly separable, which, as it happened, Whittlesea's original training sets all were. Shanks (1995) reports that he has replicated just such instance effects in categorization (involving genuinely competing categorizations) in a pattern associator, using back propagation in a network containing enough units to allow the outlying members of the categories to be represented distinctly. However, they were linearly separable categories, and, thus, he too did not have to address the issues of overlapping, or nonlinearly-separable, categories and the consequent heterogeneity of form and outcome for those categories.

## 2.2 Instance-based models of categorization

Virtually all the work in cognitive psychology on instance-based models over the past twenty years follows either directly or indirectly from Medin & Schaffer 1978, a paper truly deserving the appellation "seminal". What Medin and Schaffer were the first to notice and demonstrate was that all the work up to that time on prototype-based theories of categorization had confounded similarity to the presumed prototype of a category with similarity to the individual members of the category. The conventional wisdom of the time was that degree of similarity to a prototype, real or implied, correlated significantly with such things as recall, recognition, categorization accuracy, and categorization speed. Medin and Schaffer showed that similarity to even an outlier member of a category, an instance, creates essentially the same effects that similarity to a prototype seems to. This includes so-called "gang effects". An item close to a cluster of items, whether near the prototype or closer to a cluster of outliers, will show "gang effects" (that is, a stronger influence from that nearby cluster). What followed naturally from those findings was the hypothesis that comparison to memories for individual instances of experience was alone sufficient to account for the observed categorization behavior. Twenty years of subsequent research has continued to debate whether exemplar-based models alone are always sufficient to account for category learning and classification behavior or whether there remains any evidence strong enough to compel us to posit additional neuropsychological processes, whether symbol-rule based or connectionist based, for abstracting a schematicized category representation of our experience,

of encoding that schematicized representation as some sort of long-term structural change in our brain, and then using that schematicized representation to interpret and respond to new encounters with stimuli.

Psychologists have long recognized that in addition to whatever abstract category knowledge people may have, they also have the ability to remember and recall episodic memories – that is, memories for specific, biographical experiences with specific exemplars (tokens) of people, places, and things. Such episodic memories, however, have generally been seen as some distinct kind of memory experience or memory system such as encompassed in Tulving's (1983) semantic versus episodic memory systems or Squire's (1992) declarative (episodic) versus nondeclarative memory systems (the latter comprised of various generalized cognitive skills based on abstracted semantic generalizations). The theoretical claim for structured knowledge representations is that somehow our brains extract generalizations from our everyday experiences (and in the process discard or ignore much of the individualizing detail of the experiences that fed into the generalized knowledge structure) and somehow organize those abstracted generalizations into schematicized knowledge structures that we use subsequently to interpret and respond to new experiences.

In recent years Pinker and others have associated Pinker's dual-system model of linguistic behavior to dual-memory models such as Squire's (see, for example, Jaeger et al. 1996; Ullman et al. 1997; Hagiwara et al. 1999). The challenge that exemplar-based or instance-based models pose to all such dual-system models is the claim that a system which can account for all the relevant data by comparing input directly to memory representations for individual episodes of experiences obviates any empirical justification for a separate system of schematicized knowledge representation, including any schematicized knowledge of language – which is to say, grammar.

### 2.2.1 *Instance-based models versus connectionist models of categorization*

In this section I describe in more detail some of the characteristics of exemplar-based models which make them both theoretically and empirically superior to the schema abstraction approaches of connectionism (for other discussion of this issue see Chandler 1994, 1995). Exemplar- or instance-based models are models that do not posit schematicized representations of knowledge which have been somehow abstracted away from our episodic memories for the experiences that underlie and motivate learning. Instead of interpreting new or on-going experiences by comparing current perceptual input to schematicized representations of past experiences, an exemplar approach posits that we interpret the new input by comparing it directly to one or more episodes of past experiences evoked collectively into working memory by experiential (perceptual, motor, affective) similarities between the current input – the probe – and those episodic memory representations. Those evoked



memories provide the basis for interpreting the probable significance of the new input and for responding to it.

As mentioned earlier, exemplar-based models appear to account for many of the prototype effects or central-tendency effects seen in concept learning and categorization studies just as well as the prototype and connectionist models do, models which were created explicitly for doing so. Thus, Hintzman's multiple trace model (1986) replicates closely various sets of human data derived through experimental studies of categorization, recall, and recognition. Nosofsky, using his generalized context model (1986, 1990, 1992), has also replicated very closely (accounting for 97–99% of the variance) a variety of studies on categorization, recognition, familiarity judgments, and category learning. Nosofsky and Palmeri (1997) have modeled reaction time changes during learning, and Logan (1988) has demonstrated an instance-based model of automaticization during learning. That exemplar-based models account for the prototype effects as well as prototype and connectionist models do is a necessary but not sufficient prerequisite for theoretically preferring the former class of models over the latter. However, as described earlier, exemplar-based models also account for experimental data on category outliers and the learning of linear versus nonlinear separability that the connectionist models do not account for, at least not without incorporating memories for instances into the system. However, exemplar-based models also exhibit other important advantages over the other types of categorization models represented in Figure 1.

One of the most obvious advantages enjoyed by exemplar-based models of cognition is that they do not have to posit a separate system for the autobiographical memories that we all retain for episodes of personal experiences – including personal experiences of being told something by someone. So far as I know, no one has ever attempted a connectionist model of episodic memory processes, and indeed, the general view appears to attribute episodic memory to a different neurological and psychological memory system.<sup>6</sup> Unfortunately for the proponents of a schematic abstraction component in dual-memory systems, there is abundant evidence that even in seemingly abstract acts of categorization, our brains retain, have available, and use on demand much presumably irrelevant perceptual information about the physical details of our previous experiences (e.g., Alba & Hasher 1983; Kolers & Roediger 1984; Burton 1990). In two recent papers, Barsalou has summarized evidence for his position that all categorizations, even seemingly very abstract ones such as “fairness” or “honesty” are perceptually based on episodic memories for personal experiences (Goldstone & Barsalou 1998; Barsalou 1999).

Kolers and Roediger (1984), for example, noted that repeating in a test such “non-significant features” as speech cadence, voice pitch, typography, word spacing, modality (written versus oral) and other physical attributes of stimuli all enhance subsequent performance on memory and categorization tests. In a study

of instance-based versus prototype learning effects, Whittlesea and Dorken (1993) also found evidence that measures of stimulus similarity alone were not adequate to account fully for his results. He noticed that details about the perceptual context – the entire episodic context – were important for predicting and accounting fully for subsequent test behavior. Work such as Kelly 1996 and Peters 1999 continues to demonstrate that virtually any detectable physical characteristic of words may contribute to how people interpret them as probes. Word-internal characteristics such as word length (letters, phonemes, syllables), stress patterns, vowel nuclei, and syllable structure all contribute to the likelihood that people will categorize a given nonce word as a noun or a verb. No one of those features alone nor even any collection of them operates as a “rule”, yet people know intuitively that those differences in physical characteristics correlate somewhat with lexical category and therefore all contribute to the probabilities that a test word might be taken as a noun or as a verb. Such perceptually-based categorizations also appear to play a role in language change as well as language acquisition (Cassidy & Kelly 1991). Johnson and Mullenmix (1997) have also recently compiled and reviewed evidence that such seemingly nonsignificant perceptual details as voice quality affect speech perception and processing – that is to say, on line recognition and classification of speech segments.

Research such as that just cited validates Skousen's (1997) claim that “all variables of the dataset are considered a priori equal”; in other words, we cannot determine ahead of time which variables might become important for a subsequent probe. In small datasets of exemplars, such as often is the case in experiments, and perhaps in the early stages of language acquisition, stimulus sampling and changes in attentional focus may show up in experimental effects. Over the long run, however, the law of large numbers takes over and such effects effectively cancel out one another except as a more generalized notion of “noise” contributing to “imperfect memory”. The fact that all perceptual information encoded into memory may remain potentially available for subsequent, unanticipated use does not mean that all features will be used equally. As we will see below, there is evidence suggesting that subjects do not consider all segments of letter or sound strings as equally important for arriving at analogical comparisons.

The fact that apparently any perceivable features that are encoded as part of the memory for an experience may be tapped subsequently as part of the probe seeking memories for comparison has profound implications for theories of category learning and classification behavior. Instead of requiring our brains to try to anticipate before the fact what perceptual variables and what categories might be important to us in some future circumstance, exemplar-based systems allow us to form ad hoc categories on demand. The input probe itself creates the category by specifying which variables are to be searched for and used to activate a set of experiences from memory. Consider as a thought experiment a room full of a thousand people.<sup>7</sup> We could use many different labels to pick out indefinitely many different

subsets of the population. We do not know ahead of time what subsets we might want to identify for some given purpose, and there is no need to, indeed no advantage for trying to (or reason to try to) organize them into a myriad of intersecting potential categories based on our experiences with some of them as teachers, husbands, wives, Presbyterians, or mathematicians who speak Swahili, etc. Barsalou (1983, 1989) has studied the effects of just such ad hoc categorizations experimentally and found that ad hoc categories also show the very same prototype effects and exemplar effects exhibited for allegedly prestructured categorizations, and a moment's reflection will confirm that the same effects emerge for the categorizations named in the above thought experiment. (Consider, for example, Lakoff's 1987 discussion of "bachelor".) This is an important finding because it suggests that the effects arise from assembling any collection of items sharing any arbitrarily identified set of features rather than resulting from our brains constructing knowledge structures. Our ability to create ad hoc, perceptually-based categorizations on demand implies that our memory systems encode a rich store of sensory information about our experiences without regard for what information may or may not be useful to us at some later date.<sup>8</sup>

Earlier, I described the theoretical and empirical problems that such incidental learning of perceptual features poses for connectionist models. Shanks (1995) argued that the perceptual learning effects and latent inhibition demonstrated by McLaren et al. (1989) and the conditional cue extinction demonstrated by Zimmer-Hart and Rescorla (1974) are all problematic for the exemplar-based learning models that he examined (i.e., Hintzman's 1986 multiple trace model and Nosofsky's 1986, 1992 generalized context model). As noted earlier, however, Skousen's AM suggests a straightforward interpretation of those learning effects in the form of some further thought experiments.

An analogical account may begin with a learner who is going along blithely laying down episodic memories rich in perceptual as well as affective, kinesthetic, and motor features, most likely without any particular regard for which features may or may not turn out to be important in the future, as proposed by Burton (1990) and Barsalou (1999). The memories for individual episodes may be filtered through stimulus sampling (Neimark & Estes 1967) or through selective attention (Nosofsky 1986). They may be contaminated with "noise", all of which contribute to imperfect memory, but over the long run, effects such as those will tend to cancel out one another. These memory instances are available, nonetheless, to become part of a dataset created on the fly by entering memory with an externally presented probe, such as occurs in an item discrimination learning task. Any collection of episodic memories evoked and linked by an input probe are apt to include coincidentally co-occurring stimuli or stimulus features which will be available to contribute to any new attempts to link any two of those stimuli as newly paired associates. Thus, we get the perceptual learning effect observed by McLaren et al. (1989).

Now consider the same starting point as before, a learner blithely collecting episodic memories full of coincidentally occurring and recurring stimuli and stimulus features. This time an experimenter does not want just to teach the association of repeatedly co-occurring stimuli but wants to associate a new outcome or response to some previously recurring stimulus now used as the probe. It is just because previous representations of the probe stimulus itself are present in the new data set of memories activated by the probe that one sees a retarded learning curve compared to control sets in which the cue and the outcome are both new to the subject. Early in the learning sequence, the familiar probe will access both episodic memories in which the probe stimulus is not associated with any particular outcome and episodic memories in which the probe stimulus is paired with such outcomes. With practice – accumulating episodes – the proportion of memories in the dataset containing the probe and no outcome, versus those containing the probe plus the newly modeled outcome, will shift, and with the shift, the probability of responding to the probe with the associated outcome will also shift.

A parallel process accounts for the extinction of a previously associated cue and response. Given a memory store that includes episodes of a cue being associated with a particular outcome, one can cause that associated outcome to appear to be extinguished by adding new episodes in which the cue no longer predicts the outcome. When a probe activates the memories for the cue, some will have the associated outcome and some will not. As the proportion of no-response episodes increases, it will eventually overwhelm the number of paired-response episodes, leading to a decreasing probability of responding until the behavior appears to be extinguished, even though nothing has really been forgotten or removed from memory. The “curious” example of the “conditional inhibitor” described by Shanks (1995: 34) actually provides a particularly apt example of such extinctions and lack of extinction at work in an analogical system. As Shanks describes it, the subject has already learned – that is, accumulated sufficient episodes of – *light* → *shock* and of *light* + *tone* → *no shock*. Now, if the experimenter were to change the trials to *light* → *no shock*, the response to *light* would begin to extinguish, as just described. However, the fact which Shanks finds “curious” is that presenting the *tone* alone, with no associated outcome (or information value), apparently has no effect on the information value of the *light* + *tone* cue. The presentations show no evidence of reducing or extinguishing the response. The reason, of course, according to the analogical model, is that adding the less specific context *tone* has no effect on the ability of a more specific probe, e.g., *light* + *tone* to activate the episodes containing the more specific combination of cues. This contrasts, as Shanks recognized, with the predictions of Hintzman's (1986) multiple trace model in which the separate episodic memory representations in memory consolidate with one another to arrive at a composite interpretation, as we will see below.

Since exemplar-based models do not try to abstract permanently stored schematicized representations of knowledge from episodes of experiences, such models do not need to posit special learning mechanisms beyond those already required independently for explaining how experiences are encoded into memory in the first place. The problem of accounting for the learning of such structured abstractions of knowledge has long been a core issue in psychology and linguistics as theorists have sought to explain how stimulus response contingencies might have formed or how rules or schematic prototypes for categories might have emerged from one's learning experiences. As noted above, learning about our perceivable world appears to progress with or without explicit feedback, or without overt outcomes being associated with experiences, yet there is still much to be studied and understood and described about learning. It is not adequate to say that it is "just analogy". For example, we do not know yet what counts as an episode or what kinds of pieces our brains chunk experiences into, nor do we have any good ideas about how those pieces of experiences are reconstructed into complex representations and behaviors (for suggestions, see Hintzman 1986; Burton 1990; Barsalou 1999). Nevertheless, we have identified some characteristics of learning which any successful model will have to account for and which an analogical model such as Skousen's appears to account for more successfully than do connectionist models or the alternative exemplar-based models discussed below.

Conceptually opposite to learning is forgetting. So ubiquitous is our subjective experience of forgetting and so commonplace is the observation of it in recognition and recall experiments that many associative models of learning incorporate some factor to account for it (e.g., Hintzman 1986; Rumelhart & McClelland 1986a). Nevertheless, Estes (1994) and Shanks (1995), among others, note that there is virtually no evidence of normal, healthy brains forgetting experiences once they have been encoded into long term memory, and, indeed, Shanks goes so far as to label as suspect any learning model that relies on memory decay to work properly. Instead, what we take subjectively to be forgetting is probably much more accurately described as "noise" in the system, in the memory forming process in the first place (stimulus sampling effects, selective attention effects) or as proactive interference from competing memories which make it difficult or seemingly impossible for a probe to individuate them sufficiently to evoke a particular episodic memory. For these reasons, Skousen's appeal to "imperfect memory" as a modeling device provides a more apt descriptive term than does "forgetting" or "decay". We will examine the effects of "imperfect memory" further in Sections 3 and 4 below.

### *2.2.2 Alternative instance-based models*

The exemplar-based models of categorization extant in the literature differ, among other ways, in how they compare test probes (new input) to remembered episodes and how the systems choose which exemplars in memory to use as the basis for

interpreting the probe. These differences turn out to be what, more than anything else, differentiate the alternative exemplar-based models from one another. I discuss Hintzman's (1986, 1988) multiple trace model (implemented in a computer model as MINERVA), Nosofsky's (1986, 1990, 1992) generalized context model (GCM), several nearest neighbor models (NN) compositely, but especially those of Tversky and Hutchinson (1986) and Aha, Kibler, and Albert (1991), and, finally, Skousen's (1989, 1992) analogical model (AM).

Hintzman's multiple trace model, MINERVA, operates by encoding each encounter (episode) with a stimulus as a separate memory trace. (In his simulations, the traces are series of plus or minus values for features.) The accumulating traces all lie dormant but accessible in a kind of long term memory (LTM) store. Processing begins when a current input experience, a probe, is presented to the system. MINERVA compares the probe features (also a series of plus or minus values for the encoded features) to all of the memory traces in parallel. Each LTM trace that shares any feature values with the probe becomes activated to a greater or lesser extent (strength) depending upon how many like feature values the trace and the probe share. Next, the corresponding features in all the activated traces are summed algebraically to create an "echo". In essence an echo is a temporary, ad hoc schematic representation created by summing the feature values for all of the memory traces that were activated. Thereafter, the echo functions very much as would the pattern of units that would have been activated by that same input into a connectionist system.

MINERVA simulates most prototype effects as well as connectionist systems do, but because it also retains memory traces for individual exemplars, it can also reproduce closely within a single system the exemplar effects that are problematic for connectionist models, such as outlier effects in competing categorizations and nonlinearly separable categories. However, because it produces a schematicized, composite "echo" as an intermediary representation of the activated memory traces rather than responding directly to the information stored in LTM, it also shows some of the liabilities of schematic representations. For example, it will not accurately model the conditional cue extinction problem, as noted by Shanks (1995) and described earlier. The memory traces for the new *tone* → *no outcome* exemplars will be combined with, and therefore interfere with, the older *light + tone* → *no shock* traces to create a composite echo because both traces contain the feature *tone*. For similar reasons MINERVA does not model probabilistic responses. Much like the winner-take-all interpretation in connectionist systems, MINERVA assigns a probe to a category on the basis of the algebraic sum of the corresponding feature values across memory traces. Thus, given the same input to the same system, it will always return the same response, a response equivalent to what Skousen (1989) calls the "plurality" decision rule. Actually if the probe happened to activate the same number of exemplars from category A and from category B equally strongly,

which can happen with probes near outlying exemplars of overlapping categories, MINERVA may generate an algebraic sum of zero for the category representation, i.e., no categorization, which is not empirically accurate (cf. Chandler 1994 for a description of this effect by MINERVA on English past-tense verb morphology). On the other hand, if the sums happened to come out 51 for category A and 49 for category B, MINERVA would always assign the probe to category A, also not an empirically accurate response (cf. Estes 1976). Finally, we will see below another case in which MINERVA did not model accurately human performance on the artificial grammar task because it sums values across exemplar traces.

Nosofsky's (1986, 1990, 1992) Generalized Context Model (GCM) is by most accounts in the literature of experimental cognitive psychology the most successful exemplar-based model tested to date (cf. Ashby 1992; Estes 1994; Shanks 1995). The GCM calculates a conditional probability that a subject will assign a stimulus to a given category based on summing the feature-by-feature similarity to all exemplars in a given category and dividing that value by the summed similarity – again feature-by-feature – to all exemplars in all categories. In most of his simulations, Nosofsky also factors in a multiplier representing an independently derived response bias for a given category; in some studies he has used an exemplar-strength multiplier to represent exemplar frequency (token) values, and in some studies he uses a stimulus-sensitivity factor to simulate imperfect memory as sensitivity or insensitivity to the training exemplars. In studies in which Nosofsky has first obtained from subjects subjective measures of how similar or dissimilar they consider different pairs of stimuli to be, his GCM is able – using those previously obtained similarity values – to predict very accurately the categorizations or recognition probabilities for individual stimuli, on the order of 99% correct for identification and 97% correct for classification (Nosofsky 1986). In studies relying on objective measures of stimulus similarity (e.g., McKinley & Nosofsky 1995; Nakisa & Hahn 1996), the GCM performed less well, as low as 75% correct.

Although inspired by Medin and Schaffer's (1978) Context Model, Nosofsky's GCM has not been tested strongly on its performance with Medin and Schaffer's most important contribution: the exemplar effects described earlier. Instead, Nosofsky and his colleagues have focused on its superior performance in replicating closely the prototype effects. The published studies on it have compared its behavior with that of real subjects on small, artificial categories which are deterministically separable (e.g., line drawings of faces exhibiting "family resemblances", dot patterns sometimes presented as "categories of constellations", and categories of arcs plus radii). It is true that some of the test categories (pairs of categories) have more complex boundaries between them than a simple linear division (e.g., McKinley & Nosofsky 1995) and that the GCM has performed well on these, but they are all deterministically separable categories and do not contain any overlapping, or exceptional, categorizations. Indeed, when applied to language – specifi-



cally, predicting English past-tense forms (Nakisa & Hahn 1996) – the GCM performed much less well, doing only slightly better than a nearest neighbor model and actually worse than a connectionist model. The problem appears to be that in summing feature similarities over all the exemplars of a category, the GCM misses the role of supracontext heterogeneity, identified by Skousen (1989, 1992) as crucial to deriving the most accurate behavior in categorization models. Because the GCM does not take heterogeneity into account, it performs only slightly better than some nearest neighbor models.

The final class of exemplar-based models of categorization that I will describe before turning to Skousen's AM is the nearest neighbor (NN) models (Tversky & Hutchinson 1986; Aha, Kibler, & Alber 1991; Cost & Salzberg 1993; Nakisa & Hahn 1996). Despite their intuitive appeal, nearest neighbor models have not generally enjoyed the interest and popularity afforded other exemplar-based approaches. Most likely this is so because it has been so easy to show empirically that they are wrong, at least in their simplest, most straightforward instantiations (e.g., Whittlesea 1987; Ashby 1992; Nosofsky 1992). The simple empirical fact is that significantly often subjects do not choose the exemplar in memory that is most similar to the input as the basis for their responses. In a study of past-tense forms for nonce English verbs, I found (Chandler 1998) that subjects frequently did not base their analogical extensions on the most similar English verbs, and across subjects almost all of the nonce verbs solicited at least three or four alternative responses, sometimes more, all based on analogies to common, but different, high frequency verbs, of which at most one would have been the nearest neighbor candidate to the target word (and frequently the most common analogies were not to the nearest neighbor). In a direct comparison of a NN model to Nosofsky's GCM and to a connectionist model, Nakisa and Hahn (1996) found the NN model to perform the least well of the three at predicting German plural forms. Clearly, a simplistic measure of similarity in terms of shared features is not adequate for predicting analogical behavior.

### 2.2.3 *AM as an instance-based model of categorization*

In comparing Skousen's (1989, 1992) analogical approach to the other three exemplar-based models just described, I shall assume that most readers are already familiar with the overall framework of his model. (For an overview of AM see the introductory article by Skousen in this book.) Thus, I will comment only on those characteristics of it which seem to me to be most important for distinguishing it from the other models just described. Skousen's AM exhibits at least three major improvements over the other exemplar-based models, all in the algorithm that he describes for choosing a basis for the analogical behavior. One improvement is that it chooses a specific exemplar in memory to serve as the basis for a response rather than comparing the similarity of a probe ("the given context" Skousen calls it) to



some composite representation of the similar items stored in memory. The second improvement is that Skousen has chosen to compare subsets of features, what he calls “supracontexts”, as well as individual features. Finally, the third improvement is to test for heterogeneity within supracontexts.

Hintzman’s multiple trace model and Nosofsky’s GCM, as well as the connectionist models, were designed explicitly to model first and foremost the prototype effects, effects presumed to arise from the composite structure and organization of categories in the mind. Thus, those models incorporate frameworks for amalgamating information across large sets of exemplars and then using that composite representation to accomplish whatever further cognitive tasks the designer has in mind. This works well for replicating central-tendency effects, or even localized-tendency effects created by subsets of exemplars, but it means that individualizing information about instances in memory will be obscured and therefore not available in explicit form for use in those cases when individual differences are most important – namely in the overlapping regions of nonlinearly separable categories. These regions will therefore become important for testing the empirical adequacy of the analogical approach (see my discussion of past-tense verb forms below).

Those approaches that compare inputs to composite representations of categories also share another weakness not seen in the analogical approach. They do not deal with unusual novel input well because they create a central-tendency representation (explicitly in the case of connectionist models and the multiple trace model and implicitly in the case of the GCM) based on the large number of exemplars presented to them. They compare an input probe to that central-tendency representation in order to categorize the probe or to accomplish some other cognitive task. Unfortunately, as Pinker and Prince (1994) have demonstrated for connectionist models, and as Dienes (1992) found for the multiple trace model (I know of no equivalent test for the GCM), the systems do not respond well when presented with a test probe very different from the exemplars used to create the central-tendency representation in the first place. They often return weirdly and unexpectedly transformed versions of the input or simply do not “know” how to respond to them. Given an unusual word such as the nonce verb *ploamph*, used by Pinker and Prince, Skousen’s AM returns *ploamphed* unequivocally as the predicted past tense form from an analogical set that includes *plump*, *clamp*, and *poach* among others.

The second feature of Skousen’s analogical approach that constitutes a major improvement over the other exemplar-based models is his decision to compare subsets of features (supracontexts) as well as individual features among the test probes and exemplars. The validation of this decision on Skousen’s part has to rest ultimately on the empirical success of the model, yet there are at least three sets of independently derived findings which validate his use of additive supracontextual similarities to arrive at an analogically motivated task response. Two of these find-

ings I will describe here. The third, chunking effects, I will defer until my discussion of the artificial grammar learning studies in Section 3. The second and third advantages emerge as unintended consequences of the analogical approach rather than as features that motivated it in the first place and were designed into it explicitly.

In 1977 Hayes-Roth and Hayes-Roth reported the results of a comparison of 24 models of concept learning and classification with what they called their “property-set” model. They compared the performance of those models on a variety of prototype-based categories. (Their paper was published a year before Medin and Schaffer’s 1978 paper.) The major distinguishing characteristic of the property-set model was that it compared not only the individual features of test items to those of the prototype, but that it also created and compared for each category and test item a “property set” of their respective features, the property set being the powerset of all the properties of an exemplar – that is, the features plus all possible conjunctions of those features and subsets of those features. These are essentially Skousen’s supracontexts except that the latter can preserve spatial and temporal order. Hayes-Roth and Hayes-Roth indexed the elements of the property sets for frequency of occurrence, although records of the individual occurrences of exemplars were not retained in memory in their model. Nevertheless, Hayes-Roth and Hayes-Roth found that comparing a probe’s property set (i.e., its supracontexts) with the combined tokens of property sets for all of the other items yielded significantly better predictions of prototype effects on categorization and recognition than did models that simply compared individual stimulus features. We will see strong independent confirmation of this effect in the history of the artificial grammar learning studies reviewed in Section 3.1.

Using supracontexts to construct analogical sets also allows us to account both for implicational contingencies among the component features of exemplars, which the other models discussed here also do well, and for spatial or temporal sequences of features, which is a significant difficulty for connectionist models, whose proponents have had to resort to ad hoc mechanisms for avoiding the problem (e.g., McClelland & Rumelhart 1986a; Dienes 1992; McClelland & Elman 1986).<sup>9</sup> Comparing supracontexts also suggests more natural procedures for comparing items of different length and canonical shape such as a five-letter word and an eight-letter word.

Skousen’s analogical approach describes an explicit algorithm for identifying and accumulating an analogical set, the set of candidate exemplars evoked from memory to serve as the basis for the subsequent analogical process. Skousen describes two “rules of usage” for choosing one of the exemplars from the analogical set, (1) random selection or (2) selection by plurality. The former simply selects any one item from the analogical set at random, but in the latter a person must examine the analogical set and choose the most frequent outcome. In his book, *Analogy and Structure*, Skousen (1992) describes the consequences of using one

rule rather than the other and suggests some of the considerations that might lead a person to rely on one rule or the other. (He even cites an eight-year-old child in a study by Messick and Solley (1957) who chose the plurality rule consciously in order to maximize her “gain” – pieces of candy – in the study.)

The categorization models described above differ in whether they permit the alternative decision rules, yet it is just this option which permits a model to replicate the probabilistic behavior characteristic of human subjects. Real people do not always respond to the same stimulus in the same way. Although the pattern of unit activation strengths that a connectionist network settles into in response to a given input probe might reflect the probabilities of alternative outcomes, in practice the system designers typically take the strongest activation as the response. This is equivalent to Skousen’s selection by plurality rule. Hintzman’s multiple trace model also consistently responds with the equivalent of the plurality rule. The nearest neighbor models identify the one nearest neighbor as the basis for their responses, although ties may be resolved randomly. Thus, none of these models, as typically implemented, is genuinely probabilistic. Nosofsky, on the other hand, has used both kinds of responses (Nosofsky 1986; McKinley & Nosofsky 1995), modeled in this case as a conditional probability versus a plurality rule based on the most probable outcome.

Estes (1994) and Ashby (1992) have examined what kinds of circumstances – i.e., experimental tasks and settings – might dispose a subject to respond one way or the other. Ashby noted that even if a subject were trying to respond deterministically that “noise” in the categorization system could cause him to respond differently once in a while. Moreover, Ashby continued, probabilistic responses may typically model group behavior closer than a deterministic response might. In support of these observations, Ashby cited a survey of categorization studies in which he found that overall more than 94% of the subjects (all adults) were responding deterministically (i.e., by plurality). However, perusal of Ashby’s survey reveals that those studies all involved the conscious forced-choice classification of visual items into one of two simple categories. Moreover, in several of those studies, the experimenters actually offered the subjects a graduated monetary bonus if their response accuracies exceeded a specified minimum level. These subjects did indeed seek to maximize their gains.

In contrast to Ashby’s work, in which the procedures motivated use of a plurality decision rule, Estes (1976, 1994) has often emphasized experimental tasks and materials likely to encourage probabilistic responses. The basic finding of probability learning is that if during the training phase of a category learning task the subjects who are trying to learn the category membership of a given stimulus are told 70% of the time that it belongs to category A and 30% of the time that it belongs to category B, those subjects very quickly begin to guess category A as the answer 70% of the time and category B 30% of the time. In reality there is no evi-

dent basis for assigning the stimulus to one category or the other. The experimenter is simply manipulating the re-enforcement schedule, but the subjects' guesses soon reflect the re-enforcement schedule very closely. Estes has found from such studies that when experimental materials lack any obvious distinguishing characteristics or that when subjects do not know during training that they will later be trying to categorize new material based on what they are currently seeing, they essentially have no choice but to respond probabilistically, sometimes choosing one outcome, sometimes choosing another. Moreover, simply telling subjects that data are probabilistic or asking them to base their responses on their impressions of event probability also leads to probabilistic responses.

Estes (1976: 37) professed not to know what could "explain" probability learning, yet he noticed that event frequencies were the single best predictor of subjects' alternative responses. Identifying a plausible basis for probability learning has long puzzled psychologists (e.g., Hintzman 1988; Shanks 1995). Skousen's random selection from an analogical set provides a simple, elegant, and empirically accurate explanation of what might underlie probabilistic response learning.

#### 2.2.4 *Some considerations of AM as a model of cognitive processes*

In the previous section, I have described how AM operates as an exemplar-based model of categorization, and I have alluded to how an AM approach might be extended to other cognitive operations closely associated with category learning and classification. I have shown briefly how AM might account for certain data on perceptual learning, latent inhibition, and conditional inhibition, data which Shanks (1995) found problematic for the exemplar-based models that he considered and which I find problematic for connectionist models. I have also described how AM appears to offer a ready, natural account of probability learning as described in Estes (1976). Although more extensive demonstrations go beyond the scope of this paper, key components of AM analysis – supracontext comparisons, testing for heterogeneity, and the decision rules – suggest natural, AM-like extensions to a number of other cognitive behaviors such as item recall, recognition, similarity judgments, familiarity judgments, word associations, and response latencies in such categorization tasks as word naming and lexical decision.

AM consists of a procedure for deriving an analogical set of candidate exemplars that are to provide the basis for whatever cognitive operation is to follow on the input probe. The model then uses one of the two decision rules, or "rules of usage", to arrive at a particular response. One rule, random selection, simply chooses one of the candidate forms from the analogical set at random. The other rule, selection by plurality, chooses the most frequent outcome represented in the analogical set, a procedure which implies inspection and comparative quantification of the items in the analogical set. In my discussion of the artificial grammar learning task, the next section, I will suggest a third rule which appears to oper-

ate in certain circumstances. Several of the cognitive behaviors listed above would appear to be explainable as different kinds of inspections of an analogical set. For example, Nosofsky (1990) examined how his GCM accounts for categorization, similarity judgments, identification and recognition as different kinds of comparisons of an input probe to a data set of exemplars. Hintzman (1986, 1988) demonstrated similar extensions of his multiple trace model to the same cognitive activities. They both treated recognition as a special case of categorization with a category of one item whereas similarity judgments were based on comparisons to all exemplars. Nosofsky noted (citing Shepard 1987) that judgments of similarity fall off exponentially as “psychological distance” increases in terms of a decreasing number of shared features. Skousen’s comparisons of supracontexts captures that behavior accurately in that items (a probe and members of the data set) sharing  $n$  features will share  $2^n$  supracontexts and would, therefore, model the relative similarity judgments of pairwise item comparisons accurately as a ratio of the number of supracontexts shared by two items to the total number comprising the two items.

Whereas a sense of similarity may follow from the number of supracontexts shared by a probe and its analogical set, a feeling of familiarity might follow from the relative frequency of an item in an analogical set, influenced also by degree of similarity (cf. Humphreys, Bain, & Pike 1989). Similarly, recall, free and cued, involves finding a match between input and items in the data set, or an item sufficiently close to trigger a sense-of-recall threshold (Humphreys, Bain, & Pike 1989). Stimulus sampling effects (i.e., incomplete or indistinct representations of items in memory) or imperfect memory effects may lead to over retrieval of items into an analogical set. For example, apparent free variation in past tense forms for *sit* and *set* might arise because they are not accurately distinguished phonologically in many exemplars of usage in memory. Such situations, as well as imperfect memory in general, would give rise to apparent proactive interference.

Word association is another cognitive task commonly used to study lexical representation and organization (see Clark 1977). Indeed, there are normed lists of the words most commonly evoked as associations for a given word under different circumstances (e.g., Nelson, McEvoy, & Schreiber 1994). Traditionally such associations are described and explained in terms of shared semantic features (e.g., *doctor* → *nurse*, *table* → *chair*, etc.). However, recent work by Buchanan and her colleagues has revealed that the frequency with which two words co-occur within small chunks of text (ten-word chunks in her studies) predicts the strength of their association better than does analyses into shared semantic features (Buchanan, Cabeza, & Maitson 1999; Buchanan, Westbury, & Burgess 2001). This finding suggests an exemplar-based account of word associations rather than a semantic features account.

Seemingly countless psycholinguistic studies have employed reaction time (RT) as a dependent variable in such tasks as lexical decision, naming (word and

picture), semantic verification, and word inflection. Each of these tasks implies its respective cognitive operations, matching, retrieval, recall, etc., plus time for implementing the appropriate response. Skousen (1992) has described a possible AM interpretation of RT data from a published word naming task. The number of steps needed to derive an analogical set for a given input probe correlated well with RT. More recently, Nosofsky and Palmeri (1997) dipped back into the history of experimental psychology to apply a “random walk” model of predicting RTs to the categorization judgments of Nosofsky’s GCM. Once a set of candidate exemplars has been evoked from memory (into an analogical set in Skousen’s AM), the random walk is a mathematical model of a decision mechanism for choosing among them. A subject is said, metaphorically, to visit the alternative outcomes randomly until the sampling reaches a confidence threshold that triggers the prevalent response. Stimuli evoking alternative responses take longer for the subject to identify and to confirm at a given level of confidence which response is prevalent. Adapting a random walk model to AM exceeds the limits of this paper, but it suggests that RTs may be modeled in AM by applying a random walk model to an analogical set derived according to the procedures of AM.

### 3. The artificial grammar learning paradigm

If the debate over whether morphological processes are better characterized as dual system or single system seems to have preoccupied some linguists over the past decade, certain cognitive psychologists have engaged in a virtually identical debate for more than 30 years now over how best to explain performance on the artificial grammar learning (AGL) paradigm. Although there are many variations on the theme, the basic AGL paradigm dates from Reber (1967). In it a simple finite state grammar (FSG) is used to generate strings of letters, usually three to eight letters long, e.g., *VXV*, *XXVTVJ*, *MVRVVR*, etc. The usual task is for the subjects to study some subset of the strings generated by the grammar as a training set to induce category learning. Sometimes subjects are told ahead of time about the existence of a rule system for generating the strings (the explicit learning condition) and sometimes they are not (implicit learning). After the training presentations, the paradigmatic task is to view both previously seen strings and newly presented strings and to judge them (classify them) as members of the studied category (grammatical) or not (ungrammatical). Subjects are also often asked to identify test strings as “old” (previously seen) or “new”. Although the details of a given experiment can cause the results to vary somewhat, the robust finding is that subjects typically identify whether strings are old or new with better than 90% accuracy (in other words, they have excellent memory for whether they have seen a given string before) and

that they judge newly presented strings 60 to 70% accurately for “grammaticality” (category membership).

Reber (1967, 1989) and others have attributed the better-than-chance categorization performance of their subjects to their having abstracted the underlying FSG and then using that abstracted knowledge to categorize subsequently seen strings as “grammatical” (a member of the category defined by the FSG) or “ungrammatical” (not a member of the defined category). Brooks (1978), and others since, have argued that one does not need to posit induction of some unseen FSG to explain the results. Instead one can explain them equally well by assuming that the subjects remember previously seen strings or at least frequently recurring letter sequences in the training set and then used those memories for the training exemplars to categorize the test exemplars and to rate them as “old” or “new”. More recently, some researchers have sought to demonstrate that the subjects were abstracting connectionist-like prototype representations of the training sets and responding to the test items on that basis rather than on the basis of comparison to some presumed grammar (e.g., Dienes 1992; Altmann, Dienes, & Goode 1995). In either case, for more than three decades now researchers have debated vigorously whether the data show subjects to be responding on the basis of comparison to remembered exemplars or on the basis of some sort of schematic knowledge representation, including rule schema or possibly on the basis of both, a dual-system account.

To readers familiar with Skousen’s notion of supracontext (see Section 2.2.3 above), much of the history of the AGL research looks like a halting progression toward an inevitable end. Brooks (1978), for example, showed that fragmentary memory for frequently recurring letter pairs and triplets predicted actual classification performance closely. Reber and Allen (1978) showed that memory for frequently recurring string-initial and string-final letter combinations in the learning strings served as important “anchor positions” for learning what are permissible letter sequences and consequently for judging the grammaticality of new strings. Dulany, Carlson, and Dewey (1984) and Perruchet and Pacteau (1990) also demonstrated further that subjects based their grammaticality judgments, at least in part (and possibly in large part), on their memories for frequently recurring bigrams and trigrams. In 1990 Servan-Schreiber and Anderson consolidated the previous research into a coherent, and highly predictive, theory of “associative chunk strength”. They demonstrated that subjects noticed hierarchical arrangements of frequently recurring chunks (letter string fragments) and used that knowledge to classify new strings.

In two studies Vokey and Brooks (Brooks & Vokey 1991; Vokey & Brooks 1992) sought very carefully to unconfound overall similarity of test exemplars to training exemplars from grammaticality. Their multivariate analysis showed that both grammaticality and similarity to the training exemplars contributed inde-



pendently to the classification performance (grammaticality judgments) of their subjects, thus a dual-system model. Unfortunately, in controlling for overall similarity between whole training items and whole test exemplars, measured only in terms of individual letter differences, Vokey and Brooks failed to control adequately for the associative chunk strength just recently described by Servan-Schreiber and Anderson, which turned out to be seriously confounded with grammaticality in their study. Knowlton and Squire (1994) attempted to replicate Vokey and Brooks' studies while controlling better for surface similarity versus grammaticality, but as Perruchet (1994) has demonstrated, Knowlton and Squire's replication was also seriously confounded in that the "grammatical" test items shared many more bigram and trigram letter sequences with the learning strings that did the "ungrammatical" test items. In AM terms, this meant that the grammatical test items shared significantly more supracontexts with the training items and generated significantly larger analogical sets, almost 50% larger (assuming perfect memory), than did the ungrammatical test items.<sup>10</sup>

In what appears to be the most carefully controlled study to date, Meulemans and Van der Linden (1997) presented subjects with four sets of test strings carefully balanced for associative chunk strength (bigrams, trigrams, and anchor positions) and grammaticality. Indeed, no test items contained any "illegal" bigrams or trigrams. The results were that subjects who had been trained on only a relatively small subset of the possible letter strings generated by the FSG (the usual practice in AGL studies) showed a strong bias for judging grammaticality on the basis of surface similarity to the training strings, i.e., exemplars. On the other hand, for subjects who were trained on a very large subset of the possible strings generated by the grammar (virtually the entire set minus the test items), the effect for surface similarity disappeared and underlying grammaticality appeared to be the only significant basis for their classifications. Unfortunately, as Johnstone and Shanks (1999) showed, those researchers too had overlooked yet another important distributional variable. While Meulemans and Van der Linden had controlled well for anchor positions and for global (i.e., overall) bigram and trigram chunk frequency, they had not controlled for where in the strings different chunks typically occurred, and as Johnstone and Shanks showed, the occurrence of chunks in novel positions in the test items correlated significantly with grammaticality in Meulemans and Van der Linden's test items.

In controlling for all the significant variables identified in Meulemans and Van der Linden, Johnstone and Shanks control essentially for skewed distributions of supracontexts shared by the test set items and the training set items. Surprisingly, however, a full comparison of supracontexts reveals a distribution skewed in the wrong direction. As shown in Table 1, the nongrammatical test items actually share about 11% more supracontexts with the training items than the grammatical test items do. However, since the test items contained, by design, no illegal two or three



**Table 1.** Supracontext comparisons for AGL

	Associated	Non-associated	Total
Grammatical	284/122	257/148	541/270
Non-grammatical	382/114	218/217	600/321
Total:	666/236	475/365	

Each entry denotes shared supracontexts/unattested supracontexts.

(Data from Muelemans & Van der Linden 1997)

letter sequences, the subjects had to be evaluating longer sequences in order to classify any items correctly, and indeed the ungrammatical test items do contain significantly more four-letter sequences that are not attested in the training examples than the grammatical test items do (22 versus 8 respectively). Table 1 also shows the total numbers of supracontexts occurring in the different test sets but not attested in the training set. We see that the ungrammatical test items contain almost 20% more supracontexts that are not attested in the training set than do the grammatical items. If we focus on the so-called anchor positions, supracontexts that contain string-initial or string-final segments (thought to be more important in the AGL tasks), the confound becomes even more severe with the ungrammatical strings having 45% more unattested supracontexts than the grammatical items do and 100% more supracontexts for which both the initial and final letter strings are unattested in the training set.

Taken together, Johnstone and Shanks' "novel chunk position" criterion and the supracontext distributions just described suggest that the subjects in that particular experiment (experiment 2b in Meulemans and Van der Linden's study) were responding more on the basis of item differences than item similarities. Instead, the differences in performance between the first set of experiments reported in Meulemans and Van der Linden and their second set suggests a change in response strategy – decision rule, we would say – on the part of their subjects. Given a relatively small training set, the subjects appeared to look for similarities between the test items and the training set. Given a very large training set, one in which individual items are necessarily less distinguishable in memory (probably due to proactive interference), subjects appear to focus on looking for unusualness (or unfamiliarity) in the test data. If confirmed, this suggests the possibility of a third – possibly task specific – kind of decision rule (in addition to the two described by Skousen) being applied to an analogical set. The new decision rule, which might apply when the data set appears to provide exemplars for almost all of a test item's supracontexts, would judge the item as not belonging to the category represented most commonly in the analogical set.<sup>11</sup>

In their concluding comments on Meulemans and Van der Linden's study, Johnstone and Shanks note that all the significant variables identified by them

(those listed above) together account for at most only about 25% of the variance in performance and that the performance of only 8 out of 40 subjects in the experimental groups was predicted by these variables. They call for more research to identify further what kinds of knowledge and strategies underlie performance on the AGL tasks. I would submit that the AGL studies are especially vulnerable to Type I errors, finding a significant effect when in fact none really exists. Random performance on the classification tasks is 50%. Given 32 items (16 grammatical, 16 ungrammatical), a subject only needs to classify four or five items (13%) correctly on the basis of remembered knowledge about the learning items while responding to the others randomly in order to score 65% correct on the test. As Perruchet (1994) and others have pointed out, even control groups having no training sometimes classify the test items as well as 55% correctly, evidencing an apparent learning effect during testing. Thus, remembering only a very small subset of training items for whatever idiosyncratic reasons could prepare a subject to classify four or five test items correctly. In any case, Johnstone and Shanks concluded that it was time to rethink the AGL paradigm and to look for evidence other than simple classification of test strings to try to determine what kind of knowledge base subjects are using as the basis for their classifications. An AM interpretation does suggest further tests of the AGL task yet to be conducted, but meanwhile we should consider at least two other aspects of AGL studies which are commonly cited as offering strong support for the grammar abstraction view: the performance of amnesic patients and performance on the so-called transfer tasks.

### 3.1 Performance of amnesic patients on the AGL task

In a series of studies, Knowlton and Squire and their colleagues have reported the performance of amnesic patients on a variety of classification learning and recognition tasks, including the artificial grammar learning task just described (Knowlton, Ramus, & Squire 1992; Knowlton & Squire 1993, 1994, 1996; Squire & Knowlton 1995). Consistently across all these studies, amnesic patients have shown normal or near normal performance on most classification tasks, classifying exemplars correctly at least 60% of the time or better. However, in sharp contrast to the normal control subjects, who typically recognized 60 to 80% of the time whether they had seen a test exemplar previously in the training set (depending on how large the training set was), the amnesic patients scored much lower on recognition, suggesting that the amnesic subjects had very poor memory for the individual training exemplars. Squire and his colleagues took these results as strong evidence that their subjects were abstracting some sort of schematic representation (whether rules or a prototype) from the training examples because the major clinical manifestation of hippocampal amnesia is the inability to learn or recall new episodic memories,

including memories for the training episode. (Indeed, by the time they did the test phase, the amnesic patients often did not recall having undergone the training phase in the experiments.) Thus the performance of these amnesia patients, he concluded, cannot be attributed to exemplar-based comparisons because the evidence shows that they did not remember the exemplars, yet they were able to classify exemplars, both new and old ones, just as accurately as the normal control subjects could. I examine this claim in some detail in this section because Ullman, Corkin, Coppola, Hickok, Growdon, Koroshetz, and Pinker (1997) adopted a parallel argument to explain a dissociation between regular and irregular past-tense verb performance seen in patients with a variety of neurological impairments, and Ullman et al. (1997) attributed that dissociation specifically to differences between declarative memory for episodic events and procedural memory for implicit cognitive skills as posited by Squire.

Squire (1992) has long been the major proponent of the neuropsychological distinction between declarative memory versus procedural memory, a distinction based mostly on many years of study of memory impairments associated with damage to the hippocampus. Bilateral injury to the hippocampi typically results in severe anterograde amnesia, the inability to recall new experiences. Although the definitions of declarative and procedural memory have evolved over the years, declarative memory deficits are identified most closely with hippocampal damage and show up as an inability to recall specific episodic events that have occurred since the onset of the amnesia and as an inability to learn new factual knowledge or even new words. Sometimes the working definition of declarative memory is associated with the conscious recollection of events and other times with just their implicit, tacit recollection, as in these AGL studies. Procedural memory, on the other hand, refers to generalized, decontextualized cognitive and motor skills and a generalized semantic memory (Squire 1992; Knowlton 1997).

Although Squire's declarative versus procedural memory distinction is widely accepted in the literature of neuropsychology, it is by no means uncontroversial, and there are both neurologists and psychologists who question the theoretical and empirical justification for the dichotomy (e.g., Humphreys, Bain, & Pike 1989; Hintzman 1990; Cohen & Eichenbaum 1993; Shanks & St. John 1994; Shanks 1995). Critics of the declarative versus procedural memory hypothesis argue on theoretical grounds that behavioral dissociations on different tasks (e.g., categorization versus recognition) do not warrant logically the conclusion that different memory systems are being recruited and used because the different tasks may only indicate that different processes are operating within a common system. They go on to note that in the case of Korsakoff's syndrome (one of the more common causes of hippocampal amnesia), the neurological deficits are seldom confined to the hippocampus and often spill over into the frontal areas which are implicated in the coordination of spatial and temporal contexts in retrieving episodic memo-

ries, especially of verbal memories (e.g., Buckner 1996; Nyberg, Cabeza, & Tulving 1996). Importantly, brain scan data (MRI) show frontal area activations during the categorization phase of the AGL tasks, tasks that rely on processing tempero-spatial sequences of letters (Reber, Stark & Squire 1998).

The neurological evidence aside, other behavioral evidence also suggests that amnesic patients such as those studied by Squire and his associates actually do retain at least some of the training episodes in memory, but have difficulty accessing them on demand for subsequent comparison (Moscovitch & Umiltà 1991). Graf, Squire and Mandler (1984) and Graf, Shimamura and Squire (1985), for example, asked a group of amnesic patients to study lists of words and then tested their recall for those words in response to different kinds of cues. They found the amnesic patients responding either near normally to a partial-word cue to be completed with “whatever word first comes to mind” or very poorly for the same words when told to complete a word-fragment cue with a specific word from the training list. This disparity suggests strongly that the words were present in the patients’ memory, but that they could not isolate on demand a specific word from the training list to match to a specific word cue, a result that is consistent with other observations (e.g., Hintzman 1990) that amnesiacs are especially vulnerable to proactive interference – that is, they have difficulty trying to separate and distinguish among different episodic memories with very similar, partially overlapping contents and contexts.

This information suggests an alternative interpretation for the results of the AGL studies reported by Squire, Knowlton, and their colleagues. In all of those studies the amnesic patients categorized the test strings virtually as well as the normal control subjects did even though their performance on recognizing whether the test strings were “new” or “old” was significantly worse than that of the normal control groups, yet although significantly worse, their memory performance was still significantly above chance. Thus, they actually did show evidence of having some episodic memory basis for classifying test strings. The question is whether the degraded performance on the recognition tasks should, in a single process model such as AM, predict degraded performance on the classification task as Squire, Knowlton, and others have assumed (e.g., Knowlton, Ramus & Squire 1992).

Three additional observations provide indirect evidence that the amnesic patients in these studies actually were having difficulty accessing recorded memories accurately rather than failing to register (at least temporarily) new memories. First, Knowlton and Squire (1993) reported that a group of amnesic patients who had seen *fewer* training items actually performed better on the recognition test than had a control group of amnesic patients who had seen more training items. Second, in Knowlton, Ramus, & Squire 1992 both amnesic and normal subjects did worse on the task after a second pass through the materials. Together these results suggest a reduced opportunity for proactive interference in the first study – and thus

improved performance – and an increased opportunity for it in the second study – and thus reduced performance from a kind of interference to which amnesiacs are known to be especially vulnerable. The third observation is that Nosofsky and Zaki (1998) first replicated almost exactly Knowlton and Squire’s (1993) methods, materials, and results for a closely related task (involving learning, recognizing, and classifying prototypical dot patterns) on a new set of amnesic patients and normal controls, and then simulated both the near-normal classification performance and the degraded recognition performance of the amnesic patients with the same exemplar-based (GCM) framework. The only difference in the formulae used to derive the control versus amnesic performance were lower settings for the “sensitivity [to a stimulus] parameter” and a different “power exponent” used to model the “psychological similarity” of exemplars in memory. The effect is “imperfect memory” and is as if the exemplars are not as distinguishable in memory, resulting in the functional equivalence of proactive memory interference.

The strongest invalidation of Squire and Knowlton’s argument that the dissociation between recognition and classification performance by amnesic patients on the AGL task strongly supports their dual memory-system model is the demonstration that Skousen’s analogical approach accounts for both sets of data within a common framework. Again, I shall apply a supracontext analysis to the test items and training items and compare those results to the performance of the amnesic and normal subjects in Knowlton and Squire’s various studies.

Knowlton, Ramus and Squire (1992) and Knowlton and Squire (1993, 1994, 1996) all used the finite state grammar borrowed from Vokey and Brooks (1992) to generate training and test strings. Since Knowlton and Squire trained their subjects on a relatively small number of possible strings, those subjects were probably responding on the basis of surface similarities, as discussed above for Meulemans and Van der Linden’s (1997) study. After 1992, Knowlton and Squire were aware of the confound between grammaticality and surface similarity in Vokey and Brooks’ test materials. They sought to eliminate that confound by controlling for associative chunk strength in the test sets as described by Servan-Schreiber and Anderson (1990). Nonetheless, as seen in Table 2, both their earlier effort and the later study contained a serious confound between supracontext overlap and grammaticality.

As discussed earlier with respect to Johnstone and Shanks’ (1999) discussion of Meulemans and Van der Linden’s (1997) study, subjects need only be able to classify 10 to 15% of the test items correctly in addition to an otherwise random classification in order to achieve the classification scores reported in these AGL studies. Table 3 shows the results of comparisons between test performance and training sets adjusted for increasingly imperfect memory effects as recorded in the recognition scores for the amnesic patients in those studies. Both the amnesic and control subjects scored from 60 to 65% correctly in classifying test items as grammatical or not, but the amnesic patients “recognized” only about 63% of the test

Table 2. Amnesic performance on AGL

	Amnesic % correct	Shared Supracontexts		Unattested Supracontexts	
		Grammatical	Non-grammatical	Grammatical	Non-grammatical
Knowlton & Squire '93	57	518	437	16	35
Knowlton & Squire '96	64	1539	1508	35	81

Table 3. Imperfect memory effects on AGL

	Shared Supracontexts		Unattested Supracontexts	
	Grammatical	Non-grammatical	Grammatical	Non-grammatical
100% memory	518	437	16	35
50% memory	254	216	21	38
25% memory	83	66	65	74

(Based on test materials in Knowlton & Squire 1996)

items correctly (still significantly above chance) as “old” (previously seen in the training set) or as “new”, while the control subjects score from 67 to 85% correctly on recognition. Table 3 shows how the proportions of supracontext overlap changes with increasingly imperfect memory for the training set. I simulated increasingly imperfect memory by reducing the training set by 50% (using every other item) and by 25% (using every fourth item). Also shown in Table 3 are the changes in proportions of unattested supracontexts derived through applying imperfect memory. As Table 3 shows, the predicted classification performance based on supracontextual comparisons (i.e., an analogical model) does not degrade proportionately to the memory impairment suggested by the recognition scores for the amnesic test groups. Thus, the dissociation between classification and recognition does not warrant positing different memory systems for the control groups and amnesic groups.

### 3.2 The transfer condition in AGL

To many researchers working in the artificial grammar learning (AGL) paradigm, some of the most convincing evidence that subjects are indeed abstracting some sort of underlying schematic category structure away from their experiences with the training strings occurs in what has come to be called the transfer condition (Mathews, Buss, Stanley, Blanchard-Fields, Cho, & Druhan 1989; Altmann, Dienes, & Goode 1995; Shanks, Johnstone, & Staggs 1997). The transfer condition involves training subjects on strings generated using one set of letters and testing them on

strings in which different letters have been substituted systematically for those used in the training strings (e.g., *VXTTTV* might become *MPDDDM*). The consistent finding is that although the classification performance of subjects may degrade somewhat, it is still significantly above chance, suggesting that some abstract basis for patterning letters has been learned and abstracted away from the specific set of training exemplars. Knowlton and Squire (1996) found amnesic patients performing comparably to normal control subjects on the classification of letter-set transfer strings (about 60% correct classification in the same-letter condition and about 55% correct in the changed-letter condition). Altmann, Dienes, and Goode (1995) have even demonstrated that the structural regularities abstracted from the letter-set training items can transfer to data types represented in other sensory modalities such as tone sequences or visual symbols.

The two major unresolved questions regarding the transfer condition studies are (1) whether subjects really do transfer structural information about the training set to classification of a changed test set, and (2) if so, whether the transfers represent knowledge abstracted away from the training set during training or might they represent local analogies between test items and particular training items whose retrieval has been triggered by some perceived structural parallel between the test item and one or more training items. Perruchet and Pacteau (1990), Perruchet (1994), and Redington and Chater (1996) all argue that the above-chance performance seen in classification tasks in the transfer conditions arises as a learning effect during the test phase of the experiments. They note that subjects start out responding randomly but improve significantly during the test. This account is plausible because the test strings, both grammatical and ungrammatical, necessarily share more similarities than differences with the training strings. Otherwise, there would be no contest in the classification test. Consonant with the learning effect interpretation is the fact, pointed out by Redington and Chater (1996), that control groups who have never seen the training strings sometimes classify the test strings as well as subjects in the transfer conditions do, i.e., about 55% correctly.

Whatever the basis for the transfer condition performance, it apparently is very limited, only one or two items beyond random guessing. By changing systematically the location of impermissible letters in their test strings, Shanks, Johnstone and Staggs (1997) found that violations in repeating a letter were the only errors noticed reliably in changed-letter test strings, yet that information alone was sufficient to allow their subjects to classify 59% of the test strings correctly.

In conclusion, the AGL studies do not provide any compelling evidence in favor of the grammar abstraction interpretation of the tasks. In the standard classification and recognition tasks, surface similarity measured in terms of shared supracontexts and unattested supracontexts appears to be inextricably confounded with grammaticality. The amnesic studies might appear to argue for distinct memory systems and therefore different knowledge bases for recognition versus classifica-

tion performance on test strings, but Skousen's analogical approach appears capable of modeling both amnesic and normal control behavior accurately within a single-process model. Finally, the transfer conditions also do not compel either a grammar or a prototype abstraction interpretation of AGL.

#### 4. Inflectional morphology in special populations

Ever since Pinker and his colleagues proposed their version of a dual-system approach to explaining regular versus irregular morphological processes (see especially Pinker 1991; Prasada & Pinker 1993; Pinker & Prince 1994), they and other proponents have sought to demonstrate theoretical, linguistic, developmental, experimental, and neurological dissociations confirming their dual-system model. In particular, Pinker and Prince (1994) have argued that lexical characteristics such as frequency of occurrence and word similarity would influence performance on irregular verbs but not regular verbs. The attempts to identify neuropsychological dissociations between regular and irregular verb processing fall into two groups. One uses neural imaging to try to identify different areas of brain activation as subjects process regular versus irregular forms (e.g., Jaeger et al. 1996). This research paradigm in general is encountering increasingly difficult questions about its validity (e.g., Van Orden & Paap, in press; Paap 1997). The neural imaging study by Jaeger et al. has been criticized on both theoretical and methodological grounds as well as for the authors' interpretation of their results (see Chandler & Skousen 1997; Seidenberg & Hoeffman 1998).<sup>12</sup> The other neuropsychological approach to the study of morphological processing has been to compare the performance of subjects manifesting a variety of aphasias or other neurological impairments (e.g., Ullman et al. 1997; Hagiwara et al. 1999). These latter studies adopt much the same logic applied by Knowlton and Squire to their studies of amnesic performance on the AGL task as discussed in the previous section. In the remainder of this section, I examine the study by Ullman et al. (1997) in some detail and compare their results to the predictions of an AM simulation using the same test items and memory variables.

Ullman et al. (1997) reported on the ability of subjects with a variety of neurological impairments to produce orally past tense forms for 16 irregular English verbs, 20 regular verbs, and 20 'novel' or nonce verbs designed explicitly to elicit regular past tense endings rather than irregular forms.<sup>13</sup> Figure 3 shows the overall performance in percentage correct reported by Ullman et al. for the three verb sets as produced by patients diagnosed with Alzheimer's disease (AD), posterior aphasia (PA), Parkinson's disease (PD), anterior aphasia (AA), and Huntington's disease (HD) as well as the performance of matched sets of normal controls (which



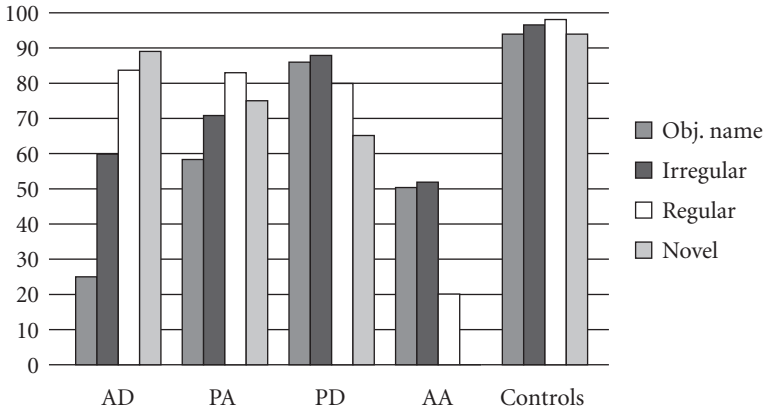


Figure 3.

I show collectively in Figure 3). The first bar for each group of subjects is the group score for an “object [picture] naming task”. The authors took this score as a rough measure of the subjects’ lexical memory, as will I in the AM simulations described below.

The central argument of the study by Ullman et al. (1997) is that collectively the subject groups show strong dissociations between their performance on regular verbs versus irregular verbs and that their performance on irregular verbs appears to be linked to their lexical memory, whereas their performance on the regular and novel-regular verbs does not. Note though that the dual-system model makes no quantitative predictions about performance on either individual verbs or on categories of verbs beyond a general dissociation between these two major verb categories. Finally, the authors correlate the differences in performance to impairments in brain areas thought to underlie declarative (and lexical) memory versus procedural (rule usage) memory.

In the AM simulations and comparisons that follow, I have chosen not to include the AA patients reported by Ullman et al. This may appear disingenuous on my part because those data appear to provide some of the strongest evidence for the dissociation claimed by Ullman et al. My reasons, however, are briefly these. The task used with the AA patients was different. Whereas all other subjects heard and saw the present (stem) form of a verb and then supplied the past-tense form orally to fill in the blank in a contextualizing sentence, the AA patients saw the past tense forms printed and simply read them aloud. As has been well attested elsewhere, anterior aphasics (more familiarly, Broca’s aphasics) often omit the suffixes from words that they are reading aloud (e.g., Marin, Saffran, & Schwartz 1976). Moreover, there is credible evidence that this difficulty is confounded with both phonological and semantic variables (e.g., cue validity, or information value, in

a given context) (e.g., Bates, Wulfeck, & MacWhinney 1991) and may represent a reading error at least as much as an underlying linguistic deficiency, a reading miscue in which the aphasic subjects simply pronounce the shortest recognizable word within a letter string (see Chandler 1993 for further discussion of this point). Indeed, by far the greatest error committed by the AA subjects in the study by Ullman et al. was to read 33% of the regular verbs as “unmarked”, that is without a suffix (for another 6% they read the suffix as *-ing*, a much more frequent verb suffix in English). Putting the unmarked responses back into the data would bring the AA performance up (perhaps only coincidentally) to almost exactly the same level as their object naming score and irregular verb score. I will return to this issue of unmarked responses after describing the performance of the other subject groups.

In the AM simulations used for the comparisons that follow, I used a data set based on all monosyllabic verbs appearing in Francis & Kučera 1982 (with the frequency values augmented by one) plus as many additional monosyllabic verbs as I could identify to arrive at a nearly exhaustive list of 1,617 monosyllabic English verbs.<sup>14</sup> For expediency's sake, I compared the written (i.e., spelled) forms of the words, although earlier work by me (Chandler 1998) showed phonological comparisons to provide more accurate simulations for oral responses than spelling comparisons do. For each subject group, I set the imperfect memory value in AM to the score reported by Ullman et al. for that group's object naming task. Finally, in comparing the test items to the AM data set, I have used only exact matches for word length. This practice is probably questionable in a strict sense (i.e., close matches for length such as *swing* and *sing* probably should influence one another), but possibly correct for a broader view. In their multivariate analysis of factors contributing to correct performance on the AGL task, Johnstone and Shanks (1999) identified overall item length (in terms of number of letters) as a significant variable. How items of different length ought to be compared in AM is an empirical issue yet to be answered.

Figure 4 presents the results for the irregular English verbs reported for the different subject groups (except for AA) in Ullman et al. (1997) compared with the results of an AM simulation in which imperfect memory is matched to the appropriate lexical memory score (object naming score). In AM predictions of past tense forms, including the test item itself in the data set virtually guarantees correct usage because of the heterogeneity constraint on generating the analogical set. On the other hand, not including the test item in the data set leads to 13 out of the 16 irregular verbs used by Ullman et al. being regularized, including the most frequent verbs such as *make*, *come*, and *give* (which have no near neighbors to motivate the correct analogies for them). Thus, the real issue in performance on irregular verbs appears to be the probability of that verb being remembered, which is to say that it is indeed a matter of lexical memory as Ullman et al. claim. Taking item frequency as a reasonable approximation to the likelihood of a given verb being remembered,

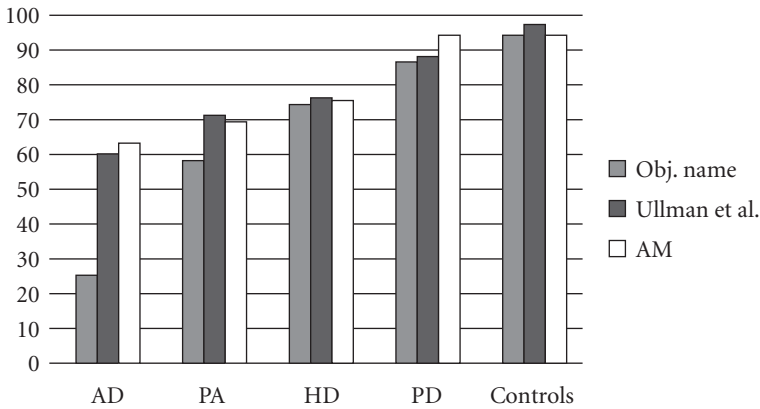


Figure 4.

I divided the 16 test items into groups based on their occurrence in word (not verb) frequency multiples of 1,000 – that is, the 1,000 most frequent words, second thousand, third thousand, etc.

In order to simulate the performance of the AD patients on irregular verbs, I set the imperfect memory factor at 0.25 to match the corresponding object naming score obtained for that group. I implemented the memory factor in the simulation by adjusting all Francis and Kučera frequency counts to one fourth their cited values and using only every fourth word of frequency value one. Assuming that the subjects remembered at least the thousand most frequent words means that they would remember, and therefore inflect correctly, the 9 most frequent verbs out of the 16 test items, plus they would have a 62% chance of inflecting *bend* correctly even if they did not remember it. This predicts a score of 10 out of 16 items inflected correctly or the 63% shown in Figure 4. To simulate the PA group, who scored 58% on the object naming task, I set the AM memory factor at 0.50 and included the second thousand most frequent words in the ‘remembered’ set. That combination predicted a score of 10 out of the 16 test verbs being remembered plus *bend* being analogized correctly for a simulated score of 69% correct. For the HD group, with a 74% object naming score, I set the AM memory factor at 75% and included the next thousand most frequent words in the ‘remembered’ set, yielding a prediction of 12 verbs correct, or 75%. Finally, for the PD group, with an 84% object naming score, and for the control groups, with a composite 94% object naming score, I included the next thousand words (all test verbs except *wring*, which at 0.000044 of the dataset falls well beyond the 6,000 most frequent words), yielding a predicted score of 94% correct.

Based on the data represented in Figure 4, lexical memory appears to track the performance on the irregular verbs closely, which is noncontroversial and is what

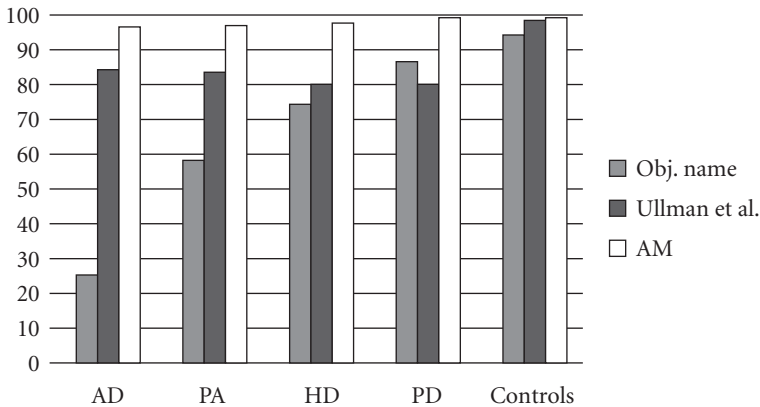


Figure 5.

we would expect and what both kinds of models predict. Figure 5, on the other hand, shows the subject performance on regular verbs compared with the performance predicted for the respective object naming (lexical memory) scores. In order to provide the strongest test of the model, I assumed that each test verb was not remembered as part of the dataset (again, ‘remembering’ a verb virtually guarantees that it will be inflected correctly). This means that all inflections were derived analogically from the comparison to the dataset, not retrieved directly from memory. Ullman et al. argued that the dissociation that they found between irregular and regular verbs (linked to object naming scores) and regular verb performance (independent of object naming) demonstrated the operation of separate morphological systems. However, as we see in Figure 5, the AM approach also predicts only very minor differences between performance on regular verbs at 25% memory, 96.35% correct, and 100% memory (except for the test item itself), 98.90% correct. Thus, performance on regular verbs is not linked significantly with lexical memory in AM. The four groups of neurologically impaired patients all scored consistently below the control group and AM predictions across the board, but for every group the ‘unmarked’ (stem form) and ‘no response’ errors account for virtually all the difference between the test groups and the control/AM groups. Thus, the AM approach models within a single process system the dissociations observed in performance between regular versus irregular verbs.

Figure 6 shows the respective comparisons for performance on the “novel” verbs, designed by Ullman et al. to solicit regular inflections (in fact, they elicit only about 90% regular inflections in the AM). Figure 6 shows a considerably less good looking match between subject performance and AM predictions. The simulation predictions are much less variable, ranging from 89.15% regular at 25% memory to 89.55% regular at 100% memory (the AM predictions at 25% and 50% memory

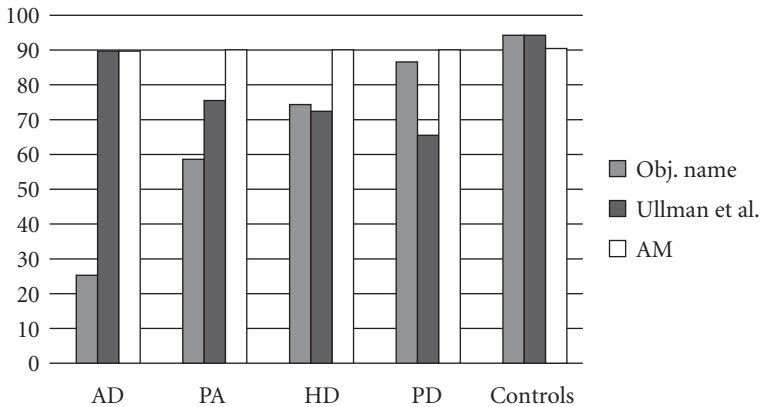


Figure 6.

each include two verbs that show close to a 50–50 split between regular and irregular inflection). The PA group errors include 13% “unmarked” verbs and the HD group 11% “unmarked”. AM, of course, does not predict “unmarked” or “no response” data, which presumably occur for other, unknown, reasons. AM, after all, is not a complete model of speech production. Moreover, except for the AA subjects, all of the other subjects in Ullman et al.’s study produced essentially the same number of “unmarked” and “no response” errors for irregular verbs as for regular verbs. So the difficulty does not appear to lie in some fundamental difference in the ease or difficulty of producing past tense forms for one type of verb over another. Finally, the PD group produced 7% “unmarked” errors and 24% “other” errors. Thus, again, virtually all the variance seen between the test groups and the AM predictions appears to be due to other factors outside the scope of either model. The most important finding, though, is that in no case does regular inflection performance appear to be directly linked to lexical memory even though AM predicts performance based on frequency of verb usage and similarity to other verbs.

Although the analogical model did an excellent job of replicating the performance patterns reported in Ullman et al. (1997) for patients with various neurological impairments, the simulation evidenced, nonetheless, some problematic behaviors that need to be addressed. As noted earlier, it is not readily clear how AM ought to compare items of different length. Skousen’s AM algorithm does include procedures for normalizing words to the vowel nuclei and then comparing syllables with different numbers of consonants, but we need to ascertain better what length contributes empirically as a perceptual feature and then consider further how to incorporate comparisons among items of different lengths into the AM formalisms. A second issue is Skousen’s position that all features in a context ought to be considered equally. It does not seem empirically accurate to claim that all

positions in a letter string or phonological string are equally salient or contribute equally to analogical selection. For example, Bybee and Morder (1983) found that the ends of words such as *swing* or *drink* were more important for analogizing to new forms than were the beginnings of the words. Similarly, numerous studies of AGL have found that initial and terminal bigrams and trigrams provide important 'anchor positions' for comparing strings, more useful to subjects than similarities and differences in the middle of the strings. In my simulations reported here, the nonce form *cug* derived an analogical set 87% in favor of zero marking on analogy with *cut*, an intuitively and empirically incorrect result because the word-final *t* in *cut* is the more relevant variable. Similarly, the very high frequency *bring* incorrectly overpowered *cling*, *fling*, *sting* and *sling* (97% to 2%) as the analogical basis for *wring*.

## 5. Discussion and conclusion

In this paper, I have sought to locate Skousen's analogical approach to modeling language within the larger context of psychological models of category learning and representation. AM is an exemplar-based approach to modeling cognitive – in this case linguistic – behavior. It operates by comparing an item of interest, the given context or the probe, to remembered instances of experiences with perceptually similar items. Therefore, I sought to summarize briefly how instance-based models in general are like other models of categorization, especially prototype-connectionist models and how they are different. In particular, I wanted to review why it is incorrect for linguists not to distinguish between exemplar-based models and other nondeclarative models, again especially connectionist models. I have argued in Section 2.1 that, despite McClelland and Rumelhart's (1986a) claim to the contrary, connectionist models do not model instance effects naturally, easily or accurately, at least not without becoming themselves in effect instance-based models, especially when the instances are distributed across nonlinearly-separable categories, as is the case with English verb forms.

The next step in my exposition was to compare certain key theoretical features of four instance-based models: nearest neighbor models, Hintzman's multiple-trace model, Nosofsky's generalized context model, and Skousen's analogical model. I tried to show how and why Skousen's comparison of items in terms of supracontexts, his test for heterogeneity, and his decision rule for choosing a specific item as the basis for analogy on a given occasion afford AM a theoretical and empirical advantage over the competing models reviewed here, especially when working with nonlinearly-separable categories such as those found in natural language.

In the final two sections of this paper, I have sought to demonstrate empirically the ability of AM to model accurately categorization behavior in two experimental paradigms that are widely said to present the strongest evidence in support of dual-system processing. In the first set of demonstrations, I showed that supracontext comparisons not only predict performance on the artificial grammar learning task by normal subjects at least as well as the dual-system alternative does, but that the comparison accounts equally well for the performance of amnesic patients, data said to show a dissociation between exemplar-based comparisons and rule-based processing. AM accounts for both sets of data within a single-system approach. Finally, I showed that AM also accounts closely for a similar behavioral dissociation identified in the inflecting of regular versus irregular verbs by normal versus neurologically impaired subjects. Again AM models the dissociation very closely within a single-process system.

The success of exemplar-based models, including AM, at modeling behavior once thought to argue for a schematic knowledge base for category learning and classification behavior suggests that there are no compelling reasons for believing that our brains abstract information from our experiences and construct some sort of separate, structurally autonomous, and schematicized neuropsychological representation of categories from those experiences. In short, there is no empirical evidence that categories exist as long-term knowledge structures in our heads. So, where do the phenomena of categories and categorization come from? Collectively, the exemplar-based models imply that they arise spontaneously when a probe enters our working memory and evokes into activation those memories that share experiential features with the probe. Through some process functionally equivalent to Skousen's analogical model, our working memory arrives at an interpretation of that input probe.

However, profound such a change in views may be for psychology and linguistics, it appears to me to have much more profound implications for linguistic theory (cognitive psychology having already moved far towards accepting exemplar-based models). If categories do not exist as real structures in the brain, then there are no substantive universals such as noun or verb or clause except as those categories arise on demand during language comprehension. Linguistic usage creates grammar, not the other way around. In his 1921 book, *Language*, Edward Sapir wrote:

The fact of grammar, a universal trait of language, is simply a generalized expression of the feeling that analogous concepts and relations are most conveniently symbolized in analogous forms (p. 38).

At the dawn of a new century, analogical modeling brings us back to a new beginning for linguistics and asks us to start over in much of our thinking about what language is.

## Notes

1. Although exemplar-based and instance-based are often used as fully synonymous terms, some researchers prefer the latter because it emphasizes the representation of experiential tokens – each encounter – versus the sometimes ambiguous use of exemplar – as in McClelland and Rumelhart's usage – to mean either a type or a token representation.
2. Newmeyer (1998) objects specifically to the characterization of syntactic categories as exhibiting prototype effects. He argues that the prototypical syntactic categories alleged by others actually appear to exhibit nonlinear separability (not his term) rather than the linear separability that he mistakenly assumes well-formed categories ought to exhibit. His solution is to posit deterministic category symbols and to transfer the prototype characteristics from the category symbols to lexical items. Unfortunately, his objections are applicable neither to nonlinearly separable linguistic categories nor to exemplar-based models of categorization.
3. This is a major distinction which I failed to make in my earlier discussions of connectionist systems (Chandler 1993, 1994, 1995).
4. In their past-tense simulation Rumelhart and McClelland (1986a) avoided the linear separability problem by combining both *sing* type verbs and *swing* type verbs into the same category, and they did not attempt to differentiate them in their data analyses or subsequent discussion. Largely as a consequence of those decisions, these verbs showed the weakest performance in Rumelhart and McClelland's PDP simulation even though they showed some of the strongest prototype effects in experimental studies by Bybee and Moder (1983) and in Chandler (1998).
5. The real nature of the relationship becomes readily evident if we simply change the labels from "dog" to "mammal" and "Rover" to "dog", etc. A much more realistic simulation would be to train the network on 50 individual dogs, each with its own name and individuating characteristics, and see whether the network could both abstract the general characteristics of "dogs" and retain each individual representation.
6. Burton (1990) has described informally a theoretical basis for a model of episodically-based cognition (although it also includes a hypothesized mechanism and process for abstracting knowledge structures from perceptual experiences). He posits a mechanism for chunking the continuous perceptual stream of input into "episodes", possibly one every one or two seconds, marked off by endogenous eye blinks and triggered by sudden discontinuities in the perceptual input. Perhaps congruent with this is the finding that the number of times a person fixates on a stimulus (a given token) predicts the likelihood of that stimulus being recalled or recognized later better than does the duration of an individual eye fixation (Loftus 1972). In other words, the number of fixations is more important than the duration of a given fixation.
7. This example comes from Rob Freeman and is used with permission.
8. Some critics of exemplar-based models find them "implausible" (e.g., McClelland & Rumelhart 1986a:193), apparently because of what has come to be called the "head-filling-up problem", the notion that such models assume an unrealistic episodic memory capacity, but that strikes me as a bogus issue. Burton (1990) has suggested that episodes may



be chunked at a rate of about once a second or so. Now, a common estimate (Kuffler & Nicholls 1976) is that the neocortex contains some 28 billion neurons (admittedly perhaps only a minority of them dedicated to memory). Each of these estimated 28 billion neurons, and especially those implicated in memory processes, makes several hundred, and many several thousand, synaptic connections with other neurons (Mountcastle 1998). This works out to trillions of synaptic connections, and synapses are important information processing units, not just individual neurons. Moreover, the evidence now emerging suggests that a given synapse can participate in many different patterns of neural ensembles simultaneously because the dispersed components of a neural representation appear to be “bound” by a common frequency of neural spikes across wide regions of the cortex rather than by simple off/on values (Damasio 1990). This works in much the same way that a single telephone wire can carry and keep separate several conversations at the same time among multiple pairs of phone connections. So, how long does it take to fill up a million or a trillion simple connections at the rate of chunking information every second or so? It takes only 11 days for a million seconds to pass, but it takes about 33 years to live one billion seconds (hence 900 years for 28 billion seconds). Humans haven’t been on earth long enough yet to fill up a trillion connections at the rate of one per second, even if we were using them that way. Now, this somewhat facetious argument aside, there is in fact some evidence that does suggest, very very slightly, a brain-filling-up phenomenon in parts of the brain of very elderly patients, and in cases of neural reorganization following extensive brain injury or after hemispherectomy in the very young. In all three cases, there is subtle evidence of crowding in the function of some of the remaining neural areas (Calvin & Ojemann 1994).

9. Elman’s (1988) work with recurrent networks was an effort to address this issue, but to date it has not been applied to any of the data types discussed in this paper. As connectionist models, however, even recurrent entry networks are susceptible to the general criticisms of the approach discussed in this paper and elsewhere.

10. Strictly speaking these are not analogical sets as Skousen describes them. Since the AGL tasks never compare competing categories, there is no role for heterogeneity in deriving an analogical set. Thus, it is simply the set of training items sharing supracontexts with a test item.

11. Skousen, personal communication, has suggested that the same effect may represent a subject using the plurality decision rule to minimize loss rather than to maximize gain.

12. See also the comments on Jaeger et al. by various contributors indexed at <<http://lloyd.umich.edu/archives/info-childes/infochi/PET-fMRI>>.

13. The forms used by Ullman et al. are (a) *regular*: chop, cook, cram, cross, drop, flap, flush, look, mar, rob, rush, scour, scowl, shrug, slam, soar, stalk, stir, tug, walk; (b) *irregular*: bend, bite, cling, come, dig, drive, feed, give, keep, make, send, stand, swim, swing, think, wring; and (c) *novel* (pseudo-regular): brop, crog, cug, dotch, grush, plag, plam, pob, prap, prass, satch, scash, scur, slub, spuff, stoff, trab, traf, tunch, vask.

14. Limiting the data set to monosyllabic verbs seems reasonable in this case. In a previous study of nonce-verb inflection (Chandler 1998), the subjects did not return a single multisyllabic response in the more than 3,500 responses to the monosyllabic test verbs (except to append the syllabic regular-past allomorph).

## References

- Aha, David W., Dennis Kibler, & Marc K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Alba, Joseph W., & Lynn Hasher (1983). Is memory schematic? *Psychological Bulletin*, 93, 203–231.
- Altmann, Gerry T. M., Zoltan Dienes, & Alastair Goode (1995). On the modality independence of implicitly learned grammatical knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21, 899–912.
- Ashby, F. Gregory (1992). Multidimensional models of categorization. In F. G. Ashby (Ed.), *Multidimensional models of perception and cognition* (pp. 449–483). Hillsdale, NJ: LEA.
- Barsalou, Lawrence W. (1983). Ad hoc categories. *Memory and Cognition*, 11, 211–227.
- Barsalou, Lawrence W. (1989). Intraconcept similarity and its implications for intraconcept similarity. In S. Vosmiadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 76–121). Cambridge: Cambridge University Press.
- Barsalou, Lawrence W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, 22, 577–609.
- Bates, Elizabeth, Beverly Wulfeck, & Brian MacWhinney (1991). Cross-linguistic research in aphasia: An overview. *Brain and Language*, 41, 123–148.
- Broadbent, Donald (1987). Simple models for experimental situations. In P. Morris (Ed.), *Modelling cognition* (pp. 169–185). London: Wiley.
- Brooks, Lee (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B. Lloyd (Eds.), *Cognition and categorization* (pp. 169–211). Hillsdale, NJ: LEA.
- Brooks, Lee R., & John R. Vokey (1992). Abstract analogies and abstracted grammars: Comments on Reber (1984) and Mathews et al. (1989). *Journal of Experimental Psychology: General*, 120, 316–323.
- Buckner, Randy L. (1996). Beyond HERA: Contributions of specific prefrontal brain areas to long-term memory retrieval. *Psychonomic Bulletin & Review*, 3, 149–158.
- Buchanan, Lori, Norman R. Brown, Roberto Cabeza, & Cameron Maitson (1999). False memories and semantic lexicon arrangement. *Brain and Language*, 68, 172–177.
- Buchanan, Lori, Chris Westbury, & Curt Burgess (2001). Characterizing the neighborhood: Semantic neighborhood effects in lexical decision and naming. *Psychonomics Bulletin & Review*, 8, 531–544.
- Burton, Peter G. (1990). A search for explanation of the brain and learning: Elements of the synchronic interface between psychology and neurophysiology I: A cognitive approach to early learning. *Psychobiology*, 18, 119–161.
- Bybee, Joan L., & Carol Lynn Morder (1983). Morphological classes as natural categories. *Language*, 59, 251–270.
- Calvin, William H., & George A. Ojemann (1994). *Conversations with Neil's brain*. Reading, MA: Addison-Wesley.
- Cassidy, Kimberly Wright, & Michael H. Kelly (1991). Phonological information for grammatical category assignments. *Journal of Memory and Language*, 30, 348–369.
- Chandler, Steve (1993). Are rules and modules really necessary for explaining language? *Journal of Psycholinguistic Research*, 22, 593–606.

- Chandler, Steve (1994). An exemplar-based approach to language acquisition. Paper presented to the Workshop on Cognitive Models of Language Acquisition at the University of Tilburg, April 21–23, 1994.
- Chandler, Steve (1995). Nondeclarative linguistics: Some neuropsychological perspectives. *Rivista di Linguistica*, 7, 233–247.
- Chandler, Steve (1998). Instance-based reference for past-tense verb-forms: An experimental study. Paper presented at First International Conference on the Mental Lexicon at the University of Alberta, Edmonton, Alberta, September 3–5, 1998.
- Chandler, Steve, & Royal Skousen (1997). Analogical modeling and the English past tense: A reply to Jaeger et al. 1996. <<http://humanities.byu.edu/am/>>
- Clark, Herbert H. (1970). Word associations and linguistic theory. In J. Lyons (Ed.), *New horizons in linguistics* (pp. 271–286). Baltimore, MD: Penguin Books.
- Cohen, Neal J., & Howard Eichenbaum (1993). *Memory, amnesia, and the hippocampal system*. Cambridge, MA: The MIT Press.
- Comrie, Bernard (1989). *Language universals and linguistic typology* (2nd ed.). Chicago: University of Chicago Press.
- Cost, Scott, & Steven Salzberg (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10, 57–78.
- Crick, Francis (1989). The recent excitement about neural networks. *Nature*, 337, 129–132.
- Croft, William (1990). *Typology and universals*. Cambridge: Cambridge University Press.
- Damasio, Antonio R. (1990). Synchronous activation in multiple cortical regions, a mechanism for recall. *Seminars in the Neurosciences*, 2, 287–296.
- Dienes, Zoltan (1992). Connectionist and memory-array models of artificial grammar learning. *Cognitive Science*, 16, 41–79.
- Dulany, Don E., Richard A. Carlson, & Gerald I. Dewey (1984). A case of syntactical learning and judgment: How conscious and how abstract? *Journal of Experimental Psychology: General*, 113, 541–555.
- Edelman, Gerald M. (1987). *Neural Darwinism: The theory of neuronal group selection*. New York: Basic Books.
- Elman, Jeff (1988). Finding structure in time. CRL Technical Report 8801. Center for Research in Language, University of California, San Diego.
- Estes, William K. (1976). The cognitive side of probability learning. *Psychological Review*, 83, 37–64.
- Estes, William K. (1994). *Classification and cognition*. Oxford: Oxford University Press.
- Francis, W. Nelson, & Henry Kučera (1982). *Frequency analysis of English usage: Lexicon and grammar*. Boston: Houghton Mifflin.
- Goldstone, Robert L., & Lawrence W. Barsalou (1998). Reuniting perception and conception. *Cognition*, 65, 231–262.
- Graf, Peter, Arthur P. Shimamura, & Larry R. Squire (1985). Priming across modalities and priming across category levels: Extending the domain of preserved function in amnesia. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11, 386–396.
- Graf, Peter, Larry R. Squire, & George Mandler (1984). The information that amnesic patients do not forget. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10, 164–178.

- Hagiwara, Hiroko, Yoko Sugioka, Takane Ito, Mitsuru Kawamura, & Jun-ichi Shiota (1999). Neurolinguistic evidence for rule-based nominal suffixation. *Language*, 75, 739–763.
- Hayes-Roth, Barbara, & Frederick Hayes-Roth (1977). Concept learning and the recognition and classification of exemplars. *Journal of Verbal Learning and Verbal Behavior*, 16, 321–338.
- Hintzman, Douglas L. (1986). 'Schema abstraction' in a multiple-trace memory model. *Psychological Review*, 94, 411–428.
- Hintzman, Douglas L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review*, 4, 528–551.
- Hintzman, Douglas L. (1990). Human learning and memory: Connections and dissociations. *Annual Review of Psychology*, 41, 109–139.
- Humphreys, Michael S., John D. Bain, & Ray Pike (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review*, 96, 208–233.
- Jaeger, Jeri J., Alan H. Lockwood, David L. Kemmerer, Robery D. Van Valin Jr., Brian W. Murphy, & Hanif G. Khalak (1996). A positron emission tomographic study of regular and irregular verb morphology in English. *Language*, 72, 451–497.
- Johnson, Keith, & John W. Mullenmix (Eds.). (1997). *Talker variability in speech processing*. San Diego: Academic Press.
- Johnstone, Theresa, & David R. Shanks (1999). Two mechanisms in implicit artificial grammar learning? Comment on Meulemans and Van der Linden (1997). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25, 524–531.
- Kelly, Michael H. (1996). The role of phonology in grammatical category assignments. In J. L. Morgan & K. Demuth (Eds.), *Signal to syntax: Bootstrapping from speech to grammar in early acquisition* (pp. 249–262). Mahwah, NJ: LEA.
- Klatzky, Roberta L. (1975). *Human memory: structure and processes*. San Francisco: W.H. Freeman.
- Knowlton, Barbara (1997). Declarative and nondeclarative knowledge: Insights from cognitive neuroscience. In K. Lamberts & D. Shanks (Eds.), *Knowledge, concepts and categories* (pp. 215–246). Cambridge, MA: MIT Press.
- Knowlton, Barbara J., Seth J. Ramus, & Larry R. Squire (1992). Intact artificial grammar learning in amnesia: Dissociations of classification learning and explicit memory for specific instances. *Psychological Science*, 3, 172–179.
- Knowlton, Barbara J., & Larry R. Squire (1993). The learning of categories: Parallel brain systems for item memory and category knowledge. *Science*, 262, 1747–1749.
- Knowlton, Barbara J., & Larry R. Squire (1994). The information acquired during artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20, 79–91.
- Knowlton, Barbara J., & Larry R. Squire (1996). Artificial grammar learning depends on implicit acquisition of both abstract and exemplar-specific information. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22, 169–181.
- Kolers, Paul A., & Henry L. Roediger (1984). Procedures of mind. *Journal of Verbal Learning and Verbal Behavior*, 23, 425–449.
- Kruschke, John K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, 28, 43–67.

- Kuffler, Steven W., & John G. Nicholls (1976). *From neuron to brain*. Sunderland, MA: Sinauer.
- Lakoff, George (1987). *Women, fire, and dangerous things*. Chicago: University of Chicago Press.
- Lindsay, Peter H., & Donald A. Norman (1977). *Human information processing* (3rd ed.). New York: Academic Press.
- Logan, Gordon D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95, 492–527.
- Marin, Oscar S. M., Eleanor M. Saffran, & Myrna F. Schwartz (1976). Dissociations of language in aphasia: Implications for normal function. *Annals of the New York Academy of Sciences*, 280, 868–884.
- Mathews, Robert C., Ray R. Buss, William B. Stanley, Fredda Blanchard-Fields, Jeung Ryeul Cho, & Barry Druhan (1989). Role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15, 1083–1100.
- McClelland, James L., & Jeff L. Elman (1986). Interactive processes in speech perception: The TRACE model. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Psychological and biological models* (pp. 58–121). Cambridge, MA: MIT Press.
- McClelland, James L., & David E. Rumelhart (1986a). A distributed model of human learning and memory. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Psychological and biological models* (pp. 170–215). Cambridge, MA: MIT Press.
- McClelland, James, & David Rumelhart (Eds.). (1986b). *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Psychological and biological models*. Cambridge, MA: MIT Press.
- McKinley, Steven C., & Robert M. Nosofsky (1995). Investigations of exemplar and decision bound models in large, ill-defined category structures. *Journal of Experimental Psychology: Human Perception and Performance*, 21, 128–148.
- McLaren, I. P. L., Helen Kaye, & N. J. Mackintosh (1989). An associative theory of the representation of stimuli: Applications to perceptual learning and latent inhibition. In R. G. M. Morris (Ed.), *Parallel distributed processing, implications for psychology and neurobiology* (pp. 102–130). Oxford: Clarendon Press.
- Medin, Douglas L., & Marguerite M. Schaffer (1978). Context theory of classification learning. *Psychological Review*, 85, 207–238.
- Medin, Douglas L., & Paula J. Schwanenflugel (1981). Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 355–368.
- Messick, Samuel J., & Charles M. Solley (1957). Probability learning in children: Some exploratory studies. *The Journal of Genetic Psychology*, 90, 23–32.
- Meulemans, Thierry, & Martial Van der Linden (1997). Associative chunk strength in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23, 1007–1028.
- Miller, George A., & Philip N. Johnson-Laird (1976). *Language and perception*. Cambridge, MA: Harvard University Press.

- Moscovitch, Morris, & Carlo Umiltà (1991). Conscious and unconscious aspects of memory: A neuropsychological framework of modules and central systems. In R. G. Lister & H. J. Weingartner (Eds.), *Perspectives on cognitive neuroscience* (pp. 229–266). Oxford: Oxford University Press.
- Mountcastle, Vernon B. (1998). Brain science at the century's ebb. *Daedalus* (Spring), 1–35.
- Nakisa, Ramin Charles, & Ulrike Hahn (1996). Where defaults don't help: The case of the German plural system. In G. W. Cottrell (Ed.), *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 177–182). Mahwah, NJ: LEA.
- Neimark, E. D., & W. K. Estes (Eds.). (1967). *Stimulus sampling theory*. San Francisco: Holden-Day.
- Nelson, Douglas L., Cathy L. McEvoy, & Thomas A. Schreiber (1994). The University of South Florida word association, rhyme and fragment norms. Manuscript.
- Newmeyer, Frederick J. (1998). *Language form and language function*. Cambridge, MA: MIT Press.
- Nosofsky, Robert M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115, 39–57.
- Nosofsky, Robert M. (1990). Tests of an exemplar model for relating perceptual classification and recognition memory (Research Report 12). Indiana University Department of Cognitive Science.
- Nosofsky, Robert M. (1992). Similarity scaling and cognitive process models. *Annual Review of Psychology*, 43, 25–53.
- Nosofsky, Robert M., & Thomas J. Palmeri (1997). An exemplar-based random walk model of speeded classification. *Psychological Review*, 104, 266–300.
- Nosofsky, Robert M., & Safa R. Zaki (1998). Dissociations between categorization and recognition in amnesic and normal individuals: An exemplar-based interpretation. *Psychological Science*, 9, 247–255.
- Nyberg, Lars, Roberto Cabeza, & Endel Tulving (1996). PET studies of encoding and retrieval: The HERA model. *Psychonomic Bulletin & Review*, 3, 135–148.
- Paap, Kenneth R. (1997). Functional neuroimages do not constrain cognitive models of language processing. Paper presented to the 22nd Annual Interdisciplinary Conference, Jackson Hole, Wyoming, February 4, 1997.  
<<http://www.u.arizona.edu/~kforster/psyling/brain2.htm>>
- Perruchet, Pierre (1994). Defining the knowledge units of a synthetic language: Comment on Vokey and Brooks (1992). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20, 223–228.
- Perruchet, Pierre, & C. Pacteau (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General*, 119, 264–275.
- Peters, Julia (1999). Native speaker awareness of phonological patterns in nouns and verbs. Paper presented at the Twenty-Sixth LACUS Forum, Edmonton, Alberta, August 3–7, 1999.
- Pinker, Steven (1991). Rules of language. *Science*, 253, 530–535.
- Pinker, Steven, & Alan Prince (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28, 73–193.

- Pinker, Steven, & Alan Prince (1994). Regular and irregular morphology and the psychological status of rules of grammar. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 321–351). Amsterdam: John Benjamins.
- Plunkett, Kim, & Virginia A. Marchman (1991). U-shaped learning and frequency effects in a multi-layered perceptron: Implications for child language acquisition. *Cognition*, 38, 43–102.
- Posner, Michael I., & Steven W. Keele (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, 77, 353–363.
- Prasada, Sandeep, & Steven Pinker (1993). Generalization of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8, 1–56.
- Reber, Arthur S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, 6, 855–863.
- Reber, Arthur S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, 118, 219–235.
- Reber, Arthur S., & Richard Allen (1978). Analogic and abstraction strategies in synthetic grammar learning. *Cognition*, 6, 189–221.
- Reber, P. J., C. E. L. Stark, & L. R. Squire (1998). Cortical areas supporting category learning identified using functional MRI. *Proceedings of the National Academy of Science, USA*, 95, 747–750.
- Redington, Martin, & Nick Chater (1996). Transfer in artificial grammar learning: A reevaluation. *Journal of Experimental Psychology: General*, 125, 123–138.
- Rosch, Eleanor (1973). On the internal structure of perceptual and semantic categories. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language* (pp. 111–144). New York: Academic Press.
- Rumelhart, David E., & James L. McClelland (Eds.). (1986a). *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 1. Foundations*. Cambridge, MA: MIT Press.
- Rumelhart, David E., & James L. McClelland (1986b). On learning the past tenses of English verbs. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Psychological and biological models* (pp. 216–271). Cambridge, MA: MIT Press.
- Sapir, Edward (1921). *Language*. New York: Harcourt, Brace & World.
- Seidenberg, Mark S., & James H. Hoeffner (1998). Discussion notes: Evaluating behavioral and neuroimaging data on past tense processes. *Language*, 74, 104–122.
- Servan-Schreiber, Emile, & John R. Anderson (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16, 592–608.
- Shanks, David R. (1995). *The psychology of associative learning*. Cambridge: Cambridge University Press.
- Shanks, David R., Theresa J. Johnstone, & Leo Staggs (1997). Abstraction processes in artificial grammar learning. *The Quarterly Journal of Experimental Psychology*, 50A, 216–252.
- Shanks, David R., & Mark F. St. John (1994). Characteristics of dissociable human learning systems. *Behavioral and Brain Sciences*, 17, 367–447.



- Shepard, Roger N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237, 1317–1323.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1997). Analogical modeling of language: Background and perspectives. Paper presented to the Round Table on Algorithms for Memory-Based Language Processing, Corsendonk, Turnhout, Belgium, December 12–13, 1997.  
<<http://humanities.byu.edu/am/am-intro.html>>
- Smith, Edward E., & Douglas L. Medin (1981). *Categories and concepts*. Cambridge, MA: Harvard University Press.
- Squire, Larry R. (1992). Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99, 195–231.
- Squire, Larry R., & Barbara J. Knowlton (1995). Learning about categories in the absence of memory. *Proceedings of the National Academy of Science, USA*, 92, 12470–12474.
- Taylor, John R. (1995). *Linguistic categorization prototypes in linguistic theory* (2nd ed.). Oxford: Clarendon Press.
- Theios, J., & J. G. Muise (1977). The word identification process in reading. In N. J. Castellan, Jr., D. B. Pisoni, & G. R. Potts (Eds.), *Cognitive theory* (Vol. 2, pp. 289–320). Hillsdale, NJ: LEA.
- Tulving, Endel (1983). *Elements of episodic memory*. Oxford: Oxford University Press.
- Tversky, Amos, & J. Wesley Hutchinson (1986). Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93, 3–22.
- Ullman, Michael T., Suzanne Corkin, Marie Coppola, Gregory Hickok, John H. Growdon, Walter J. Koroshetz, & Steven Pinker (1997). A neural dissociation within language: Evidence that the mental dictionary is part of declarative memory, and that grammatical rules are processed by the procedural system. *Journal of Cognitive Neuroscience*, 9, 266–276.
- Van Orden, Guy C., & Kenneth R. Paap (in press). Functional neuroimages fail to discover pieces of the mind in parts of the brain. *Philosophy of Science*.  
<<http://www.u.arizona.edu/~kforster/psyling/pet.htm>>
- Vokey, John R., & Lee R. Brooks (1992). Saliency of item in learning artificial grammars. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18, 328–344.
- Whittlesea, Bruce W. A. (1983). Representation and generalization of concepts: The abstractive and episodic perspectives evaluated. Ph.D. dissertation, MacMaster University.
- Whittlesea, Bruce W. A. (1987). Preservation of specific experiences in the representation of general knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13, 3–17.
- Whittlesea, Bruce W. A., & Michael D. Dorken (1993). Incidentally, things in general are particularly determined: An episodic-processing account of implicit learning. *Journal of Experimental Psychology: General*, 122, 227–248.
- Zimmer-Hart, Charles L., & Robert A. Rescorla (1974). Extinction of Pavlovian conditioned inhibition. *Journal of Comparative and Physiological Psychology*, 86, 837–845.





PART III

**Applications to specific languages**



## CHAPTER 4

# Applying Analogical Modeling to the German plural\*

Douglas J. Wulf

### 1. Introduction

As any first-year student of German will attest, it is difficult to perceive a comprehensive systematic relationship between German singular nouns and their corresponding plural forms. We may certainly locate islands of predictability, though it is not apparent how the relationship might be characterized overall. The problem is obvious upon consideration of even a small sample of German singular and plural forms, as shown in Figure 1.

As is evident from even this short list of nouns, German plural formation is distinguished by a wide variety of plural morphology and a high degree of idiosyncrasy in the distribution of this morphology across the lexicon.

We may identify three major morphological processes in the formation of the vast majority of German plurals. First of all, the plural may be indicated by *umlauting*. For example, the difference between the singular noun *Vater* and its plural *Väter* is indicated only by fronting the back vowel of the singular form. Indeed, this plural morphology also occurs in a handful of English nouns (e.g. *goose/geese*). Unfortunately, an umlauted vowel is not a reliable marker of plurality in German. Note that the singular nouns *Rücken* and *Bär* also contain umlauted vowels. Secondly, *suffixation* is employed to form the plural. A variety of suffixes are used (e.g. *Tag/Tage*, *Motor/Motoren*, *Bild/Bilder*) and certain suffixes may potentially co-occur with umlauting (e.g. *Gast/Gäste*, *Mund/Münder*). Lastly, a relatively small number of German plurals involve the *replacement* of one or more phonemes (e.g. *Ministerium/Ministerien*). However, as an additional complication, there is a significant number of plurals that involve none of these three options so that the singular and plural pairs are actually identical (e.g. *Rücken/Rücken*, *Berater/Berater*). Such an identity mapping from singular to plural has sometimes been treated formally as the addition of a null suffix (-Ø). Although such identity plurals occur occasionally in English (e.g. *deer/deer*, *fish/fish*), they are far more common in German.

Gloss	Singular	Plural	Plural morphology
“back”	Rücken	Rücken	- Ø
“adviser”	Berater	Berater	- Ø
“father”	Vater	Väter	“- Ø
“farmer”	Bauer	Bauern	-n
“bear”	Bär	Bären	-en
“motor”	Motor	Motoren	-en
“gate”	Tor	Tore	-e
“day”	Tag	Tage	-e
“guest”	Gast	Gäste	“-e
“ministry”	Ministerium	Ministerien	-um → -en
“picture”	Bild	Bilder	-er
“ribbon”	Band	Bänder	“-er
“bond”	Band	Bande	-e
“volume”	Band	Bände	“-e
“band”	Band	Bands	-s
“dog”	Hund	Hunde	-e
“association”	Bund	Bünde	“-e
“mouth”	Mund	Münder	“-er
“guardian”	Vormund	Vormunde or Vormünder	-e or “-er

Figure 1. Sample of German singular and plural forms

Besides the phonological form of the singular noun, the choice of plural morphology is also influenced by a number of other considerations. For example, the plural system in German interacts with the similarly idiosyncratic system of grammatical gender (i.e. nouns in German may be masculine, feminine, or neuter). Thus, a masculine or neuter noun ending in *-er* typically has a plural form identical to the singular, whereas a feminine noun ending in *-er* typically takes the suffix *-n* in the plural. Furthermore, semantics certainly plays some role in plural formation. For example, the singular form *Band* has four possible plural forms (*Bänder*, *Bande*, *Bände*, *Bands*) which correspond to the four separate meanings of the noun *Band* (ribbon, bond, volume, band). In addition, although rarely discussed in the literature, there is a significant number of singular nouns for which the intuitions of native speakers vacillate between two or more possible plural forms. When native speakers are asked to supply the plural of *Vormund*, a noun which occurs in the plural rather infrequently, there is often uncertainty between *Vormunde* and *Vormünder*. Indeed, many dictionaries list both of these as acceptable alternatives (e.g. *Duden* 1989: 1684).

In much the same position as students of the language, linguistic scholars have been frustrated in attempting to dissect the German plural system adequately. In

this paper, I discuss the results of my investigation of this problem under analogical modeling (AM), as advanced by Skousen (1989).

## 2. Traditional approaches to the German plural

Before discussing the analogical treatment of the German plural, it is helpful to provide some background. Many linguists have long assumed the reasoning of Chomsky and Halle that “regular variations are not matters for the lexicon, which should contain only idiosyncratic items” (Chomsky & Halle 1968:12). This picture of mental processes underlying language production has greatly shaped both linguistic study and language instruction. As a result, inflectional morphology has traditionally been modeled as assembling morphemes and applying transformational rules to make adjustments to the form. In German, we might therefore hope to define a rule to umlaut the internal vowel where necessary. We would want to predict the plural of *Tag* as *Tage*, but the plural of *Gast* not as \**Gaste*, but rather *Gäste*. However, devising an umlauting rule and the other necessary rules for such an account has proven extremely problematic. Indeed, there is good evidence from historical linguistics that determining an umlauting rule is actually impossible. We have a clear picture of the historical evolution of German inflectional morphology and this historical perspective is very telling. What we discover is that the distribution of umlaut in the plural is fundamentally random.

Old High German had a variety of plural-marking suffixes on nouns. These were inherited ultimately from old Proto-Indo-European theme vowels found on nouns in both their singular and plural forms. However, in Proto-Germanic, the movable stress of Proto-Indo-European became fixed on the root syllable. This stabilizing of the stress accent caused a progressive sloughing off of phonetic elements in final position (Waterman 1976:23). The various old theme vowels were lost on singular forms and were therefore eventually reinterpreted as plural-marking suffixes. Old High German had the *i*-theme plural ending on *gást/gásti* (guest/guests) but the *a*-theme ending on *tag/taga* (day/days).

Meanwhile, beginning in Old High German and continuing into Middle High German, a vowel harmony (umlauting) rule came into effect. The result of this rule for plural nouns was to umlaut plural nouns which happened to be marked with *i*-theme vowels, while leaving nouns marked with other theme vowels untouched. Therefore, the Old High German plural form *gásti* became *gésti*, whereas the vowel in *taga* did not change.

Finally, in the Middle High German period, the fixed stress continued its influence by reducing all the old theme vowel endings to schwa (e.g. *gésti* → *géstə* and *taga* → *tagə*). As a result, the difference in plural suffixes which had been the

original phonological motivation for the umlauting rule was obliterated. The plural of the modern German noun *Gast* therefore inherited an umlauted vowel while the plural of *Tag* did not, though the plurals now share the same plural suffix (i.e. a schwa). We discover that since its original phonological motivation was lost, the distribution of umlaut in Modern German plural forms is truly random in a fundamental way. Each plural form is etymologically based on its earlier form handed down through the generations. The appearance of umlauted vowels in German plurals is largely an accident of history.

Although we might thus plausibly conclude that an umlauting rule is not possible to formulate, linguists working under a variety of frameworks have continued to try over the years to devise such a rule. The existence of an undiscovered umlauting rule has been assumed a priori since speakers of German obviously use noun plurals productively – that is, when German speakers are presented with an unfamiliar singular noun or a nonce (i.e. invented) singular form, speakers have intuitions about how such a word may be pluralized, including the possibility of umlauting.

A synchronic system of German pluralization thus seems to exist, though it eludes easy description under a traditional, rule-based framework. As Salmons (1994:213) writes,

By any standard, the German plural system is highly marked and its connection to umlaut especially problematic . . . The role of umlaut in German morphology is . . . challenging, bringing forth a range of proposals . . . Morphological uses of umlaut have wrought havoc in such prominent theoretical frameworks as Lexical Phonology / Lexical Morphology . . . This indicates unambiguously that German plural marking and German morphological uses of umlaut represent particularly difficult phenomena for linguistic analysis.

Because of such challenges, many traditional accounts amount to little more than taxonomies. Essentially, whatever plural morphology a particular nominal requires is just satisfied by assigning it to a class whose principle characteristic is that it supplies exactly that plural morphology. Such analyses do not reliably explain why a particular noun might fall into one or another of the classes. Therefore, there is no ability to predict the plural form given a novel or nonce word without first being told the class of that word, which is tantamount to being told the plural form anyway. If such categories exist, how is it that speakers formulate categories and assign novel/nonce forms to them without being told?

The inability of traditional linguistic approaches to yield a satisfactory account of German plurals has led to a growing interest in treating this topic under a computational analysis. Expressed in the terminology used by computer scientists, we may say that although traditional accounts may boast *coverage* of the data (claiming that each word in the lexicon is assigned to some morphological category),

they have little or no *robustness* (the ability to handle novel or unexpected data). As Nakisa and Hahn (1996:177) write,

The German plural system has become a focal point for conflicting theories of language, both linguistic and cognitive . . . What is now required is the development of explicit computational models which allow quantitative assessment against real data.

### 3. Computational approaches to the German plural

A recent comparison of computational treatments addressing the German plural is advanced by Nakisa and Hahn (1996). In their study, they test three simple approaches to predicting the plural form of a German noun given the singular: an ordinary nearest-neighbor algorithm, the Generalized Context Model (GCM) proposed by Nosofsky (1990), and a standard, three-layer back-propagation network. The measures of coverage they have attained with these simple models is noteworthy.

Nakisa and Hahn drew their data from the 30,100 German nouns in the CELEX database. Plural categories with a type frequency of less than 0.1% were discarded, resulting in a database of 24,640 nouns with 15 possible plural forms. In testing this entire set of nouns with the simple nearest neighbor algorithm, the predictive accuracy was 72% on novel items. They then tested a subset of 8,598 non-compound nouns, defined as any noun “that did not contain another noun from the database as its rightmost lexeme” (1996: 177). Splitting this subset roughly in half with a training set of 4,273 words and a testing set of 4,325 words, the nearest neighbor approach still maintained a predictive accuracy of 71%. Using this same subset, the GCM approach scored 75% and the network scored 83.5% on novel items.

Given this level of success with other computational models, I was interested in contrasting this performance with Skousen’s AM approach. Although my ultimate aim was a careful, side-by-side comparison with the results obtained by Nakisa and Hahn, my immediate aims were more modest. In Wulf 1996a and 1996b, I noted that AM is quite successful in predicting the German plural when given a dataset numbered in only hundreds of examples rather than thousands. Thus, I discovered that the AM approach was robust and offered good coverage even with a dataset of examples drastically smaller than the training sets used for many other computational models. However, in focusing on AM’s performance with particular lexical items, my earlier studies had not calculated an overall quantitative measure of performance with a small dataset, so this is what I set out to measure.



#### 4. Setting up the analogical approach to the German plural

In my most recent investigations of the German plural, I have used the updated analogical modeling program in Perl (aml10) available for download from the AM homepage and have run these on a Macintosh. The program itself contains no rules of German plural formation, but merely compares an input form with those in the dataset and, according to the principles of analogical modeling, as outlined in Skousen 1989, makes an “educated guess” at its plural form.

The datasets and test items are drawn from the CELEX database. Three examples of dataset entries are shown here:

```
F fb==@-====ts=i-====U+==NC Beziehung/Beziehungen 1245/1061
E fm==i-====n==u+====t==@-====V Minute/Minuten 1428/1039
A n===a+====ng==@-====b==o+====tC Angebot/Angebote 1190/718
```

The dataset entries are specified in three fields, separated by spaces. The first field is a one-letter code indicating the plural morphology taken by the noun in the plural. There are 13 possible options of plural morphology, designated with the letters A through M. The second field is twenty-six characters long and specifies first the gender of the noun (*m*, *f*, or *n*) followed by an encoding of three syllables of the word (discussed below) and then followed by an indication of whether the final phoneme is a consonant or a vowel (*C* or *V*). The third field is a comment field ignored by the AM program. This simply cites (for a human reader) the singular and plural forms of the noun as well as the noun’s lemma frequency and the plural form’s frequency from the CELEX database. The test items are identical to the dataset items except that the first field indicating the plural morphology has been removed, as this is what the program attempts to predict.

The codes for the various options of plural morphology are shown in Figure 2. Also indicated here are the percentages (rounded values) of words from the CELEX database which take each particular option (as cited in Nakisa & Hahn 1996: 179).

These options are the same as those used by Nakisa and Hahn (1996) with two exceptions. Because of the small size of the datasets used in the tests, there happens to be no dataset or test item which forms the plural by replacing *-um* with *-a* (approximately 0.5% of the words in the CELEX database), so this possibility is excluded. Secondly, I group the morphology associated with such lexical items as *Thema/Themen* under option E in Figure 2. This is because the final *-a* in the singular form is pronounced as a schwa and the ending *-en* is pronounced as a schwa plus the consonant *n*. Thus, it seems reasonable to me that the distinction may be largely one of spelling convention rather than phonology, though this point is debatable.

Memory limitations allow only a maximum of approximately 26 variables to be compared. The description for each word in the data is therefore limited, and it

Outcome		Example	% in CELEX Database
A	-e	<i>Jahr/Jahre</i>	18
B	- <i>¨e</i>	<i>Kraft/Kräfte</i>	8
C	-er	<i>Kind/Kinder</i>	1
D	- <i>¨er</i>	<i>Land/Länder</i>	3
E	-n	<i>Frage/Fragen</i>	18
F	-en	<i>Mensch/Menschen</i>	28
G	-	<i>Leben/Leben</i>	17
H	- <i>¨</i>	<i>Schaden/Schäden</i>	1
I	-s	<i>Auto/Autos</i>	4
J	-ten	<i>Bau/Bauten</i>	0.1
K	-ien	<i>Prinzip/Prinzipien</i>	0.1
L	-is → -en	<i>Thesis/Thesen</i>	0.4
M	-um → -en	<i>Datum/Daten</i>	0.6

Figure 2. Possible outcomes for plural morphology

has been necessary to decide what information should be entered and what withheld. Nakisa and Hahn (1996) do not include gender in their tests, but since the plural system obviously interacts with the gender system, I allocate one variable to its specification, as I had done in my previous studies. Since the suffixation morphology is broadly influenced by whether the word-final phoneme is a consonant or a vowel, I devote one variable to this distinction as well. This leaves 24 variables for the specification of three syllables of eight variables each: three for the onset consonants, two for the vowel, and three for the coda consonants. Onset and coda consonant variable slots are zero-padded (with =) whenever all three slots are not filled. For words shorter than three syllables, the unused variables are likewise zero-padded. For words longer than three syllables, only the first and final two syllables are specified.

Finally, each vowel is marked as a back vowel (indicated with +) or front vowel (indicated with -). My reason for this is motivated by the way the analogical method works. This is best illustrated with an example. Suppose an AM analysis of the German plural is attempted with a dataset of only two forms *Wert* (plurality option A: *Werte*) and *Wort* (plurality option D: *Wörter*) and we wish to predict the plural of *Wirt*. For purposes of the illustration, we may ignore the contribution of gender. Without marking explicitly for front/back vowels, AM judges *Wert* and *Wort* to be equally similar to *Wirt* and predicts each plurality option 50% of the time. This is because both words in the dataset differ from the test item by just one variable each (i.e. the vowel). However, as far as the question of umlauting (vowel fronting) is concerned, *Wirt* is obviously more similar in an important phonetic

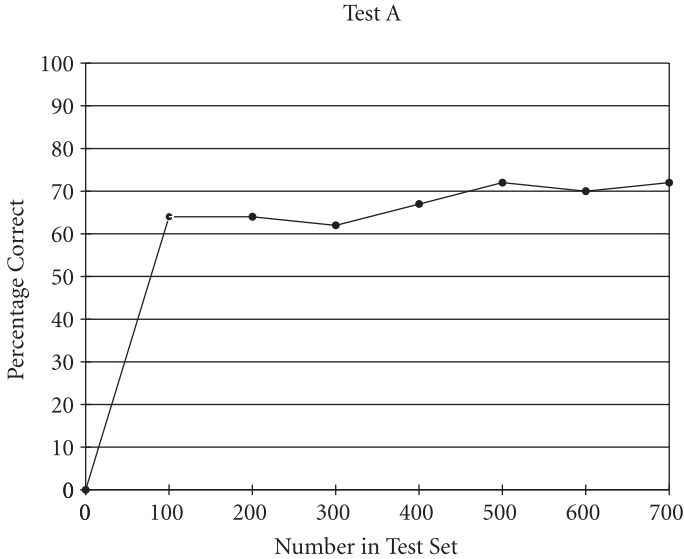
respect to *Wert*. The vowels in both of these words are front vowels, whereas *Wort* has a back vowel. If this fact is specified, AM predicts the correct plural for *Wirt* (option A: *Werte*) by analogy with *Wert* with a frequency of 100%. Therefore, it seems to me that in order to be faithful to the theory of AM, it is reasonable to cite vowel fronting explicitly.

Which words to include in a dataset is also an important issue. In theory, the most frequently used words in the language have the greatest analogical effect. For this reason, the nouns in the CELEX database were first sorted in order of the decreasing frequency of their plural forms. Unfortunately, some commonsense editing was then required in entering the nouns into datasets due to the problem of homographs. For example, the word *Bayern* may translate either as the proper noun ‘Bavaria’ or as the plural common noun ‘Bavarians.’ The frequency of one use versus the other is not specified in the CELEX database. For this current study, I therefore eliminated a small number of homographic forms from the datasets, though this is obviously not an ideal solution.

There are a number of such homographic problems, but obviously the most serious concerns the identity plurals in German. Since the singular and plural forms are identical, the wordform frequencies for such nouns are inaccurately high since they record every occurrence of this noun in the plural *and* the singular. If not adjusted in some way, the AM system would tend to overpredict the identity plural. As a reasonable approximation, since the identity plural is taken by approximately 17% of all nouns in the CELEX database, for every hundred words entered, I simply included 17 identity plurals, added in their order of frequency. Again, this was not ideal, but it at least avoids the gross distortion that occurs if the frequency numbers are used directly.

## 5. Analogical Modeling of the German plural

My study consists of two tests, which I call Test A and Test B. For Test A, I use the 800 most frequent nouns in German. Of these, 700 serve as the analogical dataset and 100 as the test set. In order to score performance, I select by plurality – that is, the most frequently selected result is taken as AM’s prediction for a trial. With the 700-word dataset, the AM approach scores 72% on this test set of 100 items. For the subsequent trials, I delete sets of 100 words successively from the dataset and run the same 100 test items against the 600 most frequent words, the 500 most frequent words, and so forth, down to only the 100 most frequent. Even with a dataset reduced to only 100 examples, AM still scores 64%. Although the scores vary somewhat up and down, the general trend is one of a gradual increase in performance as the size of the dataset is increased, as shown in Figure 3.



**Figure 3.** Results of Test A

The results are remarkably good given the small sizes of these datasets. First of all, without ever being told a morphological rule and with very few examples, the model is able to predict plurals confidently for many forms in German which behave categorically in their formation of the plural. As it so happens, plural forms which have traditionally been the easiest to characterize in terms of a rule have tended to involve forming the plural by adding *-n*, *-en*, or the identity plural (options E, F, and G). Together, these three plural options account for approximately 63% of all plural forms in German. Considering the trial of Test A with 700 words in the dataset, AM’s accuracy for only these three “regular” (i.e. largely regular) plural options is 87%.

The scores for the remaining options are somewhat lower, yet this is not particularly surprising. It is certainly the case that there are a large number of idiosyncratic choices in German plural formation, especially among the most commonly occurring nouns in the language. Most theories of German plural formation admit the notion that a greater than average number of exceptional plural forms must simply be memorized. However, AM demonstrates that if a certain number of regular forms are also memorized, rule-based behavior may be demonstrated through analogy alone. Thus, as the dataset is increased, exceptional forms are memorized and patterns of regular behavior are strengthened.

It is also interesting to note AM’s success in predicting umlauting in the plural versus suffixation/replacement. The 700-word trial of Test A scores 72% and thus

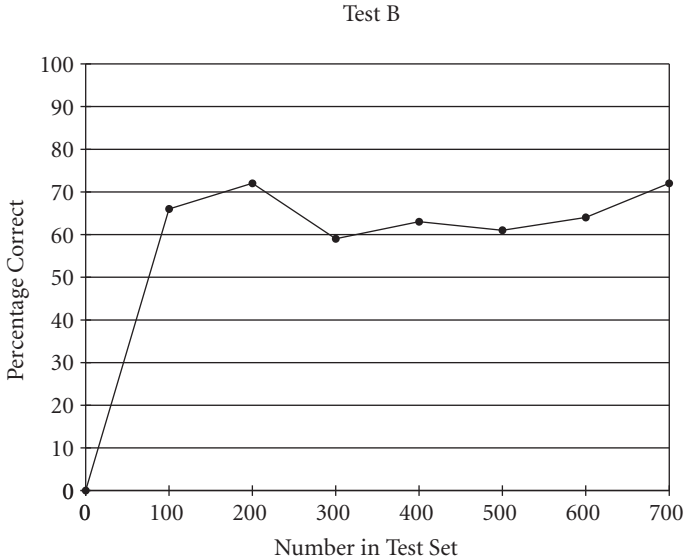


Figure 4. Results of Test B

misses the correct form for 28 examples. Of these 28 mistakes, 24 of them involve selecting an incorrect suffix/replacement, without any error in umlauting. By contrast, only three mistakes involve a correct selection of a suffix, but with incorrect umlauting. In only one example does AM select both the wrong umlauting option and the wrong suffix. Thus, for this set of one hundred examples, AM has an accuracy of 75% in selecting the correct suffix/replacement and an accuracy of 96% in predicting the umlauting of the vowel in the plural.

Of course, it is important to demonstrate that AM's performance in Test A is not simply a lucky occurrence. Test B is designed to do this. I begin with the same 800 words as in Test A. For the first trial, I use the first hundred words as a dataset and the second hundred words as a test set. Next, I use the first two hundred words as a dataset and the third hundred words as a test set. I proceed upwards in this fashion so that each trial involves a new test set of one hundred novel words. Of course, the final trial of Test B involves a 700-word dataset and is thus identical to the trial from Test A. The results of Test B are shown in Figure 4.

The performance varies up and down, yet remains fairly close in each case to the average across these seven trials (namely, 65.3%). This percentage is probably also reasonably close to the percentage of plurals in these datasets that might be characterized as "regular," in the sense that it would be fairly straightforward to account for them with a morpholexical rule.

In many cases AM performs much as a nearest neighbor approach. However, on certain occasions, AM goes well beyond what could be achieved under a straightforward nearest neighbor algorithm. For example, consider AM's predictions in Test B for the plurals of *die Steuer* ('tax') and *das Steuer* ('control') as shown in Figure 5.

#### Part 1

Project Name: german400B.data

Given Context:

f = = = = = S t = Q - = = = = @ - = = r C

S t e u e r / S t e u e r n 2 0 6 / 1 1 6

Include context even if it is in the data file

Number of data points: 400

Probability of including any one data point: 1

Total Excluded: 0

Nulls: exclude

Gang: squared

Number of active variables: 33

Number of active contexts: 8589934592

#### Statistical Summary

E	-n	10076	75.18%
F	-en	101	0.75%
G	-	3225	24.06%

#### Part 2

Project Name: german500B.data

Given Context:

n = = = = = S t = Q - = = = = @ - = = r C

S t e u e r / S t e u e r 2 0 6 /

Include context even if it is in the data file

Number of data points: 500

Probability of including any one data point: 1

Total Excluded: 0

Nulls: exclude

Gang: squared

Number of active variables: 28

Number of active contexts: 268435456

#### Statistical Summary

E	-n	96	9.21%
G	-	946	90.79%

Figure 5. Plural forms predicted for *die Steuer* and *das Steuer*

As we see in the first run, given a dataset of 400 words, AM predicts the correct plural morphology for *die Steuer* (E -n) with a frequency of about 75%. Next, AM is required to predict the plural for *das Steuer*. Because this word in the plural is slightly less frequent according to the CELEX frequency measures, the dataset for this test happens to contain 500 examples, including *die Steuer*. The nearest neighbor of *das Steuer* in the dataset is obviously *die Steuer*, so a pure nearest neighbor algorithm would incorrectly select E -n as the plural morphology of *das Steuer*. However, AM looks beyond the nearest neighbor. Even though the presence of *die Steuer* in the dataset causes some leakage, AM correctly predicts the identity plural (G -) for *das Steuer* with a frequency of about 91%.

Even when AM's predictions are at variance with the plural forms of Standard German, there is often something interesting to discover in the results. For example, in various runs of the program with small datasets, AM predicts the plural of *die Saison* as *die Saisonen* (F -en) and often with a frequency of 100%. As with many nouns borrowed from foreign languages into German, however, *Saison* takes its plural in -s (i.e. *Saisons*) in the standard language. Nevertheless, it is interesting to note that *Saisonen* is the plural form used in the dialectal variant of German spoken in Austria. The treatment of German plurals under AM gives us a possible explanation for this dialectal usage. In cases where AM fails to predict the standard form, this could provide an indication forms that might be encountered in non-standard dialects or in production errors in the usage of the standard dialect.

## 6. Goals for further research

Certainly, the next step in studying the German plural under AM is to consider performance with increasingly large datasets. Such trials have already been conducted by Daelemans (in this volume), who has discovered that, as expected, performance does climb as the dataset is raised to thousands, rather than hundreds, of examples. Of particular note, the accuracy in correctly predicting the more uncommon plural morphology in German should improve. In small datasets, the rarely encountered morphology may only be represented by a small handful of forms or perhaps even just a single form. Only in larger datasets do sufficient examples accumulate to generate small islands of analogical behavior.

In Wulf 1996a and 1996b I discuss at length the intriguing behavior of AM with an example such as *Vormund*. When *Mund* and its plural are left out of the data set, the program selects the standard plural of *Vormunde* with some slippage towards \**Vormünde*, influenced largely by the presence of *Grund/Gründe*. However, when the single entry of *Mund* (D -"er) is added to the data set, the effect on plural selection for *Vormund* is considerable. Suddenly a major variant form

in *Vormünder* appears in the output and in fact is often modeled as the most popular form.

Of course, the noun *Vormund* rarely occurs in the plural. Thus, to be in conformity with the theory of AM, this plural should be predicted from the combined analogical effect of all the many thousands of more common plural noun forms in German. We may then see if the fairly even division between *Vormunde* and *Vormünder* persists even as the dataset grows, while selection of the incorrect form *\*Vormünde* declines. When tested with only small datasets, AM does not model such exact percentages of variation in German usage. Nevertheless, the analogical model is already able to capture the variation and leakage across categories so characteristic of infrequently encountered German plurals.

The analogical method can thus give us one possible explanation for the distribution of umlaut in the German plural which has caused linguists so much difficulty. For the most common words of the language, those handed down over time, the distribution of the umlaut is essentially random. Thus, forms such as *Tage* and *Gäste* are simply examples in the data set, so to speak, which are given to us as language learners. These initial examples however, are referred back to again and again in language use when forming plurals of infrequent words, such as *Vormund*. The illusion of the application of rules results from such analogical effects.

## 7. Conclusion

Traditional rule-based systems can be helpful in summarizing language behavior, but sometimes offer little in the way of predictive power. Computational approaches to such problems as the German plural clearly do much better in making predictions. Among the variety of computational approaches under consideration, AM is certainly a promising alternative. AM accounts for the productive use of German plurals by relying on analogies with given forms to generate novel ones. The highly marked quality of the dataset results in great variety in the application of plural markers, such as the umlauted vowel. Even in cases where AM is not successful in predicting the standard form, the results may often be significant as they may correspond to dialectal forms or common production errors. The analogical approach thus indeed offers hope in disambiguating intricate linguistic patterns such as the plural in German.



## Note

\* The author wishes to thank David Graff of the Linguistic Data Consortium at the University of Pennsylvania for his invaluable assistance in sorting nouns in the CELEX database by frequency.

## References

- Chomsky, Noam, & Morris Halle (1968). *The sound pattern of English*. New York: Harper and Row.
- Duden Deutsches Universalwörterbuch* (1989). Mannheim: Dudenverlag.
- Marcus, Gary F., Ursula Brinkmann, Harald Clahsen, Richard Wiese, Andreas Woest, & Steven Pinker (1995). German inflection: the exception that proves the rule. *Cognitive Psychology*, 29, 189–256.
- Nakisa, Ramin Charles, & Ulrika Hahn (1996). Where defaults don't help: the case of the German plural system. In G. W. Cottrell (Ed.), *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 177–182). Mahwah, NJ: Lawrence Erlbaum Associates.
- Nosofsky, Robert M. (1990). Relations between exemplar-similarity and likelihood models of classification. *Journal of Mathematical Psychology*, 34, 393–418.
- Salmons, Joseph C. (1994). Umlaut and plurality in old high German: some problems with a natural morphology account. *Diachronica*, 13, 213–228.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Voyles, Joseph B. (1992). *Early Germanic grammar: pre-, proto-, and post-Germanic languages*. San Diego, CA: Academic Press.
- Waterman, John T. (1976). *A history of the German language*. Prospect Heights, IL: Waveland Press.
- Wulf, Douglas (1996a). An analogical approach to plural formation in German. In *University of Washington Working Papers in Linguistics, Vol. 14: Proceedings of the Twelfth Annual Northwest Linguistics Conference* (pp. 239–254).
- Wulf, Douglas (1996b). An account of German plural formation using an analogical computer model. In *Proceedings of the First ESSLLI Student Session at the Eighth European Summer School in Logic, Language and Information* (pp. 92–96).

## CHAPTER 5

# Testing Analogical Modeling

## The /k/ ~ ∅ alternation in Turkish\*

C. Anton Rytting

### Introduction

#### Consonantal alternations in Turkish

The final consonants of many Turkish stems display two different forms, depending on the surrounding phonological context. For example, it is very common for stem-final voiceless obstruents to become voiced when followed by a vowel-initial suffix (Lewis 1967; Sezer 1981; Inkelas & Orgun 1995):

- a. *kalıp* ‘mold’    *kalıp-lar* ‘mold-PL’    *kalıb-a* ‘mold-DAT’  
(Inkelas & Orgun 1995)
- b. *kurt* ‘worm’    *kurt-tan* ‘worm-ABL’    *kurd-u* ‘worm-3SG.POSS’  
(Sezer 1981)
- c. *a:ç* ‘tree’    *a:ç-ta* ‘tree-LOC’    *a:j-m* ‘tree-GEN’  
(Lewis 1967:31)

However, stem-final velar stops behave quite differently. Final /k/ does voice to /g/ in some borrowed words, but only after /n/:

- d. *renk* ‘color’                      *rengim* ‘color-3SG.PRED’  
*denk* ‘counterweight’            *dengi* ‘counterweight-ACC’

In all other stem-final contexts, both /k/ and /g/ either remain as they are or disappear entirely. The typical pattern is for the velar to delete intervocalically:

- e. *yatak* ‘bed’                      *yatak-lar* ‘bed-PL’                      *yata-ı* ‘bed-3SG.POSS’  
*kelebek* ‘butterfly’    *kelebek-ler* ‘butterfly-PL’    *kelebe-i* ‘butterfly-3SG.POSS’  
*filolog* ‘philologist’    *filolog-lar* ‘philologist-PL’    *filolo-u* ‘philologist-3SG.POSS’

This conditional deletion of stem-final velars is known in the literature as the /k/∼∅ alternation (Zimmer & Abbott 1978; Sezer 1981; Van Schaaik 1996; Kibre 1998) or as velar drop (Inkelas & Orgun 1995). There are some known exceptional cases to the /k/∼∅ alternation, where velar deletion does not apply. The majority of these cases fall into two categories (see Sezer 1981).<sup>1</sup>

### Exceptional class 1: Monosyllabic stems

The first class of exceptions involve monosyllabic stems of the pattern (C)VC[+velar], where both voiceless and voiced velar stops are retained intervocally (Inkelas & Orgun 1995):<sup>2</sup>

- |        |                     |                                |                                  |
|--------|---------------------|--------------------------------|----------------------------------|
| (1) a. | <i>kök</i> ‘root’   | <i>kök-e</i> ‘root-DAT’        | not * <i>kö-e</i> * <i>kög-e</i> |
|        | <i>ek</i> ‘affix’   | <i>ek-e</i> ‘affix-DAT’        | not * <i>e-e</i> * <i>eg-e</i>   |
|        | <i>ok</i> ‘arrow’   | <i>ok-um</i> ‘arrow-1SG.POSS’  | not * <i>o-um</i> * <i>og-um</i> |
|        | <i>lig</i> ‘league’ | <i>lig-i</i> ‘league-3SG.POSS’ | not * <i>li-i</i>                |
|        | <i>füg</i> ‘fugue’  | <i>füg-e</i> ‘fugue-DAT’       | not * <i>fü-e</i>                |

The same general pattern of exceptions may be observed in monosyllabic roots of type C<sub>0</sub>VC[−continuant, −velar], where the voicing does not alternate as noted above – both with voiceless (1b) and voiced (1c) final consonants (Inkelas & Orgun 1995; Kibre 1998):

- |        |                     |                           |
|--------|---------------------|---------------------------|
| (1) b. | <i>at</i> ‘horse’   | <i>at-ı</i> ‘horse-ACC’   |
|        | <i>sap</i> ‘stem’   | <i>sap-ı</i> ‘stem-ACC’   |
|        | <i>koč</i> ‘ram’    | <i>koč-u</i> ‘ram-ACC’    |
| c.     | <i>öj</i> ‘revenge’ | <i>öj-i</i> ‘revenge-ACC’ |
|        | <i>ud</i> ‘oud’     | <i>ud-u</i> ‘oud-ACC’     |

Inkelas and Orgun (1995) report that 87 percent of monosyllabic roots resist final-consonant alternations. We see, then, that this class of words has a strong tendency toward exceptionality, but it itself is not a consistent exceptionless subrule.

### Exceptional class 2: Words with a long vowel before the final /k/

It seems from (1a) and (1b) that the preservation of the final velar may just be a special case of a general resistance to alternation in monosyllables. But the situation is not that simple. There is another class of words, the vast majority of which have been borrowed from Arabic or Persian sources, that have an “underlyingly” long vowel before the /k/. This vowel is usually shortened before a consonant in the coda (as seen here in the base forms), but regains its length when the /k/ is re-syllabified before a vowel-initial suffix (Sezer 1981):

- (2) *merak* ‘curiosity’      *mera:ki* ‘curiosity-3SG.POSS’  
*infilak* ‘explosion’      *infila:ki* ‘explosion-3SG.POSS’  
*ittifak* ‘alliance’      *ittifa:ki* ‘alliance-3SG.POSS’  
*tahkik* ‘verification’      *tahki:ki* ‘verification-3SG.POSS’  
*tetikik* ‘investigation’      *tetki:ki* ‘investigation-3SG.POSS’

### Various accounts of the /k/~∅ alternation

The /k/~∅ alternation and these two classes of exceptions have been accounted for in several ways. Zimmer and Abbott (1978) and Sezer (1981) postulate a rule that deletes final /k/ after a vowel in polysyllabic words. I present a modified version here:

- (3)  $C[+velar] \rightarrow \emptyset / VC_0V[-long] \_\_ + V[+denominal, +native]$

Inkelas and Orgun (1995) essentially agree with Sezer’s analysis, but they seek to provide principled justification for it. For the first class of exceptions, Inkelas and Orgun cite a language universal proposed by McCarthy and Prince (1986), the bimoraic minimal size condition, which prevents  $C_0VC$  roots from being further shortened by rules such as velar deletion. Inkelas and Orgun’s analysis does not explicitly account for Sezer’s second class of exceptions.<sup>3</sup>

Van Schaaik (1996:113), on the other hand, seems to reject the notion of “exceptional classes” governed by rules. He argues that the alternation and all its exceptions may be most elegantly accounted for by means of lexically stored “archiphonemes” /G/ → {g,∅}, /K/ → {k,∅}, which are phonetically underspecified in the lexicon, but are realized in speech according to the phonological surroundings. All polysyllabic velar-final words are supposed to contain these archiphonemes; monosyllabic words contain fully specified /k/ or /g/. This analysis avoids the problem of finding theoretically justifiable rules to account for every word. However, it also fails to account for the productive nature of the /k/~∅ alternation. Zimmer and Abbott (1978) note that the /k/~∅ alternation applies not only to native words, but also to recently borrowed words:

- (4) *frikik* ‘freekick’      *friki-i* ‘freekick-3SG.POSS’  
(Zimmer & Abbott 1978)

Furthermore, their experimental evidence suggests that the /k/~∅ alternation is both productive and psychologically real. Zimmer and Abbott describe two psycholinguistic surveys testing the productivity of the /k/~∅ alternation, in which native speakers were asked to attach the vowel-initial suffix to various nonce-words. Their surveys included monosyllabic “stems” (Exceptional class 1), and several examples of “Arabic-sounding” stems, including one example of a long final vowel

(Exceptional class 2).<sup>4</sup> Although their results reveal some variation among speakers, they do show that most speakers tend to behave in accordance with Sezer's rule (listed as (3) above). Zimmer and Abbott report a clear correlation, significant to the .05 level, between the number of syllables and /k/ deletion. Since nonce words cannot be supposed to exist *a priori* in a speaker's lexicon, these data seem to indicate the presence of a word-independent phonological rule or tendency, contrary to Schaaik's assertion. Schaaik's strictly lexical approach therefore seems less plausible from a psycholinguistic standpoint, since it does not account for Zimmer and Abbott's data.

Sezer's rule is, in part, well-supported by Zimmer and Abbott's experimental data, and elegantly accounted for within Inkelas and Orgun's theory. But only in part. With regard to monosyllables (Exceptional class 1), it provides a benchmark by which other approaches may be measured. Any alternate theory must account for the productive velar-retention of monosyllables, as well as for the normal case. However, Sezer's second exceptional class – words with long final vowels before /k/ – has not been adequately shown to be a productive rule, either in theory or empirically. The one nonce word of this type in Zimmer and Abbott's study was judged to retain the velar by only 55% of the informants. Further data from the TELL lexicon (see below) also shows considerable variation among words of this type.

In short, there seem to be two general ways to account for these exceptional classes. One, represented by Sezer (1981) and Inkelas and Orgun (1995), proposes more fine-grained rules to handle these exceptions. The other, represented by van Schaaik (1996), assumes that these exceptions are entirely lexical in nature. We have seen that both approaches encounter some difficulty. Nicloas Kibre, discussing the application of the dual-route model to Turkish final-consonant alternations, observes this same dilemma. The exceptional classes are too regular to be listed as exceptions, yet too irregular to be fully described by rules. He proposes that only a system that recognizes a continuum between regularity and irregularity will be able to account for final consonant deletion (1998).

A number of approaches meet Kibre's requirement for such a continuum of regularity. Kibre himself suggests a schema-based model incorporating "family resemblance" and other ideas from lexical connectionism (1998). Exemplar-based approaches constitute another class of alternatives to the "rule-plus-exception-list" approach and, furthermore, have working implementations that can be empirically tested. I will here consider an exemplar-based approach known as analogical modeling (Skousen 1989). By it I will show that an exemplar-based model can account for both the productive regularity of monosyllables (Exceptional class 1) and the variation in long-voweled polysyllables (Exceptional class 2).

## Analogical Modeling of language

### A brief description of the approach

Like other exemplar-based approaches to language, analogical modeling (AM) rests on the assumption that speakers do not rely on a finite set of rules in order to perform some operation on a word, such as attaching a suffix to a word. Instead, a speaker remembers specific examples of words with that suffix attached, and bases his performance on the examples in his memory. Usually, an instance of the “target word” itself will be remembered (if it is a relatively common word), and that instance will be applied. For example, if I want to add the plural suffix to the word ‘log’, I remember (subliminally) the last time I heard the plural of ‘log’ as [lɑgz], and model my own performance after it. But supposing I should forget the plural of ‘log’. I could still come up with a good “guess” by remembering that ‘dog’ becomes [dɑgz], ‘leg’ becomes [lɛgz], etc., and applying the plural selected on the basis of these neighboring words.

However, suppose I were to forget the plural of ‘ox’. This is unlikely for an adult speaker, since ‘ox’ is not a very rare word. But if I did, I would probably remember ‘ax’ → [æksəz], ‘box’ → [bɒksəz], etc., and come up with the “wrong” plural – but the same plural one would expect from applying a rule. As this example suggests, the AM model predicts that sufficiently common exceptional words (like ‘ox’) will retain their irregular forms, but that rarely used word forms will eventually be forgotten and re-analyzed by analogy to have regular forms.

Unlike nearest neighbor approaches, which base their predictions on the *n* nearest neighbors to the target word in question, AM creates on the fly an “analogical set” of variable size for the target word in its given context, consisting of (1) classes of examples that are the most similar to the target word (nearest neighbors), and (2) more general classes of examples which behave like those nearest neighbors. This second class allows for what Skousen calls the *gang effect*: many words which are part of a larger regular pattern may be included in the analogical set – even if they are only marginally similar to the word in question – as long as there are no intervening words which behave differently. This allows the more common patterns, such as the regular plural in English, to apply to nearly any unremembered word, creating a rule-like regularity across the nominal lexicon.

In short, the probability of a word being chosen as an “exemplar” for the target form in a given context *X* depends on three properties (Skousen 1989:4):

1. the similarity of the occurrence to the given context *X*;
2. the frequency of the occurrence;
3. whether or not there are intervening occurrences closer to *X* with the same behavior.

Unlike both rule approaches and connectionist systems, Skousen's model allows for multiple possible outcomes for an operation on a word. Each possible outcome has a probability associated with it, derived from the proportion of exemplars in the analogical set which use that outcome. Random selection from a set of "activated" exemplars accounts for the variation inherent in real-life language. It also accounts for the possibility of "changing one's mind", or producing several guesses (see Skousen 1989:84–85). Since the speaker has several possible outcomes at his disposal, he may try one, decide against it, and then try another. Most deterministic, rule-based approaches do not allow for such multiple guesses.

### The creation of a lexical data set for predicting the /k/ ~ ∅ alternation

Since AM bases its predictions of an operation on the outcomes of similar words, the application of AM to any particular problem requires the creation of a suitable model of the mental lexicon – or at least that small subsection applicable to the problem. In this case, an appropriate subsection of the lexicon would include noun stems ending in /k/ or /g/. It could also be expanded to include all noun stems ending in any obstruent (/p/, /b/, /t/, /d/, /č/, /j/, /k/, /g/), although the words without final velars would be expected to play a less crucial role.

In order to model a subset of the Turkish mental lexicon, I extracted data from a lexicon based primarily on spoken Turkish and developed especially for morphological studies. This lexicon, the Turkish Electronic Living Lexicon (TELL), was developed by Sharon Inkelas at the University of California, Berkeley. It is a database of 30,000 Turkish words representing both print dictionaries and actual speaker knowledge. TELL was compiled from two editions of the Oxford Turkish-English dictionary, a telephone area code directory, and an atlas of Turkey. The 30,000 resulting lexemes were elicited, in various morphological contexts, from a 63-year old native speaker of a standard Istanbul dialect. The resulting database contains orthographic representations of these 30,000 headwords as well as phonemic transcriptions of all elicited forms. The native speaker knew and supplied pronunciations for some 17,500 of the elicited lexemes.<sup>5</sup>

The TELL lexicon contains approximately 3000 words ending in the velar stops /k/ or /g/, of which roughly half are nouns. Most, but not all, of these nouns were part of the informant's active vocabulary. Of the 1511 velar-final nouns found in the lexicon, 1440 are reported as "used" by the informant, and 71 are reported as "not used" by the informant. Other than this classification, there is no information regarding word frequency in the TELL database. Nevertheless, the informant's classification of words as *used* and *not used* provides a simple way of deciding which words are likely to be common (and therefore likely to be remembered) and which are likely to be uncommon (likely to be constructed by analogy). Accordingly, those

items which are labeled *used* have been used in constructing the lexical dataset. Those labeled *not used* were used as test items, to see if the AM algorithm would predict for them the same outcome as was supplied by the informant. In addition, two additional word lists were tested: a list of Arabic loan words with long final vowels, taken from Sezer (1981), and the second nonce test from Zimmer and Abbott (1978).

The TELL database provides three different types of vowel-initial suffix for most words: the definite accusative ('the X.ACC'), the first person singular possessive ('my X'), and the "predicative" or copulative ('I am an X'). Each word was saved up to three times in the lexical data set, once for each attested suffix. If an entry was (for whatever reason) missing one or more of these suffix fields, those instances of the word were omitted from the database (so as not to invent data not present). This may have resulted in a word receiving less "weight" than its neighbors, having only one or two instances in the data set instead of three. However, this weighting seems justifiable, for if the informant is unable to provide a form for all three fields, it may not be frequently used in all three suffixes, and thus this gap ought to be reflected in the data set.

Occasionally an entry had two alternate entries for the same suffix (e.g., *infilak* + PRED → *infilak-im*, *infilak-im*), then one of each copy was saved. Essentially, this resulted in a duplicate copy of the same variable set, but with different values for the outcome, and therefore, "opposing weights".

### The variable encoding scheme

In analogical modeling, the similarity or "nearness" between two words is determined by the number of variables for which the two words have identical values. Conversely, the number of non-matching variables determines "distance" from the given context. Unlike rule-based approaches (and some nearest-neighbor approaches), AM does not try *a priori* to determine which variables are "significant" and which are not. Instead, enough variables are included to give as complete a picture of the surrounding context as possible, within the constraints of the system. Although several different variable sets were tested, the basic set of variables was as follows:

- (1) the first phoneme of the word;
- (2–4) the onset, nucleus, and coda of the first syllable (for words of two or more syllables);
- (5–7) the onset, nucleus, and coda of the second syllable (for words of three or more syllables);
- (8–10) the onset, nucleus, and coda of the last syllable (not including the /k/ or /g/);



- (11) the last phoneme of the word stem (for all test items, this was either /k/ or /g/);
- (12–14) the vowel-length of the first, second, and last syllables (long or short);
- (15) the etymology (if known) of the word stem (Turkish, Arabic, Persian, or other);
- (16) the suffix being added to the stem (accusative, possessive, or predicative).

I will illustrate the encoding of the disyllabic word *akik* ‘oppression of humidity’, whose etymology is unlisted in the TELL lexicon. The vowel in the last syllable is underlyingly long, but it only appears as long with vowel-initial suffixes. In citation form the vowel is shortened before word-final /k/. This is characteristic of words in Exceptional class 2. However, the TELL informant seems to have had trouble remembering this word’s final vowel length, for he pronounces it long with only two of the three suffixes listed. Thus the word is borderline between the second exceptional class and the normal case. In this particular word (though not in all words, as shown below), the informant’s decision to retain or delete the velar depends on the final vowel length, as predicted by Sezer.

The beginning of the line (up to the first comma delimiter) is the “outcome” of the word, as given by the informant. In this data set there are three possible outcomes: *delete* the velar, *voice* the /k/ to /g/, or leave it the *same*. The informant retained the velar (*same*) with the accusative and predicative suffixes, but *deleted* it for the first-person possessive suffix.

The next portion of the encoding contains the sixteen variables. The first phoneme of the word (1) is /a/; and the onset, nucleus, and coda of the first syllable (2–4) are *null*, /a/, and *null* respectively. The slots for the second syllable (5–7) are *null*, since the second syllable is also the last syllable. The onset and coda (5, 7) are *predictably null* (=), since the absence of the syllable is already coded in the *null* nucleus (6). The onset, nucleus, and coda of the last syllable (8–10) is /k/, /i/, and *null*, counting the final /k/ (11) as syllabified in the same syllable as the suffix. The vowel length of the first syllable (12) is *short* (.), the second syllable’s vowel length (13) is *predictably null* (since there is no corresponding syllable), and the final syllable (14) is *long* ( \_ ) twice and *short* once. The etymology (15) is unknown (NULL), and the suffix (16) is either *accusative*, *possessive*, or *predicative*.

The final part of the encoding (listed under Comments) indicates the base and suffixed forms. It is not used by the algorithm itself, but provides identifying information to the user.

Outcome	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Comments
same,	a	0	a	0	=	0	=	k	i	0	k	.	=	_	NULL	accs,	akik->aki:ki
delet,	a	0	a	0	=	0	=	k	i	0	k	.	=	.	NULL	poss,	akik->akiim
same,	a	0	a	0	=	0	=	k	i	0	k	.	=	_	NULL	pred,	akik->aki:kim

### Tests run on the AM data-set

The first test consisted of the 71 velar-final words which the TELL informant claimed not to use in everyday speech. These should qualify by definition as “uncommon” words and therefore make excellent test cases, as they are not likely to be remembered as their own exemplars. In total 199 entries were tested, comprising all the attested suffix forms.

These 71 words were split into three categories, each of which was tested separately:

1. words for which all three suffixes uniformly deleted the final velar;
2. words for which all three suffixes uniformly retained the final velar;
3. words where the informant retained the velar on only one or two of the suffixes.

The first subset were all polysyllables ending in  $VC_0V[-long]k$ , the pattern predicted by Sezer to always delete. Within the second subset, there were seven monosyllabic words: three of the pattern  $CVC[+velar]$  (Exceptional class 1), three with the pattern  $CVCC[+velar]$ , and one  $CV[+long]k$ . As well, there were two polysyllabic words with the pattern  $VC_0V[+long]k$  (Exceptional case 2), and thirteen polysyllables ending in  $V[-long]C_0Vk$ . This retention of the velar is contrary to Sezer’s predictions; therefore, these words seem to form a third and previously unknown group of exceptions. The third subset consisted entirely of polysyllables ending in  $V[-long]C_0Vk$ . Their behavior (or this informant’s usage) is also wrongly predicted by Sezer’s rule, since Sezer predicts that all three of these suffixes should behave identically.

The second test consisted of twelve words taken from Sezer (1981), exemplifying Exceptional class 2. All were of Arabic origin and had a long final vowel followed by a /k/. These were tested in three ways: treating the final variable as long, treating it as short, and ignoring the final vowel-length altogether. (The latter two turned out to have equivalent results.) They were also tested with and without the etymology variable, to isolate its effect.

The final test run was the second nonce experiment from Zimmer and Abbott. All these words were coined by the experimenters. They include ten examples of Exceptional class 1 ( $C_0VC[+velar]$ ), one example of Exceptional class 2 ( $VC_0V[+long]k$ ) and three other examples of “Arabic sounding” words. These last three, and the remainder of the words, are polysyllables of the pattern  $VC_0V[-long]k$ . In Zimmer and Abbott’s nonce experiment, each nonce word was inflected according to Sezer’s predictions by a majority of the subjects.

## Discussion of results

### Results of the three tests

In the first test, all of the words with deleted final velars, including all of subset one, and about half of the items in subset three, were predicted correctly, with little average leakage. All of the monosyllabic words (Exceptional class 1) also were predicted correctly. These are the two classes of words where Sezer's rule was shown to be productive. Thus, the AM algorithm meets the benchmark set by this portion of Sezer's rule.

In the second test, all twelve of Sezer's examples for Exceptional class 2 were predicted to retain the velar when the final vowel was treated as long and the etymological variable was present. Eleven out of twelve (all but *ahla:k* 'morals') were predicted as velar-retaining without the etymological variable, and nine out of twelve when the final vowel was treated as short or ignored, with or without the etymological variable. (*Ahlak*, *infilak* 'explosion', and *ittifak* 'alliance' were predicted to delete when final vowels were ignored.) It seems clear that the vowel length does play a factor in the retention of the /k/, as Sezer predicts. However, it may not be a crucial variable, for it appears that some of the words can be correctly remembered as /k/-retaining even without the knowledge of the vowel length. This seems to be the situation with our informant. He is unsure of the proper vowel length of several words (e.g., *istimlak* 'expropriation'), reporting them as both long and short, but retains the /k/ in both cases. For him, vowel-length and final-/k/ retention are not completely connected – both exceptional features are gradually being lost, but the latter more slowly than the former.

Ten of Sezer's twelve examples were present in some form in the data set, and most of these were highly dependent on being "remembered" to retain the /k/. Only *istintak* 'interrogation', *istihkak* 'merit', and *inhimak* 'inclination' showed a "stable" final /k/ without the final long vowel, probably due to the number of Arabic loan-words beginning with *inC-* and *isti-* that keep the final /k/. Furthermore, the two "uncommon" words in this category (*misak* ~ *misa:kim* 'solemn promise, pact' and *revak* ~ *revakim* 'porch, pavilion') failed to be predicted as /k/-retaining without the long final vowel. Therefore, the AM model predicts that only common words are likely to remain /k/-retaining in a dialect that has lost vowel length distinctions. Over time, the words would be regularized to follow the patterns described by Sezer. We see that process in action in two of the "missed" words, *infilak* and *ittifak*. The informant has already lost the long final vowel for both of these words, reporting them as short. *Infilak* is present in the database not once but twice – once with final /k/, and once without – for all vowel-initial suffixes. *Ittifak* is reported without the final /k/ in all cases. In light of this data, Sezer's generalization describes a constellation of words, but not necessarily a productive rule.

We see here that in this informant's idiolect, this exceptional class is gradually being lost. AM correctly predicts both the direction of the change and its gradual, word-by-word nature.

In the third and final test, the nonce experiment, twenty-one of the twenty-four words were predicted "correctly" (i.e. as the majority of the subjects had them), and three were missed (*ruk*, *müstemek*, and *mok*). This is not remarkable, for about half of the subjects also "missed" three or more of the twenty-four items. It seems likely, especially upon inspection of the TELL database, that our informant would do the same, for his own performance shows significant variation from Sezer's predictive rules, as we will see below.

Velar-retaining polysyllables ending in  $V[-long]C[+velar]$ :

An unanticipated exception

There is a final category of exceptional words which has not yet been directly addressed. This "other" category of exceptional words is both the largest and the least expected, for it violates both Sezer's rules and the results of Zimmer's experiment. About 434 out of 3645 instances of commonly used words ending in  $VCVC[+velar]$ , or 11.9%, exceptionally retain the /k/ in our informant's usage. Unlike all other categories seen so far, these exceptions are numerous – nearly as numerous as all the rule-predicted /k/-retaining instances combined. Also, these exceptions are not evenly distributed among the three grammatical cases. Over half the instances are the "predicative" case; the accusative accounts for roughly one quarter of the instances and the possessive about one sixth. This is totally unaccounted for in the above-mentioned rule-based accounts, which predict that all suffixes of this type should behave the same with respect to retaining or deleting the velar. Yet these trends are even more pronounced among the words not commonly used.

These "exceptional" words were also incorrectly predicted by the AM algorithm, with very large leakages. These may be true exceptions, but it seems odd that they should be so numerous, and that so many should all be labeled as "not used". There seems to be no common link between them: several, like *almanak* 'almanac' and *mihanik* 'mechanics', are clearly loan words. But others, like *benlik* 'egoism', are clearly native (though perhaps neologisms). Some are marked as /k/-retaining in the Oxford Dictionary, others are not. The cause (or causes) of these exceptions is still unknown. These may be instances of "old-fashioned" or hypercorrect speech, uncertainty in the face of uncommon words, or just a personal idiosyncrasy on the part of the informant. They could also be influenced by semantic factors not considered in this study.<sup>6</sup>

## Implications of the results: Lexical explanations for the data

### Monosyllables (Exceptional class 1)

According to Skousen's theory, large numbers of similar words grouped "close together", as it were, in the lexicon can draw the words around them away from a more general pattern. All words of the pattern  $C_0VCC[+velar]$  retain the velar as a matter of course, by virtue of the consonant adjacent to the velar. Nevertheless, because they are similar in length and sound to the  $C_0VC[+velar]$  monosyllables, they may easily influence them. If this analysis is correct, the number of syllables is important, not because of a constraint, but because of the distribution of nearby words. The 159 instances of monosyllabic velar-retaining  $C_0VCC[+velar]$  stems form a "gang" sufficiently large to influence the 137 instances of the (phonologically similar) monosyllabic  $C_0VC[+velar]$  stems. Thus the  $C_0VC[+velar]$  stems will be pulled towards retaining the velar like their  $C_0VCC[+velar]$  neighbors. However, the 86 instances of polysyllabic  $VC_0VCC[+velar]$  stems are too few to influence the 3645 instances of polysyllabic  $VC_0VC[+velar]$ , which remain velar-deleting.

It has been suggested that the "exceptional" retention of velars in  $CVC[+velar]$  monosyllables is due to a bi-moraic minimality constraint (Inkelas & Orgun 1995). No evidence has been found in this study to refute this position, and I see no reason to discount it as a possible factor. However, the results obtained from this study suggest that there are alternate explanations, or at least other possible factors. It is possible that the "exceptional" behavior of monosyllables simply arises out of the distribution of words in the lexicon as described above. Alternatively, it may be that a universal tendency and analogical factors have combined to create and maintain the current "exceptional" pattern.

### Polysyllables with long final vowels (Exceptional class 2)

The explanation for the second exceptional class is not as straight-forward, and may possibly involve sociolinguistic factors. It may be that the vowel length fills the coda, making these stems equivalent to the velar-retaining  $VC_0VCC[+velar]$  stems. However, Zimmer and Abbott (1978) have suggested that it is not just the vowel length of the last syllable, but the "Arabic sound" of the whole word which encourages the retention of the final /k/. Originally, when Arabic was a language of prestige and culture in Turkey, Arabic words would be pronounced as close to their original Arabic pronunciation as possible. This conservatism would prevent the application of "native" sound changes such as the /k/ ~  $\emptyset$  alternation. Eventually, some Arabic words will have been regularized into Turkish pronunciation patterns, as their origins are forgotten. But since phonemic vowel length is not a feature

native to Turkish, a phonemically long vowel (if preserved) would signal the foreign status of the word and encourage the retention of the velar.

Political climates change, and Arabic-like pronunciation of Turkish words is perhaps no longer a symbol of prestige. However, in those dialects where vowel length is preserved, a sufficient number of these words have remained in current use to reinforce one another and retain their velars, regardless of their etymology. Tests with Skousen's AM algorithm confirm that, when vowel length is allowed as a factor, it will predict the preservation of final /k/ for nearly all of Sezer's examples. Without vowel length, most of the common words are still correctly predicted. Uncommon words tend to be regularized, unless they contain other salient "Arabic" features, such as beginning with *inC-* (as in *inhimak*) or *isti-* (as in *istimlak*, *istintak*, *istihkak*). Kibre (1998) suggests that a variety of phonological features, none obligatory, may serve to identify a word as Arabic. The results generated by AM seem to confirm his claim.

#### Velar-retaining polysyllables ending in $V[-long]C[+velar]$

The last group of exceptions is as yet unaccounted for, both by rule-based approaches and by AM. A variety of factors may be at play, some of which have been discussed above. Another possible factor is the influence of consonant-initial suffixes. Since the velar deletes only between vowels, it is always retained before a consonant-initial suffix. If these words are used much more frequently with consonant-initial suffixes than vowel-initial ones, this may cause the generalization of /k/-retention.

As the TELL lexicon is expanded and refined, these exceptions may prove to reflect the idiosyncrasies of one speaker, or artifacts of the process of building the lexicon. But if the same variations are manifest in other speakers also, then more work will be needed to explain these phenomena.

## Conclusion

Skousen's analogical modeling is seen here to predict the particulars of the /k/~/Ø alternation in Turkish to a fairly high degree of accuracy. It reaches the "benchmark" set by the rule systems proposed by Sezer (1981) and Inkelas and Orgun (1995) with regard to monosyllables, the first class of exceptional words. This suggests that rule-based approaches with strict divisions between regular and irregular categories are not necessary to explain certain morpho-phonological patterns. Rather, these patterns may possibly emerge from the lexicon through mutual analogical influence between lexical items or exemplars.

AM also provides additional insights into the second class of exceptional words, polysyllables with final /k/ preceded by a long vowel in the last syllable. Whereas Sezer's account made no distinction between the productivity of these two exceptional classes, AM shows the second class to be much weaker than the first, and subject to regularization into the velar-deleting norm. This prediction accords with the variation in the informants' responses to Zimmer and Abbott's nonce examples, as well as the variable data encountered in the TELL lexicon.

However, there is still work to be done. Neither the current model under AM nor the rule systems proposed to date are adequate to fully describe the performance of the TELL informant. On closer examination of the TELL database, a third class of exceptions was found, as yet unexplained by rule approaches or by AM. This "class" of exceptional words, if it may be so called, does not appear to have any distinguishing characteristics that would make it amenable to a rule-based approach. If similar variations are found in other speakers, then a more refined model in the AM paradigm will be necessary to capture this type of variation. Nevertheless, by accounting for the variation in the second exceptional class, AM comes closer to descriptive adequacy than deterministic rule-based approaches.

## Notes

\* I would like to thank Royal Skousen of Brigham Young University for his support of this research and for continued guidance in the application of his algorithm. I would also like to thank Kemal Oflazer of Bilkent University for his generosity in sharing his time and expertise in answering many questions concerning the Turkish language, and for providing a wealth of data which proved very helpful in the beginning stages of this research. I also thank Sharon Inkelas and others at the University of California, Berkeley, for use of the Turkish Electronic Living Lexicon. An earlier version of this research appears in Rytting 2000.

1. Sezer also addresses a third exceptional class, roots followed by verbal or "non-native" affixes:

<i>meslek</i> 'profession'	<i>mesle-im</i> 'profession-1SG.POSS'	<i>meslek-i:</i> 'profession-al'
<i>na:zik</i> 'kind'	<i>na:zi-im</i> 'kind-1SG.COPULA'	<i>na:zik-en</i> 'kind-ly'

I have not addressed this exception in my research to date, although I assume these words are stored as separate lexical items. For the time being I am restricting my inquiry to inflectional morphology.

2. Lewis (1967) and Sezer (1981) make note of three common lexical exceptions to this global exception: *çok* 'many', *yok* 'there is not', and *gök* 'sky'.

3. They account for Sezer's third exceptional class by assuming that Turkish words go through multiple levels of representation between the 'deepest' lexical level and the phonetic realization, and that certain rules such as velar drop are only applicable at certain levels. The suffixes in Exception class 3 attach sooner than the native denominal suffixes, before the velar drop rule is active. The exceptions in Note 2 are handled by lexical pre-specification.

4. They did not test Exceptional class 3, which they evidently took to be a given.
5. Taken from <<http://socrates.berkeley.edu:7037/AboutTELL.html>> as of 30 July 1999.
6. Several of those words which retained the final velar with the copulative suffix only have semantically odd readings: for example, *işıldakım* ‘(I am a) searchlight’, *ondalıkım* ‘(I am a) tithe’, *mertekim* ‘(I am a) beam of wood’. However, there is no reason to believe that these are less semantically odd than many words which deleted the /k/ as expected, nor does this explain cases where the /k/ was retained with two or three suffixes.

## References

- Inkelas, Sharon, & Cemil Orhan Orgun (1995). Level ordering and economy in the lexical phonology of Turkish. *Language*, 71, 763–793.
- Iz, Fahir, H. C. Hony, & A. D. Alderson (Eds.). (1992). *Oxford Turkish Dictionary*. Oxford; New York: Oxford University Press.
- Kibre, Nicholas (1998). Between irregular and regular: ‘Imperfect generalizations’ in Istanbul Turkish and the status of phonological rules. In M. Darnell, E. Moravcsik, F. Newmeyer, M. Noonan, & K. Wheatley (Eds.), *Functionalism and formalism in linguistics* (pp. 131–149). Amsterdam; Philadelphia: John Benjamins.
- Lewis, Geoffrey L. (1967). *Turkish grammar*. Oxford: Clarendon Press.
- McCarthy, John, & Alan Prince (1986). Prosodic morphology. Manuscript, University of Massachusetts at Amherst and Brandeis University.
- New Redhouse Turkish-English Dictionary* (1968). Istanbul: Redhouse Press.
- Rytting, Anton (2000). An empirical test of analogical modeling: The /k/~/Ø alternation. In A. K. Melby & A. R. Lommel (Eds.), *LACUS Forum XXVI, The Lexicon* (pp. 73–84). Fullerton, CA: The Linguistic Association of Canada and the United States.
- Sezer, Engin (1981). The /k/~/Ø alternation in Turkish. In G. N. Clements (Ed.), *Harvard studies in phonology* (pp. 354–82). Bloomington: Indiana University Linguistics Club.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Van Schaaik, Gerjan (1996). *Studies in Turkish grammar*. Wiesbaden: Harrassowitz Verlag.
- Zimmer, Karl E., & Barbara Abbott (1978). The /k/~/Ø alternation in Turkish: some experimental evidence for its productivity. *Journal of Psycholinguistic Research*, 7, 35–46.





PART IV

## Comparing Analogical Modeling with TiMBL



## CHAPTER 6

# A comparison of two analogical models

## Tilburg Memory-Based Learner versus Analogical Modeling\*

David Eddington

### Introduction

Linguistics in the latter half of the twentieth century has been largely dominated by the rule-based paradigm of generativism. However, in the past few years, a number of non-rule approaches have been proposed and have gained some ground. Interest in non-rule approaches to linguistics may be the result of several different factors: disillusion with the generative paradigm, skepticism regarding the psychological relevance of generative analyses (Eddington 1996), advances in applying computer technology to questions of language (Natural Language Processing), and the heightened interest of psychologists in linguistic issues. Connectionism (see McClelland 1988 for an overview) has surfaced as the most prominent non-rule rival of the rule-driven orthodoxy, and the ongoing debate between connectionists and generativists has been intense (e.g. Clahsen et al. 1992; Daugherty & Seidenberg 1992, 1994; Marcus et al. 1995; Pinker 1991, 1997; Pinker & Prince 1994; Seidenberg 1992; Seidenberg & Bruck 1990).

In spite of its prominence, connectionism is not the sole non-rule model in existence. The present work compares two non-rule models of linguistic cognition, namely Analogical Modeling (AM) (Skousen 1989, 1992, 1995), and the nearest neighbor approach employed by the Tilburg Memory-Based Learner (TiMBL) (Daelemans et al. 1999). Both of these approaches belong to a family of models known as analogy-based, exemplar-based, or instance-based models (e.g. Bod 1998; Medin & Schaffer 1978; Nosofsky 1988, 1990; Riesbeck & Schank 1989; see Shanks 1995 for an overview of exemplar-based models). All of these models assume that previously encountered or processed information is stored in memory and is accessed and used to predict subsequent language behavior. Since each instance-based model employs a different algorithm, it is important to determine

if there are significant empirical differences between the predictions they make. Therefore, the focus of this paper will be to compare AM and TiMBL in terms of their performance on a number of different tasks. I will begin by reviewing the study by Daelemans et al. (1994b) which compares the ability of AM and TiMBL to assign stress to monomorphemic Dutch words. Next, I compare TiMBL and AM in terms of their ability to account for Spanish diminutive formation, gender assignment, and stress assignment.

### 1. The TiMBL algorithm

Before reviewing the evidence from Dutch stress assignment, it is important to understand how TiMBL calculates nearest neighbors. TiMBL is essentially an expansion of the algorithm developed by Aha et al. (1991). It is designed to take an input and find its nearest neighbor(s) in a database of exemplars. During the training session, the model stores in memory series of variables which represent instances of words (or some other entity). The words are stored along with their behavior (e.g., which syllable is stressed, the word's gender, etc.). In the case that the same word is encountered more than once, a count is kept of how often each word is associated with a given behavior. During the testing phase, when a word is given as input, the model searches for it in memory and applies the behavior that it has been assigned in the majority of cases. If the word is not found in memory, a similarity algorithm is used to find the most similar item in memory – its nearest neighbor. The behavior of the nearest neighbor is then applied to the word in question. If two or more items are equidistant from the word in question, the most frequent behavior of the tied items is applied to the word in question.

The TiMBL algorithm contains several variants. For example, it can be set to determine the behavior on the basis of a single nearest neighbor, or on the basis of several nearest neighbors. In its basic instantiation, called *Overlap*, all variables are weighted equally. However, two extended algorithms are also available. *Information Gain* is a variant of *Overlap* which precalculates how much each variable contributes to determining the correct behavior. These variables are weighted accordingly when calculating similarity and determining nearest neighbors. When a calculation of similarity is carried out using *Overlap*, the values of a variable are all considered equidistant from each other. However, the *Modified Value Difference Metric* is also an available option. It is used to precalculate the similarity between the values of a variable, and to adjust the search for nearest neighbors accordingly. In effect, this allows certain values to be regarded as more similar to each other than other values.

## 2. Dutch stress assignment in TiMBL and AM

Daelemans et al. (1994a) constructed a database consisting of 4860 monomorphemic multisyllabic Dutch words. Since stress may fall on any of the final three syllables, the phonemic content of the final three syllables of each word served as the variables in the AM and TiMBL comparisons (Daelemans et al. 1994b). Several ten-fold cross-validation simulations were performed on the database. This involved partitioning the database into ten sets of 486 words, and then running ten simulations for each experimental condition. Each of the ten sets of 486 words had its turn as a test set in one of the ten simulations; the words in the remaining nine sets formed the training sets.

Daelemans et al. (1994b) applied the basic Overlap algorithm in which the behaviors of one, two, five and ten nearest neighbors were applied to the words in the test sets. In all four conditions, AM's success rate (80.5%) was statistically superior to those produced by TiMBL. However, when varying degrees of noise were added to the four conditions, both models performed equally well (or poorly). When variables were weighted with the Information Gain (IG) algorithm, the Modified Value Difference Metric algorithm (MVDM), and both the IG and MVDM algorithms together, the success rates (81.8%, 79.4%, 81.4% respectively) became statistically equal to that of AM. In short, the findings from the Dutch stress assignment study indicate that TiMBL's modified algorithms are equally adept at correctly assigning stress as AM. However, it is important to determine if this equivalence will hold true when other data are considered. If not, it is of interest to know which model is empirically superior. To this end, data from Spanish were considered.

## 3. Spanish gender assignment

The ability to predict gender seemed an apt task for an analogical model. All Spanish nouns belong to either the masculine or feminine gender. In general, words ending in *-o* are masculine, while those which end in *-a* are feminine. However, there are many exceptions to this generalization, and it is much more difficult to predict the gender of words ending in other phonemes.

The database for the gender simulation included the 1739 most frequent nouns in the Spanish language taken from LEXESP (Sebastián, Martí, Carreiras, & Cuetos 2002).<sup>1</sup> Each noun was encoded to include the phonemic make-up and syllable structure of the penult rhyme and final syllable. The nouns were also marked as to whether they had masculine or feminine gender (for details, see Eddington 2002). Again, both TiMBL's and AM's algorithms were put to the task. AM successfully assigned gender to 94.5% of the database items.

**Table 1.** Success rate on correctly assigning gender to database items

Algorithm	#	%	$\chi^2$	p <
AM	1645	94.5		
TiBML-no weighting, 3 nn	1563	89.9	2.147	0.25
TiBML-Information Gain, 3nn	1650	94.9	0.005	0.95
TiBML-MVDM, 3 nn	1673	96.2	0.220	0.75
TiBML-MVDM and Info. Gain, 3nn	1668	95.9	0.146	0.75

MVDV = Modified Value Difference Metric; nn = number of nearest neighbors calculated

Given the fact that many nouns have the identical phonological content in their penult rhyme and final syllable, it was necessary to eliminate exact matches between the test item and any items in the database. In the AM simulation, this was done by eliminating any identical given contexts which existed in the database. In order to achieve the same effect in TiBML, it was necessary to set the option to avoid choosing neighbors which are exact matches. This option requires that more than one nearest neighbor be selected, and in order to avoid ties between neighbors with different behaviors, the number of nearest neighbors needs to be odd. For this reason, the analogical influence of three nearest neighbors was considered in the TiBML simulations. Four different TiBML simulations were run using the basic overlap algorithm with no weighting, Information Gain (IG), the Modified Value Difference Metric algorithm (MVDM), and both the IG and MVDM algorithms together. As Table 1 indicates, the success rates of all of the TiBML simulations do not differ significantly from that of AM.

### 3.1 Gender assignment task<sup>2</sup>

According to the outcome of the study on the database, no statistically significant difference was found between the two models. Therefore, each model was tested as to its ability to predict native speaker's intuitions about the gender of novel words.

#### 3.1.1 *Stimulus materials*

118 nouns were extracted from *Diccionario de la lengua española* (Real Academia Española, 1995). Each of these words is considered antiquated and of infrequent use (see Appendix). Therefore, they were highly unlikely to be known by the subjects, which also means that their gender would be unknown. Words were chosen that ended in phonemes other than *o* and *a*. In this way, the more obvious gender/phoneme correspondences were eliminated, and the subjects were obliged to make gender assignments on the more ambiguous cases.

### 3.1.2 *Subjects*

31 literate native Spanish speakers from Spain participated in the study, 18 women and 13 men. The average age of the subjects was 33.4.

### 3.1.3 *Procedure*

The 118 test items were presented in the form of a written questionnaire. The subjects were asked to circle either the feminine article *la* or the masculine article *el*, which appeared before each test item. They were instructed to choose the article that was most appropriate for the word that followed. Using the database of 1739 words previously described, the 118 words from the study were assigned gender by AM and by TiMBL's most successful algorithm (3nn, MVDM; see Table 1).

### 3.1.4 *Results*

TiMBL assigned the same gender as the subjects in the study to 67.8% of the test items. AM scored slightly higher at 71.2%. Nevertheless, the difference is once again not significant ( $\chi^2 = 0.055, p < .5$ ).

## 3.2 Gender of borrowed words

Another task which analogical models appear to be well suited is predicting the gender of foreign words adopted into Spanish. Zamora (1975) studied borrowings from English into Puerto Rican Spanish. He asked 13 bilingual speakers to determine the gender of 20 English words that are commonly used in Puerto Rican Spanish. He also discusses 67 Native American words which were adopted into Spanish and had to be assigned a gender. Gender predictions were provided by AM and TiMBL for these words based on the phonemic make-up of the penult rhyme and final syllable of the words' Spanish adaptation. TiMBL's most successful algorithm (3nn, MVDM) successfully predicted 86.2% of the 87 borrowings considered, while AM attained a success rate of 90.8%. The small difference is not significant ( $\chi^2 = 2.157, p < .25$ ).

As far as the data from gender assignment are concerned, both models perform equally well, and the superiority of one over the other cannot be asserted. Nevertheless, gender is a fairly simple phenomenon since it only entails two possible outcomes. Differences between the models may be found in predicting behaviors with many outcomes. To this end, an experiment with diminutive formation was carried out.



#### 4. Spanish diminutive formation

The formation of diminutive variants of nouns, adjectives and certain adverbs is a highly productive process in Spanish. Several diminutive suffixes exist (*-ito*, *-illo*, *-zuelo*, *-ico*, *-uco*), but *-ito* is the most common, which is why only diminutives ending in *-ito/a* were considered. All such diminutive forms were extracted from a number of databases.<sup>3</sup>

With the exception of a handful of highly irregular items, all diminutives fall into one of 13 categories. A circled *V* or *s* indicates that that element of the base form does not appear in the diminutive form:

1. -①ITO(S): *-ito(s)* is added to the singular base form, replacing the final vowel: *minuto* > *minutito*, *elefante* > *elefantito*.
2. -①ITA(S): *-ita(s)* is added to the singular base form, replacing the final vowel: *galleta* > *galletita*, *Lupe* > *Lupita*.
3. -①ECITO(S): *-ecito(s)* is added to the singular base form, replacing the final vowel: *vidrio* > *vidriecito*, *quieto* > *quietecito*.
4. -①ECITA(S): *-ecita(s)* is added to the singular base form, replacing the final vowel: *yerba* > *yerbecita*, *piedra* > *piedrecita*.
5. -CITO(S): *-cito(s)* is added to the singular base form: *traje* > *trajecito*, *pastor* > *pastorcito*.
6. -CITA(S): *-cita(s)* is added to the singular base form: *joven* > *jovencita*, *llave* > *llavecita*.
7. -ITO(S): *-ito(s)* is added to the singular base form: *normal* > *normalito*, *Andrés* > *Andresito*.
8. -ITA(S): *-ita(s)* is added to the singular base form: *nariz* > *naricita*, *Isabel* > *Isabelita*.
9. -ECITO(S): *-ecito(s)* is added to the singular base form: *pez* > *pececito*, *rey* > *reyecito*.
10. -ECITA(S): *-ecita(s)* is added to the singular base form: *flor* > *florecita*, *luz* > *lucecita*.
11. -①②ITOS: *-itos* is added to the singular base form, replacing the vowel and false plural morpheme: *lejos* > *lejitos*, *Marcos* > *Marquitos*.<sup>4</sup>
12. -①②ITAS: *-itas* is added to the singular base form, replacing the vowel and false plural morpheme: *Lucas* > *Luquitas*, *garrapatas* > *garrapatitas*.
13. -①CITA(S): *-cita(s)* is added to the singular base form, replacing the final vowel: *jamona* > *jamoncita*, *patrona* > *patroncita*.

The resulting database contained 2450 diminutive forms. Each base form was marked as to which of the 13 categories its diminutive belonged to, and the following information about each base form was included: (1) the stressed or unstressed status of the final two syllables; (2) the gender of the word: masculine,

**Table 2.** Success rate on correctly assigning diminutives to database items

Algorithm	#	%	$\chi^2$	p <
AM	2285	93.27		
TiBML-no weighting, 3 nn	2238	91.35	0.468	0.5
TiBML-no weighting, 5 nn	2136	87.18	13.604	0.001
TiBML-Information Gain, 3nn	2267	92.53	0.063	0.5
TiBML-MVDm, 3 nn	2271	92.69	0.037	0.9
TiBML-MVDm and Info. Gain, 3nn	2269	92.61	0.049	0.9

MVDV = Modified Value Difference Metric; nn = number of nearest neighbors calculated

feminine or none in the case of adverbs and gerunds; (3) the word's final phoneme; (4) the phonological content of the antepenult rhyme and the final two syllables of the word.

A ten-fold cross-validation simulation was performed using AM's algorithm, and several of TiBML's algorithms. In the no weighting conditions using TiBML, and in the AM simulation, the gender variable and the word's final phoneme were included twice in order to weight them more heavily than any other single variable. This duplication was removed in the Information Gain and Modified Value Difference Metric simulations, since these algorithms are designed to calculate the importance of the variables and values on their own. As Table 2 indicates, the TiBML simulations performed as well as the AM simulation with the exception of the simulation calculated without any of TiBML's weighting algorithms using five nearest neighbors.

Many of the erroneous diminutives predicted by AM and TiBML's most successful instantiation appeared to be plausible diminutives. This is evident in the predictions made on the doublets in the database (e.g. *cuentito*, *cuentecito*). In each case, errors on one member of the doublet always entailed assigning it the diminutive suffix of the other member. This assignment occurred in spite of the fact that, when tested, both members of a doublet were excluded from the database and were unable to serve as analogs for each other. In order to determine if other erroneously predicted forms were actually well-formed diminutives in some dialect of Spanish, the World-Wide Web was consulted. All erroneous forms, were sought on Spanish language pages. Of the 165 errors produced by AM, attested forms of 77 were found, either as an attested doublet in the database or on a Spanish language web page. Therefore, only 88 errors involved truly unattested diminutive forms. In the TiBML simulation, only 84 errors were unattested.

As far as diminutivization in Spanish is concerned, TiBML and AM are able to correctly produce the great majority of the tested forms correctly. An inspection of the errors made by both models does not yield any insight that allows one model to be declared superior to the other.

## 5. Spanish stress assignment

Stress in Spanish generally falls on one of the last three syllables. The database chosen for the present study essentially includes the 4970 most frequent words, and word plus clitic pronoun combinations, from the Alameda and Cuetos frequency dictionary (1995). (Details about the database and variables are found in Eddington 2000.) As in the Dutch study, the phonemic content of the final three syllables of each word was used as variables. However, unlike the study on Dutch, both monomorphemic and polymorphemic words appeared in the Spanish database. This is a crucial difference since in Spanish stress is often contrastive, especially in polymorphemic and verbal forms: *encontrára* ‘s/he found, imperfect subjunctive,’ *encontrará* ‘s/he will find’; *búsko* ‘I seek,’ *buscó* ‘s/he sought.’

Therefore, in addition to the phonemic information, morphological variables were included. For verbal forms, one variable indicated the person, and three identical variables indicated the tense form of the verb. Repeating a variable more than once is the only way to manipulate the weight of one variable or another prior to running the AM program. In essence, what this implies is that the tense form of the verb is considered three times more important than any single onset, nucleus or coda. In the AM simulation, the only significant difference that weighting this variable made was in the number of errors that occurred on preterit verbs with final stress. Fifty errors occurred without the weighting, in comparison to 27 when it was included three times.

Each word was encoded as a series of 13 variables. In Table 3, hyphens represent empty categories, ‘0’ indicates that the entry is a non-verb, ‘pt’ designates the verb is in the preterit tense, and ‘6’ defines the verb as third person plural.

Given the fact that the database contained several inflectional variants of many words, a possible confound exists. If one of the test items is the adjective *rójas*, the chances are quite high that its inflectional variants *rójo*, *rója*, and *rójos* will be chosen as nearest neighbors and influence it to receive penult stress. This is an undesirable state of affairs since the purpose of the study is to determine how successfully the model can assign stress to words that it is unfamiliar with. A simple way of controlling for this unwanted effect was to alphabetize the database prior to partitioning it for the ten-fold study. In this way, inflectional variants were grouped together in the same test set, and were unable to serve as analogs for each other.

Table 3. Examples of variable assignment

Examples	Stress	Morphological variables	Phonemic variables
<i>personal</i>	Final	– – – 0	p e r s o – n a l
<i>hablaron</i>	Penult	6 pt pt pt	– a – bl a – r o n

**Table 4.** Success rate on correctly assigning stress to database items

Algorithm	#	%	$\chi^2$	p <
AM	4693	94.4		
TiBML-no weighting, 1 nn	4628	93.1	0.439	0.25
TiBML-no weighting, 2 nn	4565	91.8	1.742	0.25
TiBML-no weighting, 5 nn	4019	80.8	51.989	0.001
TiBML-no weighting, 10 nn	3675	73.9	123.600	0.001
TiBML-Information Gain, 1nn	4643	93.4	0.257	0.75
TiBML-MVDM, 1 nn	4688	94.3	0.002	0.9
TiBML-MVDM and Info. Gain, 1nn	4657	93.7	0.131	0.75

MVDV = Modified Value Difference Metric; nn = number of nearest neighbors calculated

Once the database was partitioned, the stress placement of each word was determined in a ten-fold cross-validation. AM successfully assigned stress to 94.4% of the words in the database. This success rate is compared with those produced by TiBML under the same experimental conditions tested in the study on Dutch stress assignment. Note that in the no weighting conditions using TiBML, and in the AM simulation, the tense form variable was included three times. It was only included once in the Information Gain and Modified Value Difference Metric simulations, since these algorithms are designed to calculate the importance of the variables and values on their own. In five of the seven experimental conditions, the success rates for the AM and TiBML algorithms were statistically equivalent.

As previously mentioned, the database contained only the most frequently occurring Spanish words. It may be that extremely infrequent words have different stress patterns. To test this, a set of 497 words was assembled from among the items in Alameda and Cuetos (1995) that had a frequency of 0.2 per million. The words in this test set were assigned stress in a ten-fold cross-validation study according to AM and TiBML (Modified Value Difference Metric, one nearest neighbor). The resulting success rates were 91.8% and 90.2% respectively ( $\chi^2 = .0603$ ,  $p < .9$ ). It again appears that neither model may claim superiority over the other.

### 5.1 Error analysis

Given the similar success rates of both models, an analysis of the errors made by each model was performed in order to uncover any telling differences. The analysis compares AM with TiBML's most successful simulation (namely, MVDM) and calculates only one nearest neighbor. Table 5 specifies the number of errors made in each category, as well as the percentage of database items on which errors were made.

Table 5. Errors per category

	AM		TiMBL-MVDM, 1 nn		$\chi^2$	p <
	#	%	#	%		
Penult	41	1.2	122	3.4	39.2638	0.001
Final	72	6.4	59	5.2	1.0992	0.5
Antepenult	164	59.9	101	36.9	14.5056	0.001

Both models fare equally well on predicting final stress. However, TiMBL proves more consistent in predicting antepenult stress, while AM is more adept at predicting penult stress. In terms of the percentage of errors per category, both models show the same hierarchy of difficulty: penult < final < antepenult. This is consistent with the hierarchy of difficulty that native Spanish speaking children demonstrated on a number of stress placement tasks (Hochberg 1988), and provides further evidence that the models' predictions have empirical value.

One test of the models' accuracy is the extent to which they have captured the classes of regularity and irregularity in the Spanish stress system. In Spanish, penult stress is regular (or unmarked) for words ending in a vowel or /s/; final stress is regular for words ending in any consonant except /s/; antepenult stress is always marked (see Eddington 2000). A model which captures this stress system would be expected to make most of its errors on words with irregular stress. Of the 282 errors made by TiMBL, 156 (or 56%) occurred on irregularly stressed words. On the other hand, 80.1% (222 of 277) of the errors made by AM were made on irregularly stressed words.

The percentages just cited are interesting, but not indicative of a true difference between the models. It is important to ascertain, not only how many errors are made on irregular items, but the direction of the errors. That is, do the errors on the irregular items move stress onto the syllable which regularizes stress, or onto a syllable that keeps the word stress irregular. A model that correctly captures Spanish stress should also be expected to commit few errors that assign irregular stress to a word that is regularly stressed. In Hochberg's study (1988), children made more errors that regularized irregularly stressed words compared to errors that gave regularly stressed words an irregular stress.

Table 6 summarizes the rates of regularization and irregularization produced by each model. The database contains 649 irregularly stressed words and 4177 words with regular stress. In calculating rates of regularization and irregularization, all 144 monosyllabic words were excluded.

As is evident in the data, AM appears to have more correctly captured the Spanish stress system. It imposes regularity on irregular items to a greater extent than TiMBL. In addition, it assigns irregular stress to fewer regular items.

**Table 6.** Rates of regularization and irregularization

	AM	TiMBL	$\chi^2$	p <
# Regularized	220	155	10.9226	0.001
% Regularized	33.9	23.9		
# Irregularized	54	127	28.6408	0.001
% Irregularized	1.3	3.0		

## 6. Conclusions

The purpose of this study was to compare TiMBL and AM on a number of different tasks. Neither model significantly outperformed the other in the gender assignment and diminutive assignment tasks. In the previous comparison by Daelemans et al. (1994b), AM outperformed TiMBL on a Dutch stress assignment task, except than when noise was added to the system they performed equally well. The present study pitted the two models against each other in terms of their ability to assign stress to Spanish words. Both models were able to correctly assign stress to the most frequent 4970 Spanish words with about a 94% degree of accuracy. Their performance on highly infrequent words was slightly lower, but neither model was able to statistically outperform the other on either of these tasks. The only differences were evident in the error analysis. AM applied the regular stress patterns to irregularly stressed words to a greater extent than TiMBL. TiMBL, on the other hand, had a higher incidence of misapplying irregular stress patterns to regularly stressed items. This indicates that AM more successfully captured patterns of regularity and irregularity in the Spanish stress system.

## Notes

\* This study was carried out with the help of a grant from the National Science Foundation (#00821950).

1. The current study is based on an earlier pre-print version of LEXESP, a morphologically tagged frequency dictionary of Spanish of about 3 million words. A more recent printed version is based on a 5 million word corpus (Sebastián, Martí, Carreiras, & Cuetos 2002).
2. I am most indebted to Milagros Malo Fernández and Elías Álvarez Ortigosa who generously gave of their time to administer the questionnaires.
3. Alameda and Cuetos (1995); Sebastián, Martí, Carreiras, and Cuetos (2002); Marcos Marín (no date a, no date b). In addition to these sources, Mark Davies of Illinois State University graciously provided me with the diminutive forms from his corpus project totaling 39.8 million words: <<http://mdavies.for.ilstu.edu/personal/texts.htm>>.

4. In some words from groups 11 and 12, *s* represents what seems to be the plural morpheme since it appears word finally and follows a stressless vowel. In other cases, such as *cumpleaños*, the word ends in the plural morpheme derivationally speaking (*cumple* + *años* ‘complete + years’), but is used to denote both the plural and singular.

## References

- Aha, David W., Dennis Kibler, & Marc K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Alameda, José Ramón, & Fernando Cuetos (1995). *Diccionario de frecuencias de las unidades lingüísticas del castellano*. Oviedo, Spain: University of Oviedo Press.
- Bod, Rens (1998). *Beyond grammar: An experienced-based theory of language*. Cambridge: Cambridge University Press.
- Clahsen, Harald, Monika Rothweiler, Andreas Woest, & Gary Marcus (1992). Regular and irregular inflection in the acquisition of German noun plurals. *Cognition*, 45, 225–255.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1994a). Skousen’s analogical modeling algorithm: A comparison with lazy learning. In D. Jones (Ed.), *Proceedings of the International Conference on New Methods in Language Processing* (pp. 1–7). Manchester: UMIST.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1994b). The acquisition of stress: A data-oriented approach. *Computational Linguistics*, 20, 421–451.
- Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, & Antal van den Bosch (1999). *TiMBL: Tilburg memory based learner, version 2.0, reference guide* [Induction of Linguistic Knowledge Technical Report]. Tilburg, Netherlands: ILK Research Group, Tilburg University. <<http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz>>
- Daugherty, Kim, & Mark S. Seidenberg (1992). Rules or connections? The past tense revised. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (pp. 259–264). Hillsdale, NJ: Erlbaum.
- Daugherty, Kim, & Mark S. Seidenberg (1994). Beyond rules and exceptions: A connectionist approach to inflectional morphology. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 353–388). Amsterdam: Benjamins.
- Eddington, David (1996). The psychological status of phonological analyses. *Linguistica*, 36, 17–37.
- Eddington, David (2000). Spanish stress assignment within the Analogical Modeling of Language. *Language*, 76, 92–109.
- Eddington, David (2002). Spanish gender assignment in an analogical framework. *Journal of Quantitative Linguistics*, 9, 49–75.
- Hochberg, Judith (1988). Learning Spanish stress: Developmental and theoretical perspectives. *Language*, 64, 683–706.
- Marcos Marín, Francisco (no date a). Corpus oral de referencia del español contemporáneo. Textual corpus, Universidad Autónoma de Madrid. <[http://elvira.llf.uam.es/docs\\_es/corpus/corpus.html](http://elvira.llf.uam.es/docs_es/corpus/corpus.html)>

- Marcos Marín, Francisco (no date b). Corpus lingüístico de referencia de la lengua española en Argentina. Textual corpus, Universidad Autónoma de Madrid. <<http://www.lllf.uam.es/~fmarcos/informes/corpus/coarginl.html>>
- Marcus, Gary F., Ursula Brinkmann, Harald Clahsen, Richard Wiese, Andreas Woest, & Steven Pinker (1995). German inflection: The exception that proves the rule. *Cognitive Psychology*, 29, 189–256.
- McClelland, James L. (1988). Connectionist models and psychological evidence. *Journal of Memory and Language*, 27, 107–123.
- Medin, Douglas L., & Marguerite M. Schaffer (1978). Context theory of classification learning. *Psychological Review*, 85, 207–238.
- Nosofsky, Robert M. (1988). Exemplar based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14, 700–708.
- Nosofsky, Robert M. (1990). Relations between exemplar similarity and likelihood models of classification. *Journal of Mathematical Psychology*, 34, 393–418.
- Pinker, Steven (1991). Rules of language. *Science*, 253, 530–534.
- Pinker, Steven (1997). Words and rules in the human brain. *Nature*, 387, 547–548.
- Pinker, Steven, & Alan Prince (1994). Regular and irregular morphology and the psychological status of rules of grammar. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 321–351). Amsterdam: Benjamins.
- Real Academia Española (1995). *Diccionario de la lengua española*. 21st edition, CD ROM version. Madrid: Espasa-Calpe.
- Riesbeck, Chris K., & Roger S. Schank (1989). *Inside case-based reasoning*. Hillsdale, NJ: Erlbaum.
- Sebastián Gallés, Núria, M. Antònia Martí Antonín, Manuel Francisco Carreiras Valiña, & Francisco Cuetos Vega (2002). *LEXESP: Léxico informatizado del español*. Barcelona: Edicions Universitat de Barcelona.
- Seidenberg, Mark S. (1992). Connectionism without tears. In Steven Davis (Ed.), *Connectionism: Theory and practice* (pp. 84–122). New York: Oxford University Press.
- Seidenberg, Mark, & Maggie Bruck (1990). Consistency effects in the generation of past tense morphology. Paper presented at the annual meeting of the American Psychonomic Society, New Orleans.
- Shanks, David R. (1995). *The psychology of associative learning*. Cambridge: Cambridge University Press.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1995). Analogy: A non-rule alternative to neural networks. *Rivista di Linguistica*, 7, 213–232.
- Zamora, Juan Clemente (1975). Morfología bilingüe: La asignación de género a los préstamos. *Bilingual Review*, 2, 239–247.



## Appendix

### Stimulus materials

abarraz	canez	estruz
acates	carauz	evagación
acemite	ceción	fabledad
acordación	celtre	fenestraje
acumen	cifaque	fluxión
afer	cipión	folguín
afice	cobil	fosal
alancel	coce	gafez
alcaduz	cocadriz	gagate
alcalifaje	compage	garifalte
alcamiz	consuetud	grasor
alinde	consulaje	gubilete
alioj	copanete	guiaje
alizace	cotrofe	ingre
amarillor	crenche	jusente
anascote	criazón	lailán
arrafiz	crochel	lande
asperez	chivitol	lavajal
atarfe	delate	lerdez
avarientez	desdón	linamen
azcón	deslate	mandrial
azoche	destín	mansuetud
balizaje	disfrez	másticis
barrunte	egestión	menge
beudez	elébor	meridión
bitumen	emiente	merode
bocacín	entalle	nacre
botor	entenzón	orebce
broznedad	epiglosis	palude
cabción	escambrón	panol
cabrial	escorche	paraile
cafiz	escrocón	pernicie
calicud	esgambete	pólex
calonge	esguarde	primaz
cambil	esledor	
candelor	estipe	

pujés

realme

rebalaj

riste

senojil

sorce

sozprior

tabelión

trascol

velambre

venadriz

venderache



# A comparison of Analogical Modeling to Memory-Based Language Processing\*

Walter Daelemans

## 1. Introduction

Memory-Based Language Processing is inspired by the hypothesis that in learning a cognitive task from experience, people do not extract rules or other abstract representations from their experience, but reuse their memory of that experience directly. For language behavior modeling, this means that *language acquisition* involves the storage of experiences in memory, and *language processing* is the result of analogical reasoning on memory structures. Whereas the inspiration and motivation for our approach to MBLP has come mainly from statistical pattern recognition and Artificial Intelligence, a similar approach has also survived the Chomskyan revolution in linguistics, most notably in the work of Royal Skousen on Analogical Modeling. After presenting a short history and characterization of both MBLP and AM in this section, we will discuss the main algorithmic differences in Section 2, and study their effects in Section 3 in a comparative study using the German plural as a benchmark task. Section 4 discusses theoretical implications of the results.

### 1.1 Memory-Based Language Processing

As far as the algorithms used in MBLP are concerned, nearest neighbor methods ( $k$ -NN), developed in statistical pattern recognition from the 1950s onwards, have played an important inspirational role (Fix & Hodges 1951; Cover & Hart 1967). In these methods, examples (labeled with their class) are represented as points in an example space with dimensions defined by the numeric attributes used to describe the examples. A new example obtains its class by finding its position as a point in this space, and extrapolating its class from the  $k$  nearest points in its neighborhood. Nearness is defined in terms of Euclidean distance. This literature has also generated many studies on methods for removing examples from memory either for efficiency (faster processing by removing unnecessary examples) or for accu-

racy (better predictions for unseen cases by removing badly predicting examples). (See Dasarathy 1991 for a collection of fundamental papers on  $k$ -NN research.) However, until the 1980s, the impact of these nonparametric statistical methods on the development of systems for solving practical problems has remained limited because of a number of shortcomings: they were computationally expensive in storage and processing, intolerant of attribute noise and irrelevant attributes, and sensitive to the similarity metric used; and the Euclidean distance metaphor for similarity breaks down with non-numeric features or when features are missing.

From the late 1980s onwards, the intuitive appeal of the nearest neighbor approach has been adopted in Artificial Intelligence in many variations on the basic nearest neighbor modeling idea, using names such as memory-based reasoning, case-based reasoning, exemplar-based learning, locally-weighted learning, and instance-based learning (Stanfill & Waltz 1986; Cost & Salzberg 1993; Riesbeck & Schank 1989; Kolodner 1993; Atkeson, Moore, & Schaal 1997; Aamodt & Plaza 1994; Aha, Kibler, & Albert 1991). These methods modify or extend the nearest neighbor algorithm in different ways, and aim to solve (some of) the problems with  $k$ -NN listed before. Recently, the term *Lazy Learning* (as opposed to *Eager Learning*) has been proposed as a generic term for this family of methods (Aha 1997).

Since the early 1990s, we find several studies using nearest-neighbor techniques for solving problems in Natural Language Processing (Cardie 1996; Daelemans, van den Bosch, & Zavrel 1999). The general approach is to define the tasks as (cascades of) classification problems. For each (sub)problem, instances are collected of input linguistic items and their context, plus an associated output linguistic class. The German plural prediction task to be discussed later adheres to this format. The spectrum of language processing tasks that has been investigated within this framework ranges from phonology to semantics and discourse processing (see Daelemans 1999 for a recent overview).

A related framework is DOP (Data-Oriented Parsing), a memory-based approach to syntactic parsing (Scha, Bod, & Sima'an 1999), which uses a corpus of parsed or semantically analyzed utterances (a treebank) as a representation of a person's language experience, and analyzes new sentences searching for a recombination of subtrees that can be extracted from this treebank. The frequencies of these subtrees in the corpus are used to compute the probability of analyses.

In another related tradition, Nagao (1984) proposed Example-Based Machine Translation (EBMT), an approach to Machine Translation which is essentially memory-based. By storing as exemplars a large set of (analyzed) sentences or sentence fragments in the source language with their associated translation in the target language, a new source language sentence can be translated by finding exemplars in memory that are similar to it in terms of syntactic structure and word meaning, and extrapolating from the translations associated with these

examples. Especially in the UK and Japan, this approach has become an important subdiscipline within Machine Translation research.

## 1.2 Analogical Modeling

Since Chomsky replaced the vague notions of analogy and induction existing in linguistics in his time (in the work of e.g. de Saussure & Bloomfield) by the clearer and better operationalized notion of rule-based grammars, most mainstream linguistic theories, even the functionally and cognitively inspired ones, have assumed rules to be the only or main means to describe any aspect of language.

In contrast, Royal Skousen (1989, 1992) argues for a specific operationalization of the pre-Chomskyan analogical approach to language and language learning called Analogical Modeling (AM). He introduced a definition of analogy that is not based on rules and that does not make a distinction between regular instances (obeying the rules) and irregular instances (exceptions to the rules). To model language acquisition and processing, a database of examples of language use is searched looking for instances analogous to a new item, and extrapolating a decision for the new item from those examples.

Current research on AM attempts to solve the computational complexity problem (the algorithm is exponential in the number of attributes used to describe examples) and to apply the approach to a wide range of linguistic problems. The work has also been taken up as a psycholinguistically relevant explanation of human language acquisition and processing, especially as an alternative to *dual route* models of language processing (Eddington 2000; Chandler 1992; Derwing & Skousen 1989). AM has also been used in computational linguistics. Jones (1996) describes an application of AM in Machine Translation, and work by Deryle Lonsdale includes AM implementations of part-of-speech tagging and sentence boundary detection.

While AM is the most salient example of analogy-based theories in linguistics (and the most interesting from a computational point of view), other linguists outside the mainstream have proposed analogical processing. For example, in discussion about the storage versus computation trade-off in models of linguistic processing, linguists like Bybee (1988) and usage-based linguistic theories such as Cognitive Grammar (Langacker 1991) claim an important role for examples (instances of language use); nonetheless, they still presuppose rules to be essential for representing generalizations.

## 2. A comparison of algorithms

Whereas *AM* refers to a single algorithm, there are various possible ways in which ideas in *MBLP* can be operationalized in algorithmic form. In the remainder of this text, we will narrow down our discussion of *MBLP* to the specific incarnation of it that has been used intensively in Tilburg and Antwerp. Although our specific approach to *MBLP* was developed primarily with language engineering purposes in mind, its linguistic and psycholinguistic relevance like in *AM* has always been a focus of attention. As an example, work on word stress acquisition and processing in Dutch has contrasted *MBLP* with metrical phonology and studied correlations between errors made by a memory-based learner and those made by children producing word stress in a repetition task (Daelemans, Gillis, & Durieux 1994; Gillis, Durieux, & Daelemans 2000). Many of the properties which make *AM* cognitively and linguistically plausible also apply to *MBLP*: (i) there is no all-or-none distinction between regular cases and irregular cases because no rules are used; (ii) fuzzy boundaries and leakage between categories occurs; (iii) the combination of memory storage and similarity-based reasoning is cognitively simpler than rule-discovery and rule processing; and (iv) memory-based systems show adaptability and robustness. Remarkably, seen from the outside, such analogical or memory-based approaches appear to be rule-governed, and therefore adequately explain linguistic intuitions as well.

Both approaches are instances of the same general view of cognitive architecture. However, because of the different algorithms used to extrapolate outcomes from stored occurrences, the properties and behavior of both approaches may differ considerably in specific cases.

### 2.1 Similarity in *MBLP*

The most basic metric that works for patterns with symbolic features as well as for numeric features, is the *overlap metric* given in Equations (1) and (2), where  $\Delta(X, Y)$  is the distance between patterns  $X$  and  $Y$ , represented by  $n$  features, and  $\delta$  is the distance per feature. The distance between two patterns is simply the sum of the differences between the features. The  $k$ -NN algorithm with this metric is called *IB1* (Aha, Kibler, & Albert 1991).<sup>1</sup> Usually  $k$  is set to 1.

$$\Delta(X, Y) = \sum_{i=1}^n \delta(x_i, y_i) \quad (1)$$

where

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{\max_i - \min_i} & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \quad (2)$$

The distance metric in Equation (2) simply counts the number of (mis)matching feature-values in both patterns. In the absence of information about feature relevance, this is a reasonable choice. However, we can do better by computing statistics about the relevance of features by looking at which features are good predictors of the class labels. Information Theory gives us a useful tool for measuring feature relevance in this way. *Information Gain* (IG) weighting looks at each feature in isolation and measures how much information it contributes to our knowledge of the correct class label (Quinlan 1993). The Information Gain of feature  $i$  is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature:

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v), \quad (3)$$

where  $C$  is the set of class labels,  $V_i$  is the set of values for feature  $i$ , and  $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$  is the entropy of the class labels. The probabilities are estimated from relative frequencies in the training set. For numeric features, values are first discretized into a number (the default is 20) of equally spaced intervals between the minimum and maximum values of the feature. These groups are then used in the IG computation as if they were discrete values. (Note that this discretization is not used in the computation of the distance metric.) The  $k$ -NN algorithm with this metric is called IB1-IG (Daelemans & van den Bosch 1992). (For more references and information about the algorithms, refer to Daelemans, van den Bosch, & Weijters 1997; Daelemans, Zavrel, van der Sloot, & van den Bosch 1998; and Daelemans, van den Bosch, & Zavrel 1999.)

For most of our experiments in the past, IB-IG with extrapolation based on one nearest neighbor ( $k = 1$ ) has been the default MBLP algorithm. Note that setting  $k = 1$  may imply extrapolation from more than one exemplar in memory; in case there is more than one exemplar which is the nearest neighbor, the algorithm uses all of them for extrapolation and selects the class which appears most often (or the overall most frequent class in case of ties). In what follows, we will use both IB1-IG (the particular incarnation) and MBLP (the general approach) to refer to our approach, depending on the context.



## 2.2 The AM extrapolation algorithm

The main algorithmic difference between AM and MBLP is the way the selection of memory items to extrapolate from is made. In IB1-IG, the different features are assigned a relative importance, which is used during matching to filter out the influence of irrelevant features. In AM, essentially the same effect is achieved without precomputing the relative importance of individual features.<sup>2</sup> Instead, all features are equally important initially, and serve to partition the database into several disjoint sets of examples. Filtering out irrelevant exemplars is done by considering properties of these sets rather than by inspecting individual features that their members may share with the input pattern. To explain how this works, we will describe the matching procedure in more detail.<sup>3</sup>

The first stage in the matching process is the construction of *subcontexts*; subcontexts are sets of examples, and they are obtained by matching the input pattern, feature by feature, to each item in the database, on an equal / not equal basis, and classifying the database exemplars accordingly. Taking an input pattern ABC as an example, eight ( $2^3$ ) different subcontexts would be constructed, ABC,  $\bar{A}BC$ ,  $A\bar{B}C$ ,  $AB\bar{C}$ ,  $\bar{A}\bar{B}C$ ,  $\bar{A}B\bar{C}$ ,  $A\bar{B}\bar{C}$ , and  $\bar{A}\bar{B}\bar{C}$ , where the overstrike denotes complementation. Thus, exemplars in the class ABC share all their features with the input pattern, whereas for those in  $\bar{A}BC$  only the value for the third feature is shared. In general,  $n$  features yield  $2^n$  mutually disjoint subcontexts. Subcontexts can be either *deterministic*, which means that their members all have the same associated category, or *non-deterministic*, when two or more categories occur.

In the following stage, *supracontexts* are constructed by generalising over specific feature values. This is done by systematically discarding features from the input pattern, and taking the union of the subcontexts that are subsumed by this new pattern. Supracontexts can be ordered with respect to generality, so that the most specific supracontext contains examples which share all  $n$  features with the input pattern, less specific supracontexts contain items which share at least  $n - 1$  features, and the most general supracontext contains all database exemplars, whether or not they have any features in common with the input pattern. In the table below the supracontexts for our previous example are displayed, together with the subcontexts they subsume.

Supracontext	Subcontexts
A B C	ABC
A B –	ABC $AB\bar{C}$
A – C	ABC $A\bar{B}C$
– B C	ABC $\bar{A}BC$
A – –	ABC $A\bar{B}\bar{C}$ $AB\bar{C}\bar{C}$ $\bar{A}\bar{B}\bar{C}$
– B –	ABC $\bar{A}BC$ $AB\bar{C}\bar{C}$ $\bar{A}\bar{B}\bar{C}$

- – C                    ABC  $\bar{A}$ BC A $\bar{B}$ C  $\bar{A}\bar{B}$ C
- – –                    ABC  $\bar{A}$ BC A $\bar{B}$ C A $\bar{B}$ C  $\bar{A}\bar{B}$ C  $\bar{A}\bar{B}$ C A $\bar{B}$ C  $\bar{A}\bar{B}$ C

An important notion with respect to supracontexts is *homogeneity*. A supracontext is called homogeneous when any of the following conditions holds:

- The supracontext contains nothing but empty subcontexts (trivial).
- The supracontext contains only deterministic subcontexts with the same category.
- The supracontext contains only one non-empty, non-deterministic subcontext.

Heterogeneous supracontexts are obtained by combining deterministic and non-deterministic subcontexts. Going from least to most general, this means that as soon as a supracontext is heterogeneous, any more general supracontext will be heterogeneous too.

In the final stage, the analogical set is constructed. This set contains all of the exemplars from each of the homogeneous supracontexts. Two remarks are in order here. First, since some exemplars will occur in more than one supracontext, each exemplar is weighted according to its distribution across different homogeneous supracontexts. This is accomplished by maintaining a score for each exemplar. This score (under the choice of linearity) is simply the summed cardinality of each of the supracontexts in which the exemplar occurs. (Another choice, the quadratic one, involves multiplying the score by the frequency of the supracontext.) The motivation for this scoring mechanism is to favor frequent patterns over less frequent ones and patterns closer to the input pattern over more distant patterns, since the former will surface in more than one supracontext. Second, banning heterogeneous supracontexts from the analogical set ensures that the process of adding increasingly dissimilar exemplars is halted as soon as those differences may cause a shift in category. Exactly when this happens depends on the input pattern and the data.

To finally categorize the input pattern, either the predominant category in the analogical set (selection by plurality) or the category of a randomly chosen member of this set is selected.

### 2.3 AM versus MBLP

The different way in which IBI-IG and AM construct a set of exemplars to extrapolate from leads to a number of differences which have sometimes been advanced as an advantage or disadvantage for one or the other approach (Daelemans, Gillis, & Durieux 1997). We will list these differences here, and discuss them in the context of our results in Section 4.

1. Non-neighbors can affect language behavior in AM, not in IBI-IG.

2. Because of the method of constructing contexts, AM can locally determine the significance of groups of variables (feature values), whereas these are lost in the averaging over values when using information gain in IB1-IG.
3. The feature weighting in IB1-IG constitutes a type of preprocessing or learning which is unnecessary in AM.
4. The natural statistic on which AM is based can make possible the use of only a percentage of the data (imperfect memory) for optimal accuracy and robustness, whereas for IB1-IG “forgetting exceptions is harmful to language learning” (Daelemans, van den Bosch, & Zavrel 1999).
5. AM is exponential in the number of cases to consider, IB1-IG is linear in this number.
6. AM has no natural extension to numeric data (but see Chapter 15 of Skousen 1992), whereas the overlap metric used in IB1-IG can be easily generalized to different types of feature values (numeric, set-valued).

### 3. A test comparison: German plural

The diachrony of plural formation of German nouns has led to a notoriously difficult system, which is nevertheless routinely acquired by speakers of German. Because of the complex interaction (from a synchronic point of view) of regularities, subregularities, and exceptions, it is to be expected that lexicon-based methods such as AM and IB1-IG do well in this case, and that it is an interesting testing ground for comparing them.

There is another reason why the German plural is an interesting problem. Marcus and his colleagues (Clahsen 1999; Marcus, Brinkmann, Clahsen, Wiese, & Pinker 1995) have argued that this task provides evidence for the *dual route* model for cognitive architectures. A dual route architecture supposes the existence of a cognitively real productive mental default rule, and an associative memory for irregular cases which blocks the application of the default rule. They argue that *-s* is the regular plural in German, as this is the suffix used in many conditions associated with regular inflection (e.g., novel words, surnames, acronyms, etc.). This default rule is applied whenever associative memory-lookup fails. The case of German plurals provides an interesting new perspective to what is *regular*: in this case, the default rule (regular route) is less frequent than many of the ‘irregular’ associative memory cases. In a plural noun suffix type frequency ranking (see below), *-s* comes only in last place. Perhaps the behavior of AM and IB1-IG as *single route* models offers some additional insight into this phenomenon.

We collected 25,753 German nouns from the German part of the CELEX-2 lexical database.<sup>4</sup> We removed from this dataset cases without plurality marking,

**Table 1.** Data characteristics of German Plural experiments

Feature	Number of values	Example: <i>Vorlesung</i>
Onset penultimate	78	l
Nucleus penultimate	27	e
Coda penultimate	85	-
Onset last	84	z
Nucleus last	27	U
Coda last	79	N
Gender	10	F
Class	8	-en

**Table 2.** Type frequency of pluralization mechanisms in CELEX

Class	Frequency	Umlaut	Frequency	Example
(e)n	11920			Abart
e	6656	no	4646	Abbau
		yes	2010	Abdampf
-	4651	no	4402	Aasgeier
		yes	249	Abwasser
er	974	no	287	Abbild
		yes	687	Abgang
s	967			Abonnement

cases with Latin plural in *-a*, and a miscellaneous class of foreign plurals. From the remaining 25,168 cases, we extracted or computed for each word the plural suffix, the gender feature, and the syllable structure of the two last syllables of the word in terms of onsets, nuclei, and codas (expressed with a phonetic segmental alphabet). Table 1 gives an overview of the features, values, and output classes considered in these experiments. The gender feature, apart from masculine (M), neuter (N), and feminine (F), also has all possible combinations of two genders.

Table 2 lists the possible output classes with their type frequency in the dataset. There was no further preprocessing of the data. A well-known source of *noise* in the CELEX data are plain mistakes in lexical coding. However, we expect learning methods to be robust to this type of noise, and did not attempt to find and correct these coding errors.

In order to compare the accuracy of AM and 1B1-IG on the German plural task, we performed several learning experiments. We compared the learnability of the task, varying the training set size for the complete task and for the different suffixes separately. We also performed an error analysis and comparison, and we looked at the influence of some different parameter settings on algorithm accuracy.

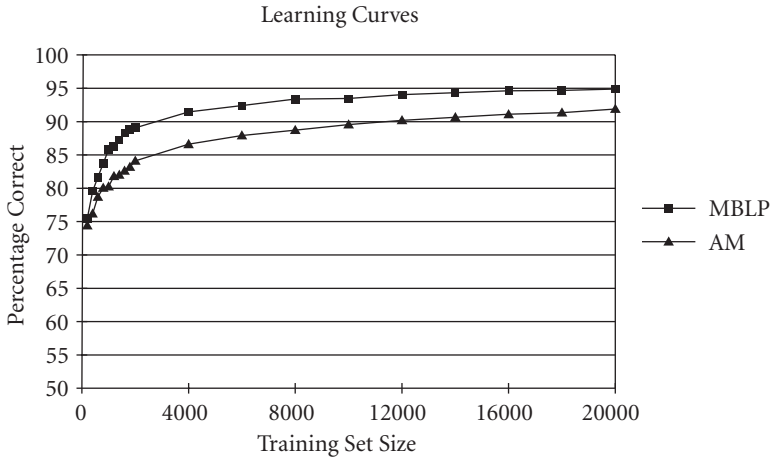


Figure 1. Learnability of German Plural with MBLP and AM

### 3.1 Learnability

In an initial learnability experiment, we randomized the dataset, selected a 5,168 word test set, and divided the remaining 20,000 words into 19 training sets with an incrementally increasing size from 200 to 2,000 in steps of 200, and from 2000 up to 20,000 in steps of 2,000. Each of the algorithms was then trained with each of the training sets and tested each time on the single test set. Figure 1 shows the learning curve for both algorithms when using their standard settings, i.e. IB1-IG with information gain and  $k = 1$  for MBLP, and AM with perfect memory and with selection by plurality.

We see that for small training sets, AM performs about the same as MBLP, but a statistically significant divergence in favor of MBLP starts after 1000 training items. Although accuracy is still increasing for both algorithms with 20,000 training cases, learning seems to come near to its upper bound already at around 2,000 training cases.

In Figures 2 and 3 the learning curves of the individual plural formation classes are shown for AM and IB1-IG, respectively.<sup>5</sup> Interestingly, for both algorithms, the suffixes seem to fall into three classes: those that are learned correctly from the start (*-en* and *-*), those that require longer learning but are learned very well in the end (*-e* and *-er*), and one which is never learned very well at all (*-s*), although accuracy is increasing with number of training items. It seems indeed to be the case that *-s* behaves differently from the other suffixes, when learned by single-route models such as AM and IB1-IG. However, this does not necessarily lend credence to a *dual route* model for the German plural. The learning curves clearly show that the suffix

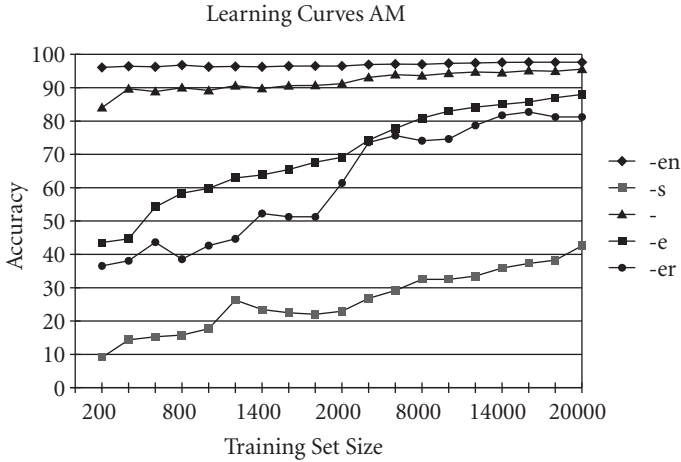


Figure 2. Learnability of German Plural classes with AM

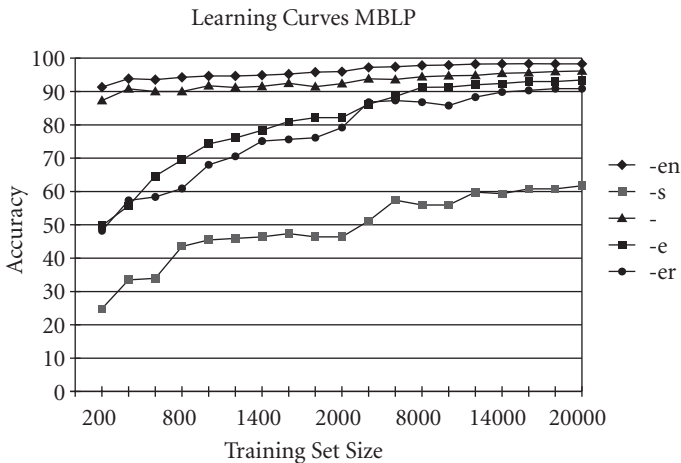


Figure 3. Learnability of German Plural classes with MBLP

is learned by single route models as well (at least some generalizations about when to use *-s* are learned), and 60% accuracy (for IB1-IG) is a respectable result given the limited information provided in the input representations. It is by no means inconceivable that additional semantic or syntactic features could further improve learnability of *-s* with the single route models discussed here. The only conclusion that can be drawn from these experiments in this regard is that whereas the other

suffixes are learnable from syllable structure and gender information, this is not the case for *-s*.

For those suffixes which are sometimes accompanied by an umlaut, there is no marked difference in the speed of learning and accuracy achieved for versions with and without umlaut. For the different suffixes, we see that AM learning is slower and reaches lower accuracies, except for the *-en* suffix which is learned very well from the start by AM.

### 3.2 Error analysis

In order to generate more data for a comparison between AM and MBLP on the German plural data, we performed a leave-one-out experiment using both algorithms. In such a set-up, each instance in the data file is held out in turn as a test item, and all remaining instances act as training material to train the classifier. In machine learning methodology, the leave-one-out method is generally accepted as the best estimator for the “real” error of a classifier. The advantage of using it in this context, is that we have access to the complete dataset to look for trends or examples. For both algorithms, we again used the default parameter settings. Table 3 shows the accuracy on the full dataset using this method for AM and MBLP distributed over the different suffixes. For clarity, we repeat the frequencies of Table 2 above on the first line for each suffix type.

The high accuracies found in both algorithms are partly due to exact matches in memory: several different words can have the same syllable structure for their last two syllables and the same gender. Disregarding these cases (i.e., using only

**Table 3.** Accuracy of AM vs. MBLP on the complete data set using leave-one-out

Suffix	MBLP Accuracy (%)	AM Accuracy (%)	Frequency
-	96.5	96.1	4651
no umlaut	96.5	96.1	4402
umlaut	96.8	96.4	249
-e	92.5	87.0	6656
no umlaut	92.1	88.2	4646
umlaut	93.3	84.3	2010
-er	92.7	81.5	974
no umlaut	92.7	79.4	287
umlaut	92.7	82.4	687
en	98.3	97.7	11920
s	66.9	46.7	967
Total	95.0	92.0	25168

unique combinations of feature set and class as data) gives an overall accuracy of 89.7% for MBLP and 86.6% for AM with roughly the same distribution of accuracies over the different suffixes. In the remainder of this paper, we will work with the results for the dataset *with* duplications of lexical representations.

For 92.5% of the words, both systems agree on the outcome, and assuming the outcome in the CELEX database to be correct, for 90% of the words they agree on the correct class. Of the 555 cases in which both algorithms predict the same but wrong class, the majority is due to words with plural suffix *-s*, being assigned to *-e* or *-(e)n*: e.g., *Autocar, Bar, Jeep, Sheriff* (*-e* instead of *-s*); *Backhand, Fondue, Tape* (*-(e)n* instead of *-s*). But many other confusions occur as well. See Tables 4 and 5 for a complete overview. In these tables, confusion between the different outcomes

**Table 4.** Confusion matrix for AM. Indicates how many times an exemplar of type (as indicated in the rows) was classified as type (as indicated in the columns). Correct predictions are on the diagonal.

	-	U	Ue	Uer	e	en	er	s	
-	4191	46	9	3	44	48	5	56	4402
U	78	171	0	0	0	0	0	0	249
Ue	4	0	1893	0	82	26	0	5	2010
Uer	0	0	7	643	31	5	0	1	687
e	33	0	79	30	4318	118	9	59	4646
en	35	13	32	3	103	11708	0	26	11920
er	1	0	0	2	14	0	270	0	287
s	64	1	12	7	153	74	2	654	967
	4406	231	2032	688	4745	11979	286	801	25168

**Table 5.** Confusion matrix for MBLP. Indicates how many times an exemplar of type (as indicated in the rows) was classified as type (as indicated in the columns). Correct predictions are on the diagonal.

	-	U	Ue	Uer	e	en	er	s	
-	4231	11	7	1	54	74	2	22	4402
U	8	240	0	0	0	1	0	0	249
Ue	31	0	1694	2	113	167	0	3	2010
Uer	15	0	22	566	70	12	0	2	687
e	104	14	139	32	4097	214	11	35	4646
en	62	6	29	3	159	11649	1	11	11920
er	11	0	0	0	34	13	228	1	287
s	73	3	30	5	185	218	1	452	967
	4535	274	1921	609	4712	12348	243	526	25168



(classes) is represented. *U* means umlaut: e.g., *Uer* is the class of nouns with plural in *-er* and with umlaut; *er* is the class of nouns with plural in *-er* without umlaut.

If we compare the confusion matrices of both systems, we see that they are almost indistinguishable in the confusions made. The Spearman correlation coefficient is 0.999 when taking into account all cells (correct predictions as well as errors). When limited to errors, the correlation is still 0.83 suggesting that both systems make the same confusions. Nevertheless, some of the error categories indicate more divergence: for the cases of grammatical conversion (no suffix is added; *-* and *U* in the confusion matrices), the errors made by both algorithms differ more markedly, both the confusion made when assigning an incorrect class to these cases (Pearson correlation 0.64) and the type of cases to which conversion is incorrectly assigned (Pearson correlation 0.41). AM especially seems to mistake words much more often for a *-* or *U* case than IB1-IG, especially words which should have received an *-e* plural.

For example, *Almosenier* generates an AM analogical set with the distribution (Uer:0, en:6, Ue:71, -:2880, U:0, er:0, e:834, s:72), whereas IB1-IG finds 3 neighbors at distance 0.3, all with the correct suffix *-e* (*Harpunier*, *Pionier*, *Kanonier*; all with masculine gender and ending in *-ier*). Clearly, looking at local neighborhood only, in combination with assigning more weight to the rhyme of the last syllable and the gender, provides the right sub-generalization here for MBLP.

For all other confusions, correlation is near to or much higher than 0.90, indicating very similar language behavior of both algorithms, except that AM makes significantly more errors than IB1-IG in absolute terms.

Moving on to other errors made by the algorithms, we see that there are 499 words where AM is correct and MBLP wrong, and more than twice that many (1190 words) where the reverse holds. When we look at the clustering of errors in these sets of words, we see that even here there is a positive correlation between the types of confusions AM and IB1-IG make when their counterpart is correct.

We have to conclude that, at least for this problem, we find no evidence that the way the AM algorithm works leads to qualitatively different language behavior compared to that when using the conceptually and computationally simpler IB1-IG algorithm. The former leads to significantly lower accuracy, however, and seems to miss certain sub-regularities in the data.

### 3.3 Related research

We are not the first to apply these methods to the German plural problem. In Nakisa and Hahn 1996 and Nakisa, Plunkett, and Hahn 2000, simulation results on CELEX data are reported for nearest neighbour (comparable to IB1, i.e. no feature relevance weighting), Nosofsky's Generalized Context Model (GCM), and a

standard three-layer backprop network. The set-up of the experiment is similar to ours (predicting plural class from phonology) but not comparable because of (1) the different data-preprocessing steps resulting in other sets of examples and classes, (2) a different encoding of the phonology (phonetic features instead of segmental syllable structure and no gender), and (3) a different methodology, viz. cross-validation instead of leave-one-out. Results were 70.8% for nearest neighbor, 74.3% for GCM, and 82.7% for backprop.

In Wulf 1996, AM is also applied to the German plurals problem. Based on a dataset of 703 frequent words, with exemplars encoding phonology and gender, he was able to show gang effects and the results of heterogeneity on selected nouns. No report was given of the accuracy of how much of the data was accounted for.

Daelemans, Gillis, and Durieux (1997) compared AM and several variants of MBLP on the task of main stress assignment for Dutch. They found that whereas AM outperforms IBI, variants such as IBI-IG outperform AM and are more insensitive to noise. The only other comparison of AM and MBLP we know of (Eddington 2000) focused on comparing both as a possible alternative implementation of a single-route model for past tense morphology to connectionist models, and reported similar results for both when testing on non-words for the past tense, but found AM sometimes working better to predict specific language usage.

## 4. Discussion

In this section, we refer back to the list of differences noted in Section 2, and discuss these, armed with our new results.

### 4.1 The effect of non-neighbors

In AM, non-neighbors frequently affect the decision of the algorithm, as we have seen. MBLP on the other hand relies on local extrapolation: a small neighborhood (typically the nearest neighbor only) is used to extrapolate from. We see that for the German plural at least, the MBLP strategy seems fruitful (e.g., in discovering the subregularity that the plural suffix of masculine nouns in *-ier* is *-e*). There are 32 cases like that with only two exceptions: *Sire* /zi:r/, plural *Sires*, and *Partikulier*, also with plural *-s*. Of these 32 cases, 28 were classified correctly by IBI-IG (the four errors being *Wesir* and *Kurier* which were pluralized *-s*, and *Sire* and *Partikulier*, classified as *+e*). On the other hand AM makes these errors as well, and in addition 6 other errors, including “clear” cases of the subregularity, such as *Almosenier*, *Fakir*, *Kanonier*, *Kurier* as well as *Kashmir*, *Mudir*.

The problem that the AM algorithm tries to solve by permitting the algorithm, if needed, to look at the complete dataset and by classifying subsets of the data as homogeneous and heterogeneous, and that IB1-IG tries to solve by estimating the information gain of each feature, is the problem of *representation relevance*. Which features are most relevant for solving the task? IB1-IG reorganizes the exemplar space (and therefore the distances in it leading to extrapolation of outcomes) by feature weighting. In principle, it is possible to extend the IB1-IG algorithm such that it takes into account all exemplars in memory, by setting the value of  $k$  to the number of exemplars, and using the inverse of their distance to the input item to weigh their importance in computing the outcome, but this seldomly leads in practice to better accuracy.

This reliance of IB1-IG on similarity-space reorganization by means of feature weighting makes the approach of course potentially vulnerable to bad relevance assignments for some features. For example, a known problem with information gain is that it computes the relevance of a feature without taking into account the other features, ignoring possible feature interactions. However, for this problem (and many other linguistic problems we have investigated), it is an accurate and robust heuristic method.

Figure 4 shows the relevance assignment of a few different feature weighting methods on our dataset. Gain ratio is a normalized version of information gain (boosting the relevance of features with few values); the  $\chi^2$  method uses statistical significance testing to compare the observed distributions of values over classes with their expected distribution (Daelemans, van den Bosch, & Zavrel 1999). Interestingly, while the relevance assignment is roughly similar, there are some marked

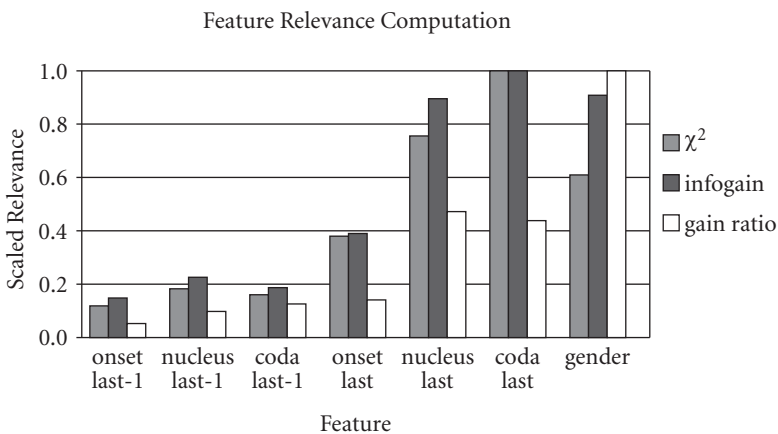


Figure 4. Feature weights using different weighting methods

**Table 6.** Effect of feature relevance assignment method on accuracy in IB1

Weighting method	Accuracy (%)
BASELINE1	46.3
BASELINE2	32.1
IB1	92.6
IB1-IG	95.0
IB1- $\chi^2$	95.1
IB1-GR	94.9

differences, e.g., gain ratio puts more weight on the gender feature and estimates the relevance of the segmental information lower than the other two methods.

The effect when using these methods in a  $k$ -NN algorithm with  $k = 1$  on our data (using leave-one-out methodology) is summarized in Table 6. The differences are not important, showing that MBLP is fairly robust to the details of the algorithm for this problem. As could be expected, the algorithm using no reorganization of the exemplar space at all (IB1) performs significantly worse than any of the weighted methods, but it is surprising to see that it outperforms AM. This indicates that all pre-selected features are indeed relevant to solving the task, and that the role of the feature weighting method is in fine-tuning the organization of the exemplar space rather than in re-organizing it. The table also lists the baseline accuracy when always selecting the most frequent suffix  $-(e)n$ , and when probabilistically guessing the outcome (knowing only the distribution of the different classes), called BASELINE1 and BASELINE2, respectively.

#### 4.2 Value relevance weighting

Another potential problem for IB1-IG is the frequency-weighted averaging of the information gain of the different values of a feature to compute the information gain of the feature. This is a source of robustness (since estimation is on the complete dataset), but may at the same time lead to unwarranted underestimation of the relevance of some feature values for some inputs, snowed under in the averaging. Because of the way the algorithm works (treating each value as distinct), AM can assign more or less importance to particular values relative to the particular input it is classifying. In Skousen's issues article in this volume, an example from Finnish past tense (*sorta-* 'to oppress') is worked out in detail, and it is indeed the case that IB1-IG incorrectly handles this item. However, this is a representation problem more than an algorithmic problem. If a particular value has a high relevance for some types of inputs, it should be assigned a separate feature. It is even

possible to explode all values of all features into separate binary features, and use general feature relevance weighting methods on this new representation. This way, the particular Finnish past tense problem can also be solved by IB1-IG (van den Bosch, personal communication).

Furthermore, whereas it will probably be possible to find similar cases also for the German plural, there will be plenty (60% more) errors made by AM which IB1-IG does not make. In the comparison of the linguistic adequacy of algorithms, the overall accuracy levels are probably more important than explaining individual cases. This is of course not the case for psycholinguistic models; here the algorithms and feature relevance metrics should be compared with human performance and acquisition (see e.g., Eddington 2000), and overall accuracy is no longer the main evaluation criterion.

### 4.3 Feature weighting as training

Yet another criticism of IB1-IG (see Section 5 of Skousen's overview of AM in this volume) is that because of the feature weighting method used, a training period is needed which makes the approach more akin, in this respect only, to connectionism than to AM. The important distinction here is that whereas connectionist learning methods such as backpropagation of errors are *batch-learning* methods (cycling several times through all training items until an equilibrium or desired error rate is reached), computing information gain is an *incremental* process and converges very quickly. For example, Figure 5 illustrates the convergence of the information gain weights in the differently sized training sets we used to compute

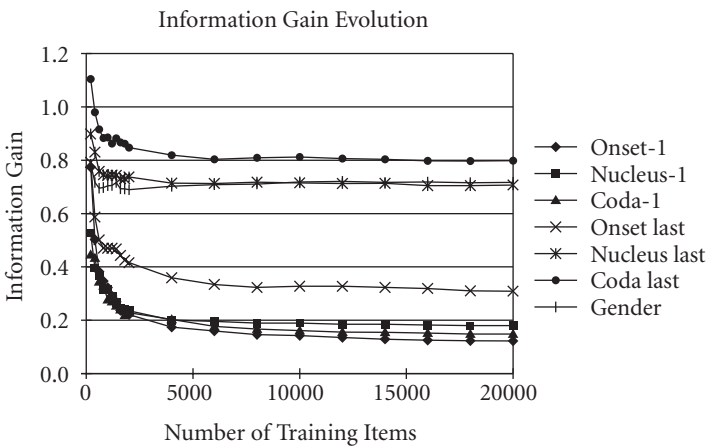


Figure 5. Convergence of information gain feature weights

the learnability results discussed earlier. Already after a few hundred training items, the information gain values are stable, and already from the very first training set, the relative ordering of the relevance of the different features remains basically the same; only the absolute values vary. In addition, the algorithm is very robust to small variations in the specific values of the information gain weights.

#### 4.4 (Im)perfect memory

In language processing tasks, low-frequency events are pervasive. Due to borrowing, historical change, and the complexity of language, most data sets representing language processing tasks contain few regularities, but many subregularities and exceptions. These exceptions and subregularities only concern a limited number of cases, yet in their small ‘pocket of exceptions’ in exemplar space they are productive in that they may correctly predict the outcome for a previously unseen member of their region. It is impossible for inductive algorithms to reliably distinguish real noise from these pockets of exceptions, so non-abstracting algorithms like 1B1-IG should be at an advantage compared to eager-learning methods such as decision tree learning or rule induction: ‘forgetting exceptions is harmful’. In Daelemans, van den Bosch, & Zavrel 1999 results are provided, with theoretical analysis, supporting this hypothesis.

On the other hand, being based on a minimization of disagreements among data-items, AM is the most powerful statistical test possible, and can be made equivalent to standard statistical procedures by introducing imperfect memory (i.e., introducing a chance that a particular training item is forgotten). Interestingly, and surprisingly from the point of view of the “forgetting is harmful” hypothesis, forgetting 25% and 50% of the training data for the German plural problem does *not* decrease generalization accuracy for AM, which remains at 92%. However, as this is significantly lower than the generalization accuracy of 1B1-IG, it is unclear what this means. One explanation could be that the way the AM algorithm works on this problem is a form of noise-reduction or smoothing in which the productive subregularities and pockets of exceptions are lost against the more powerful effect of the general tendencies in the dataset. (Remember that all data items may influence the final decision, not only the local context.) The anecdotal evidence about masculine nouns in *-ier* seems to support this view, but more analysis is necessary. Forgetting part of the data may counter this hypothesised overregularization tendency of AM.

#### 4.5 Computational complexity and representational generality

In Daelemans, Gillis, and Durieux 1997 it was argued that an important advantage of MBLP as opposed to the AM algorithm is the fact that the former is linear in the number of features and exemplars, whereas the latter is exponential in the number of features. Massive parallelism does not effectively eliminate this exponential explosion. In Skousen's issues article in this volume it is argued that the information gain feature relevance weighting in IB1-IG must take into account all possible combinations of feature values (if it is to account for all language predictability), hence there is no escaping from exponential explosion. Computation of information gain is linear in the number of data items on which it is computed. (All that is necessary is a simple computation on a feature-value outcome-class contingency matrix which can be incrementally collected as experience enters the system.) Information gain *does* make the (mostly incorrect) assumption that the features are independent; it is a heuristic. Yet, the tests show that it is an effective and robust relevance estimator for linguistic problems.

Furthermore, the more general approach to similarity used in MBLP allows for the easy and natural definition of similarity for features with numeric and set values, as opposed to AM practice where usually only symbolic (nominal) and binary features are used (see Chapter 15 of Skousen 1992). Although most language processing representations can be described adequately using nominal features, some linguistic information (e.g., distances between and lengths of linguistic objects like words and utterances; and sets of words, phonemes, or letters) can be more naturally represented using numeric and set valued features.

### 5. Conclusion

AM and MBLP are similar in spirit, but propose completely different operationalizations of similarity- or analogy-based language processing on the basis of exemplars. In an earlier comparison between AM and MBLP (Daelemans, Gillis, & Durieux 1997) dealing with the task of main stress assignment in Dutch words, we concluded that for natural language learning tasks there was no clear motivation to use the complex and computationally costly (and with many features computationally intractable) AM algorithm instead of the more general and less complex class of MBLP algorithms. In this paper we added more substance to this position by analyzing the behavior of AM and IB1-IG on the task of German plural prediction. We found that IB1-IG, a simple MBLP algorithm, significantly outperforms AM, and seems to be better at representing the subgeneralizations of the task. On the other hand, both systems are highly correlated in the errors they make (i.e., the confusions between outcomes they predict), and have very similar learn-

ing behavior. Taken together with the additional expressive power and flexibility MBLP offers in handling different types of representations, we stand by our earlier conclusion.

However, additional research is needed to get more insight into the differences between both algorithms in terms of psycholinguistic and linguistic relevance. Work by Gert Durieux (e.g., Durieux, Daelemans, & Gillis 1997) suggests that AM is better at learning regularities in the Dutch stress prediction data, whereas MBLP is better at putting to use the predictive power of (small) subregularities.

## Notes

\* Research partially supported by FWO (Belgium) and NWO (The Netherlands). Many thanks to the members of ILK and CNTS for providing inspiring working environments, and to the participants to the Analogical Modeling of Language conference at Brigham Young University (22–24 March 2000) for useful discussion and comments. Special thanks to Gert Durieux for sharing his expertise about, and implementation of, AM, and for help with preprocessing the CELEX data.

1. For our experiments we have used TiMBL, available from <<http://ilk.kub.nl/>>. It is a Memory-Based Learning software package developed in our group (Daelemans, Zavrel, van der Sloot, & van den Bosch 1998). TiMBL implements a number of important memory-based algorithms and metrics. We only describe those here which we used in the experiments below.
2. The specific analogical algorithm employed by Skousen is available in a number of implementations. See the AM group's homepage at <<http://humanities.byu.edu/am/>>. For our experiments, we used an implementation by Gert Durieux, AML 0.1, available from <[durieux@ua.ac.be](mailto:durieux@ua.ac.be)>.
3. This description of the algorithm is taken from Daelemans, Gillis, and Durieux 1997.
4. Available from <<http://www ldc.upenn.edu/>>
5. In these figures, the training set sizes on the x-axis are represented as categorical values, i.e., the 200 item training sets get as much space on the x-axis as the 2000 item datasets, hence the less steep learning curve, compared to Figure 1.

## References

- Aamodt, Agnar, & Enric Plaza (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7, 39–59.
- Aha, David W. (Ed.). (1997). *Lazy learning*. Dordrecht: Kluwer Academic Publishers.
- Aha, David W., Dennis Kibler, & Marc K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Atkeson, Chris, Andrew Moore, & Stefan Schaal (1997). Locally weighted learning. *Artificial Intelligence Review*, 11, 11–73.



- Bybee, Joan (1988). Morphology as lexical organization. In M. Hammond & M. Noonan (Eds.), *Theoretical morphology: Approaches in modern linguistics* (pp. 119–141). San Diego: Academic Press.
- Cardie, Claire (1996). Automatic feature set selection for case-based learning of linguistic knowledge. In *Proceedings of conference on empirical methods in NLP*. University of Pennsylvania.
- Chandler, Steve (1992). Are rules and modules really necessary for explaining language? *Journal of Psycholinguistic Research*, 22, 593–606.
- Clahsen, Harald (1999). Lexical entries and rules of language: A multidisciplinary study of German inflection. *Behavioral and Brain Sciences*, 22, 991–1060.
- Cost, Scott, & Steven Salzberg (1993). A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10, 57–78.
- Cover, Thomas M., & Peter E. Hart (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13, 21–27.
- Daelemans, Walter (Ed.) (1999). *Memory-based language processing*, Volume 11, Number 3 (special issue) of *Journal of Experimental and Theoretical AI*. London: Taylor & Francis.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1994). The acquisition of stress: a data-oriented approach. *Computational Linguistics*, 20, 421–451.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1997). Skousen's analogical modeling algorithm: A comparison with lazy learning. In D. Jones & H. Somers (Eds.), *New methods in language processing* (pp. 3–15). London: UCL Press.
- Daelemans, Walter, & Antal van den Bosch (1992). Generalisation performance of back-propagation learning on a syllabification task. In M. F. J. Drossaers & A. Nijholt (Eds.), *Proceedings of TWLT3: Connectionism and natural language processing* (pp. 27–37). Enschede, The Netherlands: Twente University.
- Daelemans, Walter, Antal van den Bosch, & A. J. M. M. Weijters (1997). 1Gtree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11, 407–423.
- Daelemans, Walter, Antal van den Bosch, & Jakub Zavrel (1999). Forgetting exceptions is harmful in language learning. *Machine Learning: Special issue on Natural Language Learning*, 34, 11–41.
- Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, & Antal van den Bosch (1998). TiMBL: Tilburg Memory Based Learner, version 1.0, reference manual. Technical Report ILK-9803, ILK, Tilburg University.
- Dasarathy, Belur V. (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos, CA: IEEE Computer Society Press.
- Derwing, Bruce L., & Royal Skousen (1989). Real time morphology: Symbolic rules or analogical networks. *Berkeley Linguistic Society*, 15, 48–62.
- Durieux, Gert, Walter Daelemans, & Steven Gillis (1997). Empirical comparison of analogical modeling and instance-based learning. Round table on algorithms for memory-based language processing. Corsendonk, Turnhout, Belgium.
- Eddington, David (2000). Analogy and the dual-route model of morphology. *Lingua*, 110, 281–298.

- Fix, Evelyn, & J. L. Hodges, Jr (1951). Discriminatory analysis – nonparametric discrimination; consistency properties. Technical Report Project 21-49-004, Report No. 4, USAF School of Aviation Medicine.
- Gillis, Steven, Gert Durieux, & Walter Daelemans (2000). Lazy learning: A comparison of natural and machine learning of stress. In P. Broeder & J. Murre (Eds.), *Models of language acquisition: Inductive and deductive approaches* (pp. 76–99). Cambridge University Press.
- Jones, Daniel (1996). *Analogical natural language processing*. London: UCL Press.
- Kolodner, Janet L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann.
- Langacker, Ronald W. (1991). *Concept, image, and symbol: The cognitive basis of grammar*. Berlin: Mouton De Gruyter.
- Marcus, Gary F., Ursula Brinkmann, Harald Clahsen, Richard Wiese, & Steven Pinker (1995). German inflection: The exception that proves the rule. *Cognitive Psychology*, 29, 189–256.
- Nagao, Makoto (1984). A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn & R. Banerji (Eds.), *Artificial and human intelligence* (pp. 173–180). Amsterdam: North-Holland.
- Nakisa, Ramin Charles, & Ulrike Hahn (1996). Where defaults don't help: the case of the german plural system. In G. W. Cottrell (Ed.), *Proceedings of the 18th annual conference of the cognitive science society* (pp. 177–182).
- Nakisa, Ramin Charles, Kim Plunkett, & Ulrike Hahn (2000). Single- and dual-route models of inflectional morphology. In P. Broeder & J. Murre (Eds.), *Models of language acquisition: Inductive and deductive approaches* (pp. 201–222). Cambridge University Press.
- Quinlan, J. Ross (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.
- Riesbeck, Christopher K., & Roger S. Schank (1989). *Inside case-based reasoning*. Northvale, NJ: Erlbaum.
- Scha, Remko, Rens Bod, & Khalil Sima'an (1999). A memory-based model of syntactic analysis: data-oriented parsing. *Journal of Experimental and Theoretical Artificial Intelligence*, 11, 409–440.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Stanfill, Craig, & David L. Waltz (1986). Toward memory-based reasoning. *Communications of the ACM*, 29, 1213–1228.
- Wulf, Douglas (1996). An analogical approach to plural formation in German. In *Proceedings of the twelfth Northwest Linguistics Conference. Working papers in linguistics*, Volume 14 (pp. 239–254).



## Analogical hierarchy

### Exemplar-based modeling of linkers in Dutch noun-noun compounds\*

Andrea Krott, Robert Schreuder, and R. Harald Baayen

#### Introduction

Traditionally, formal rewrite rules are understood as the normal way to create novel words, while analogy is taken as an unformalizable and exceptional way to create a new word on the basis of an existing word (see e.g., Anshen & Aronoff 1988). The rule-based approach appears to be adequate for phenomena with strong systematicities which can be easily captured by deterministic rules. However, the very same phenomena can often be described equally well by means of formal and computational models of analogy. In the analogical approach, all novel words are modeled on one or more similar existing forms which serve as the analogical set. Especially in the case of gradual phenomena, where rules often capture only the more or less deterministic sub-patterns in the data, the rule-based approach becomes unsatisfactory. It is these phenomena above all which form a testing ground for the two kinds of approaches.

One of these gradual phenomena is the use of linkers in Dutch noun-noun compounds. There are two main linkers, *-en*<sup>-1</sup> and *-s* (e.g., *boek+en+kast*, book+linker+shelf, ‘book shelf’; *dame+s+fiets*, woman+linker+bike, ‘woman’s bike’), which are historically case endings. Synchronically, they are still homographic with the two nominal plural suffixes. Nevertheless, there are two reasons why it is inaccurate to describe them as plural markers. First, the linking *-s* occurs in compounds in which it does not form a plural with the first constituent (e.g., *schaap+s+kooi*, sheep+linker+stable ‘sheepfold’; the plural of *schaap* is *schaap+en*). Second, the linking *-en*, though being always the appropriate plural suffix of the first constituent, does not always contribute plural meaning (e.g., *pan+en+koek*, pan+linker+cake ‘pancake’).

The majority of noun-noun compounds in Dutch do not contain any linker (e.g., *tand+arts*, tooth+doctor ‘dentist’). Such compounds resemble English compounds. Nevertheless, linkers appear in 35% of all Dutch compounds in the CELEX lexical database (Baayen, Piepenbrock, & Gullikers 1995), and their distribution is difficult to predict. On the one hand, there are some deterministic patterns. For instance, *bevolking*, when it is used as a first constituent in a compound, always occurs with the linking *-s*. CELEX lists 30 compounds with *bevolking* as left constituent, all of which are followed by the linker *-s* (e.g., *bevolking+s+aantal*, population+linker+number ‘number of population’). On the other hand, there is rampant unpredictable variation. The left constituent *getal* ‘number’ occurs in CELEX equally often with *-s* (3 times), *-en-* (4 times), and *-Ø-* (3 times). An examination of CELEX shows that 89.6% of all first constituents are variable in terms of their combination with linkers. These variable first constituents account for 25% of all CELEX compounds.

Rule-based approaches to the description of the distribution of Dutch linkers (see e.g., Van den Toorn 1981a, 1981b, 1982a, 1982b; Mattens 1984; ANS 1997) list phonological, morphological, and semantic factors. An example of a phonological rule is the claim that first constituents ending in a full vowel are never followed by any linker (e.g., Van den Toorn 1982a, 1982b; ANS 1997). This rule is not without exceptions, as the example *pygmee+en+volk*, pygmy+linker+people ‘pygmy people’ shows. Morphologically, constraints on linkers are based on preferences of suffixes that appear at the end of first constituents. For instance, the diminutive suffix *-tje* always appears with the linking *-s* (e.g., *kapper+tje+s+saus*, caper+diminutive suffix+linker+sauce, ‘caper sauce’). In contrast, the suffix *-heid* (similar to English ‘-ness’) usually occurs with *-s-*, but also with *-Ø-* and *-en-*. One of the semantic rules claims that linkers never follow mass nouns (e.g., *papier+handel*, paper+trade ‘paper trade’). This is not true for *tabak* ‘tobacco’, which always appears with *-s-* (e.g., *tabak+s+rook*, tobacco+linker+smoke, ‘tobacco smoke’).

There are also attempts to explain linkers by the syntactic relation between the two constituents. If the first constituent is the logical object of the second constituent, a linking element seems to be absent (counterexample: *weer+s+verwachting*, weather+linker+forecast, ‘weather forecast’). Given the large number of exceptions, Van den Toorn prefers the use of the term ‘tendencies’ rather than ‘rules’. Combining all phonological and morphological rules described in the literature,<sup>2</sup> and applying them to the compounds in the CELEX database, we find that they only apply to 51% of all the noun-noun compounds. Moreover, they correctly predict only 63% of the linkers in these compounds, which amounts to only 32% of all CELEX compounds. Thus, rules do not sufficiently describe the distribution of Dutch linkers.<sup>3</sup>

In an earlier study, we show that linkers can be predicted with a high degree of accuracy on the basis of analogy (Krott, Baayen, & Schreuder 2001). This study

revealed strong evidence that the choice of linkers in novel compounds is determined by the distribution of linkers in an analogical set consisting of compounds sharing the first or second constituent with the novel compound. We will refer to this set as the constituent family. We also demonstrated that in the case of compounds with suffixed pseudo-words as first constituents, the analogical set contains all compounds which share the same final suffix of the first constituent. We will refer to this set as the suffix family. In addition to this experimental evidence, the study also showed that the exemplar-based model TiMBL (Daelemans, Zavrel, Van der Sloot, & Van den Bosch 2000) can predict the choices of the participants in off-line production experiments with a high degree of accuracy.

The first goal of the present study is to come to grips with the problem of feature selection. The experiments reported by Krott, Baayen, and Schreuder (2001) suggest that different analogical sets are used depending on the input. In the case of novel compounds with existing first constituents, the selection is based on the constituent family. In the case of novel compounds with suffixed pseudo-words as first constituents, the suffix family is relevant. What happens if the first constituent is a pseudo-word which does not contain a suffix? Possibly, the analogical set for monomorphemic pseudo-words is based on the rime of the pseudo-word. We will refer to this analogical set as the rime family and we will test its influence in Experiment 1.

If constituents, suffixes, and rimes of first constituents individually influence the choice for linkers, the question arises whether these three factors are equally important. TiMBL provides for each feature (used for the analogical prediction) an information gain measure (IG) which quantifies how much information the feature contributes to the knowledge of the correct linker. When taking all compounds with derived nouns as first constituents and comparing the features Constituent and Suffix in terms of their information gain, it turns out that the feature Constituent has the highest IG value (1.1), while the feature Suffix has a value of 0.8. The feature with the next highest information gain (0.75) is the Rime of the first constituent. The order of IG values suggests a hierarchy in which the Constituent is a stronger factor than the Suffix, while the Suffix is a stronger factor than the Rime.

The second goal of this study is to empirically verify this Constituent-Suffix-Rime hierarchy. This hierarchy implies that lower-ranked features are effective only when higher-ranked features are absent. We present results of experiments which test the precedence of the constituent over the suffix (Experiment 2) and the rime (Experiment 3), as well as the precedence of the suffix over the rime (Experiment 4).

The third goal of this study is to compare the two state-of-the-art exemplar-based analogical models, AM (Skousen 1989) and TiMBL (Daelemans, Zavrel, Van der Sloot, & Van den Bosch 2000) with respect to classification accuracy and

prediction uncertainty. We will do this by testing how well these models predict the Dutch compounds in the CELEX lexicon as well as the responses of the participants to Dutch novel compounds in our experiments. We will also compare the uncertainty of participants with the uncertainty of the models.

In what follows, we first describe simulation studies which model the linkers of existing Dutch compounds using AM and TiMBL. These simulation studies show that the feature ‘constituent’ is the best predictor of linkers, although the features ‘suffix’ and ‘rime’ are both strong predictors as well.

In the subsequent section, we present results of simulation studies in which the prediction accuracies of both models are tested for novel compounds. We refer to results of previous experiments which test the influence of the first constituent and the suffix of the first constituent on the choice of the linker. We continue with presenting Experiments 1–4 and the corresponding simulation studies with AM and TiMBL.

### Predicting existing compounds

In this section, we test how well AM and TiMBL predict the linkers in existing Dutch noun-noun compounds attested in the CELEX lexical database. For these studies, compounds with a token frequency of zero in a corpus of 42 million words are not included. Ten-fold cross-validation simulation runs over the remaining 22,966 compounds using different analogical sets led to the results summarized in Table 1. The column Feature lists the different sets of features determining the analogical sets. The columns TiMBL and AM list the classification accuracies for these sets. The rows Constituent, Suffix, and Rime list the percentage of correctly classified CELEX compounds if the model’s training and classification is based on the analogical set of the first constituent, the suffix, and the rime of the first constituent respectively. The constituent family provides the strongest analogical set which correctly classifies about 92% of the compounds in CELEX.<sup>4</sup> This is an extremely high percentage compared to the 32% that are correctly classified by the phonological and morphological rules reported in the linguistic literature. Apparently, the rule-based approach lacks an extremely important factor. However, when AM and TiMBL have to classify the compounds on the basis of the suffix or on the basis of the rime of the first constituent, they already reach an accuracy of 74.6–78.2%, which suggests that phonological and morphological factors are strong predictors as well. If the simulation is restricted to compounds that indeed contain a final suffix, then a classification on the feature Suffix leads to an accuracy as high as 92.3%. Clearly, among the compounds ending in suffixes, the suffix family is an extremely strong predictor. Combining features for the analogical basis generally leads to bet-

**Table 1.** Classification accuracies when training is based on the features Constituent, Suffix, and Rime for both TiMBL and AM

Feature	Accuracy (%)		
	TiMBL	AM	
Constituent	92.6	92.2	
Suffix	74.6 (92.1)*	74.6 (91.3)*	
Rime	78.2	75.6	+
Rime + Suffix	79.5	76.7	+
Rime + Suffix + Constituent	93.4	92.8	+

Note. \* marks the classification accuracy when the training is based only on the 3836 first constituents actually ending in a suffix. + marks the significance of the differences in classification accuracies between TiMBL and AM, evaluated by means of a  $\chi^2$  test.

ter results than a classification which is based on only one feature. The row labeled Rime + Suffix lists the results if the models are trained on the rime and the suffix of the first constituent simultaneously. In this case, AM and TiMBL correctly classifies up to 79.5% of all CELEX compounds. The row labeled Rime + Suffix + Constituent shows the results if all three features are combined. This combination leads to the highest classification accuracies of 93.4% (TiMBL) and 92.8% (AM), which are significantly higher than the accuracies reached by training on only the constituent (TiMBL:  $\chi^2_{(1)} = 11.08$ ,  $p < .001$ ; AM:  $\chi^2_{(1)} = 5.80$ ,  $p = .016$ ).

Comparing the classification accuracies of TiMBL and AM, we find that the models perform equally well as long as the classification is based on the first constituent or the suffix of the first constituent (Constituent:  $\chi^2_{(1)} = 2.57$ ,  $p = .11$ ; Suffix, trained on first constituents ending in a suffix:  $\chi^2_{(1)} = 9.58$ ,  $p = .21$ ). Training on the rime family, however, leads to a significant higher accuracy for TiMBL than for AM ( $\chi^2_{(1)} = 43.53$ ,  $p < .001$ ). This is also true for simulations in which the feature Rime is combined with other features (Rime + Suffix:  $\chi^2_{(1)} = 52.46$ ,  $p < .001$ ; Rime + Suffix + Constituent:  $\chi^2_{(1)} = 6.36$ ,  $p = .01$ ).

Summing up, classifying existing Dutch compounds on the basis of the analogical sets of the first constituent, the suffix or the rime of the first constituent, leads to surprisingly high percentages of correct classifications. However, the features are quite different in strength. The first constituent seems to be the strongest predictor, followed by the rime and the suffix. The best result has been obtained with the combination of all three features. A comparison of AM and TiMBL revealed that the models perform equally well as long as the classification is not based on the rime of the first constituent.



## Predicting novel compounds – influences of individual features

In this section, we test how well AM and TiMBL can predict linking elements that were chosen by participants for novel compounds. We summarize two previous studies in which we observed the influence of the constituent family and the suffix family (Krott, Baayen, & Schreuder 2001). We also present a new experiment which provides evidence for the influence of the rime family. Simulation studies with AM and TiMBL reveal that these analogical models accurately predict the choices of the linkers made by the participants. Both models reveal about the same level of prediction accuracy.

### Constituent and suffix influence

Krott, Baayen, and Schreuder (2001) tested the influence of the distribution of linkers in the constituent family in two experiments in which participants had to form novel compounds from two visually presented nouns. The first experiment focused on the use of the linking *-en-* (EN-experiment), the second on the use of the linking *-s-* (S-experiment). Both experiments tested the influence of the left and right constituent family. The left constituent family was defined as the set of compounds which share the left constituent with the novel target compound, and the right constituent family was defined as the set of compounds which share the right constituent with the target compound. Constituents for the target compounds were chosen such that the distribution of linkers in the left as well as in the right constituent families varied in their bias for the linker *-en-* (EN-experiment) and *-s-* (S-experiment). The bias was defined as the percentage of compounds in the constituent family which contain *-en-* (or *-s-*). The responses of the participants in both experiments showed a strong effect of the bias of the left constituent family and a weaker, but still reliable effect of the bias of the right constituent family. The strength of the bias for a linker was positively correlated with the number of responses with this linker.

Krott, Baayen, and Schreuder (2001) also present simulation studies in which the responses of the participants were modeled with using TiMBL as analogical model. Because of the variation of the responses for each experimental compound, the prediction of TiMBL was compared with the majority choice of the participants for each compound. Using the constituent family of the first constituent, TiMBL correctly predicted 75.1% of all compounds of the EN-experiment and 82.4% of all compounds of the S-experiment. Modeling the responses with AM leads to results which do not differ significantly from the results obtained with TiMBL (EN-experiment: 82.5%,  $\chi^2_{(1)} = 2.68$ ,  $p = .10$ ; S-experiment: 82.0%,  $\chi^2_{(1)} = 0$ ,  $p = 1$ ). The results of both models do not change if the analogical set is based on the Con-

stituent, the Suffix, and the Rime. Thus, the constituent family seems to provide the main analogical basis.

Krott, Baayen, and Schreuder (2001) also investigate whether the suffix of the first constituent influences the choice of the linker, in an experiment in which all first constituents were pseudo-words ending in suffixes. The families of these suffixes differed in their bias for the linking *-s-*. Participants appear to be sensitive to this bias and used the linking *-s-* significantly more often in the case of a strong bias for *-s-* than in the case of a strong bias against *-s-*.

The choices of linkers for the experimental compounds can again be simulated by AM and TiMBL. If we base the classification on the suffix family, the models correctly predict 70.6% of the majority choices of all compounds of the experiment. This does not change if the rime is included in the feature set.

We have seen that the first constituent and the suffix of the first constituent both affect the choice of linkers in novel compounds. AM and TiMBL support these results in predicting the choices of the participants with a high degree of accuracy, using the analogical sets of the constituent family and the suffix family. The prediction accuracies of both models do not differ significantly.

### Experiment 1: Rime influence

In this section, we focus on the question whether the choices for linkers in novel Dutch compounds also depend on another feature with a high information gain, the rime of the first constituent. If the first constituent is a pseudo-word and does not contain any suffix, we assume that participants use the rime family to choose the linker. In addition to the experiment, we will test whether AM and TiMBL are again capable of simulating the experimental results.

#### *Method*

*Materials.* We constructed three sets of 24 phonotactically acceptable Dutch pseudo-words (L1, L2, L3) to be used as left constituents. L1 consisted of pseudo-words with rimes which occur in CELEX most often with a linker. Of these pseudo-words, 12 ended in *-an* (there are 117 compounds in CELEX ending in *-an*, 65.0% of which have a linker) and 12 ended in *-eid* (254 compounds, 99.6% with linker). Conversely, L3 consisted of pseudo-words ending in rimes which show a bias against being combined with a linker. Of these pseudo-words, 6 ended in *-el* (553 compounds, 86.3% without linker), 6 in *-em* (36 compounds, 97.2% without linker), 6 in *-ij* (158 compounds, 89.9% without linker), and 6 in *-a* (237 compounds, 100% without linker). The neutral set L2 consisted of pseudo-words with rimes showing neither a bias for or against a combination with a linker. Of these pseudo-words, 8 ended in *-en* (613 compounds, 52.0% with, 48.0% without

linker), 8 in *-oe* (25 compounds, 44.0% with, 56.0% without linker), and 8 in *-ap* (28 compounds, 25.0% with, 75.0% without linker). Each pseudo-word was bisyllabic. Word stress was indicated on the first syllable by using capital letters. To exclude a possible influence of an existing word, we made sure that none of the pseudo-words ended in an existing Dutch word.

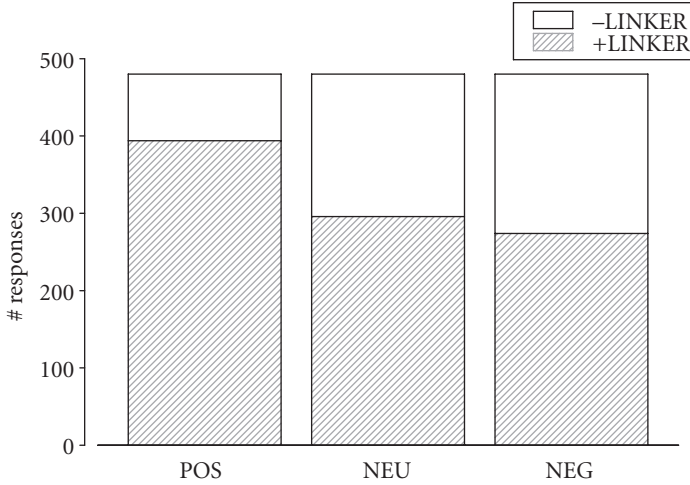
We combined each pseudo-word with an existing right constituent which can appear with all three linking possibilities (*-s-*, *-en-*, and *-Ø-*). This resulted in a factorial design with one factor with three levels: Rime Bias of the first constituent (Positive, Neutral, and Negative). Appendix A lists all  $3 \times 24 = 72$  experimental compounds. We constructed a separate randomized list for each participant.

*Procedure.* The participants performed a cloze-task. The experimental list of items was presented to the participants in written form. Each line presented a pair of compound constituents separated by two underscores. We asked the participants to combine these constituents into new compounds and to specify the most appropriate linker, if any, at the position of the underscores, using their first intuitions. We told the participants that they were free to use *-en-* or *-e-* as spelling variants of the linker *-en-*. The experiment lasted approximately 10 minutes.

*Participants.* Twenty participants, mostly undergraduates at Nijmegen University, were paid to participate in the experiment. All were native speakers of Dutch.

### *Results and discussion*

For one compound, one participant filled in a question mark. This response was counted as an error. Figure 1 displays the number of responses of linkers (+LINKER) and of no linkers (–LINKER) for the three experimental conditions: Positive (POS), Neutral (NEU), and Negative (NEG) Rime Bias. The number of responses are also listed in Appendix A. As can be seen from this figure, a Positive Rime Bias for using a linker leads to more responses with a linker than a Neutral or Negative Bias. A by-item logit analysis (see e.g., Rietveld & Van Hout 1993; Fienberg 1980) of the responses with a linker versus responses without a linker revealed a main effect of the Rime Bias of the first constituent ( $F(2, 69) = 22.2$ ,  $p < .001$ ). We can therefore conclude that the rime of the first constituent affects the choice of the linker. Participants responded to a Negative Bias surprisingly often with a linker. The Negative Rime Bias seems to be less effective. This is remarkable, since the rimes in this condition have been reported as imposing strong restrictions against the usage of linkers in Dutch in the linguistic literature (see e.g., Van den Toorn 1982a, 1982b; ANS 1997). As we will see later, a bias against using a linker seems to be easy to violate in general.



**Figure 1.** Number of responses with linkers (+LINKER) and without linkers (-LINKER) for the Positive, Neutral, and Negative Rime Bias (POS, NEU, NEG)

In contrast to the experiments which tested the effect of the constituent and suffix family, participants found this experiment extremely difficult to perform. This suggests that the phonological rules listed in the literature are not as strong as assumed and may in fact have no reality for at least some of our participants.<sup>5</sup> The difficulties with this experiment cannot be due to a weaker strength of the bias because in all experiments the bias in the positive and negative condition was equally strong (EN-experiment: Mean Positive Bias 91%, Mean Negative Bias 100%; S-experiment: Mean Positive Bias 98.7%, Mean Negative Bias 100%; Suffix Experiment: Mean Positive Bias 91.9%, Mean Negative Bias 83.3%; Rime Experiment: Mean Positive Bias 82.3%, Mean Negative Bias 93.3%).

Given the difficulties experienced by the participants to complete the task, the uncertainty in their choices (with marginal majority choices) does not come as a surprise. Interestingly, AM's and TiMBL's performance with respect to the effect of the Rime Bias reveals a high degree of uncertainty as well. Both models correctly predict about half of the majority choices if they are trained on the rime of the first constituents of the 22,966 CELEX compounds (TiMBL: 47.9%; AM: 47.2%;  $\chi^2_{(1)} = 0$ ,  $p = 1$ ). However, prediction accuracies increase (TiMBL: 64.8%; AM: 65.3%;  $\chi^2_{(1)} = 0$ ,  $p = 1$ ) if the training is based not only on the rime but also on the stress of the last syllable of the first constituent.

If the feature set contains Rime, Stress, and Suffix of the first constituent, TiMBL's accuracy drops to 53.4%, while AM's accuracy stays the same with 65.3% ( $\chi^2_{(1)} = 1.82$ ,  $p = .18$ ). The lower accuracy of TiMBL is due to its analogical mecha-

nism which can lead to level interference of factors. When training is conducted on Rime and Suffix simultaneously, derived and monomorphemic words build separate analogical sub-bases. Consequently, generalizations based on rimes can no longer take priority for the whole dataset.<sup>6</sup>

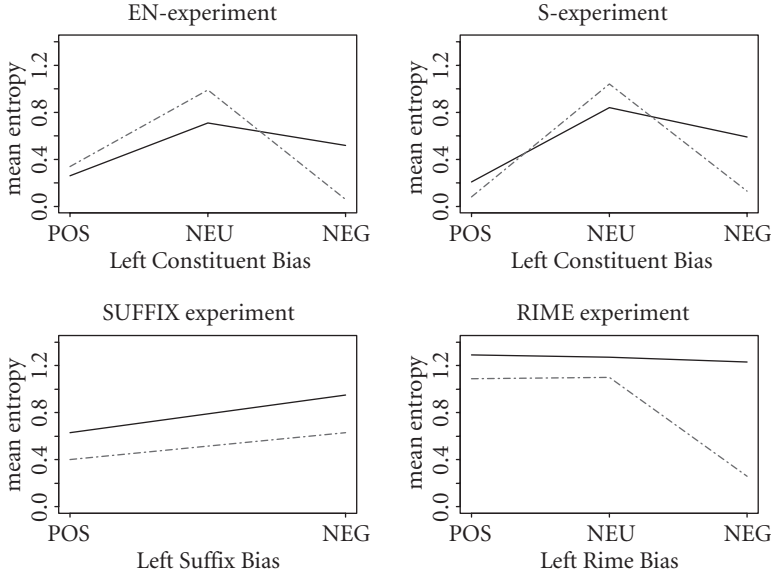
### The uncertainty of choosing linkers

In all AM and TiMBL simulation studies presented in this paper, we investigate how well these models predict the linkers in novel compounds, comparing the linker to which the models assign the highest probability value with the linker which has been chosen most often by the participants. That means that both the less probable linkers for the models and the linkers which are chosen less often by the participants are not taken into account when evaluating the models' performance. In this section, we will focus on the uncertainty in choosing a linker both on the side of the models and on the side of the participants, addressing the question whether the participants and the models are unsure or sure about the linkers for the same kinds of compounds.

We measured the uncertainty of a model for a linker in a particular compound in terms of the entropy of the distribution of the probabilities the model assigns to the linkers *-en-*, *-s-*, and *-Ø-* for this compound. The entropy value is the higher the more equally distributed the linkers are. Similarly, we measured the uncertainty of the participants in terms of the entropy of the distribution of the probability values of their choices for a given compound.

Figure 2 shows the entropy for different Left Biases in the experiments testing the influence of the Constituent Bias, the Suffix Bias, and the Rime Bias. The upper left panel shows the mean entropy for the three Left Bias conditions in the EN-experiment (Positive, Neutral, and Negative Constituent Bias). The solid line represents the mean entropy of the distribution of the participants' responses over all experimental items in the three conditions of Left Bias. As can be seen from the slope of the line, the entropy, and therefore the uncertainty, is highest in the case of a Neutral Left Constituent Bias. This is also true for the entropy of the distributions of the predictions given by the models. A Spearman correlation test revealed a significant correlation between the entropy of the participants' responses and the entropy of the models' predictions ( $r_s = 0.30$ ;  $z = 4.14$ ;  $p < .001$ ). Interestingly, for this and the following experiments, AM and TiMBL reveal exactly the same average entropy per bias condition.

The upper right panel of Figure 2 shows the mean entropy for the three Left Bias conditions in the S-experiment. Here again, both the models and the participants are most uncertain in the condition of a Neutral Constituent Bias, and



**Figure 2.** Mean entropy for the distributions of choices for linkers for both the models (superimposed dashed lines) and the participants (solid lines) for the experiments testing the influence of the Left Constituent Bias (EN-experiment and S-experiment), the Suffix Bias (SUFFIX experiment), and the Rime Bias (RIME experiment)

the entropy values of the models' predictions and the participants' responses are significantly correlated ( $r_s = 0.48$ ;  $z = 6.79$ ;  $p < .001$ ).

Surprisingly, in both the EN-experiment and S-experiment, the models are much more certain in their predictions than the participants for the condition in which the constituent family of the left constituent has a Negative Bias (EN-experiment:  $t(124) = 8.68$ ;  $p < .001$ ; S-experiment:  $t(124) = 7.19$ ;  $p < .001$ ). There are two explanations for this result. First, in the EN-experiment, participants responded most often with *-en-* (2254 out of 3778;  $\chi^2_{(1)} = 281.33$ ,  $p < .001$ ) and in the S-experiment, they responded most often with *-s-* (2092 out of 3780;  $\chi^2_{(1)} = 85.93$ ,  $p < .001$ ). Thus, there might be an overall bias for using *-en-* or *-s-*. Second, in the condition of a Left Negative Bias, either 50% (EN-experiment) or 90% (S-experiment) of the left constituents have a bias for  $-\emptyset$ . Post-hoc analyses revealed that a bias against using a linker can be violated more easily than a bias for *-en-* or *-s-*. In the EN-experiment, 75% of the responses followed the bias if it was for  $-\emptyset$ , while 93.2% followed the bias if it was for *-en-* or *-s-* ( $\chi^2_{(1)} = 11.06$ ,  $p < .001$ ). In the S-experiment, 82.4% of the responses followed the bias if it was for  $-\emptyset$ , while 93.5% followed the bias if it was for *-en-* or *-s-* ( $\chi^2_{(1)} = 4.78$ ,  $p = .003$ ). These results suggest that the  $-\emptyset$ - linker might not have

the status of a morpheme. A bias for  $-\emptyset$ - would then not be positive evidence for a zero-morpheme, but rather negative evidence against using a linker. Such negative evidence might be weaker as an analogical factor than positive evidence for  $-en$ - or  $-s$ -. Participants would then follow the negative bias less often, leading to greater uncertainty about the choice of the appropriate linker.

The lower left panel of Figure 2 shows the mean entropy for the two Suffix Biases (Positive and Negative) in the experiment testing the influence of the Suffix Bias. The models are in general less uncertain about the choices than the participants ( $t(124) = -5.29$ ;  $p < .001$ ). Possibly, using the analogical set of the suffix family is already more difficult than using the constituent family. There is again a significant correlation between the entropy values of the participants' choices and the models' predictions ( $r_s = 0.30$ ;  $z = 3.37$ ;  $p < .001$ ).

As mentioned above, participants found the experiment in which we tested the influence of the Rime Bias extremely difficult to perform. Not surprisingly, the entropy values of the participants' responses shown in the lower right panel of Figure 2 are very high. Interestingly, the entropy does not differ across the three different conditions (Positive versus Neutral Bias:  $t(46) = 0.66$ ;  $p = 0.52$ ; Positive versus Negative Bias:  $t(46) = 0.95$ ;  $p = 0.35$ ). There is also no correlation between the entropy values of the participants' responses and the models' predictions ( $r_s = -.10$ ;  $z = -.80$ ;  $p = 0.42$ ). Interestingly, the models are as uncertain in the condition of a Positive Bias as in the condition of a Neutral Bias ( $t(46) = -.12$ ;  $p = 0.90$ ). This uncertainty is probably due to the quite low bias (65%) for half of the compounds in this condition. However, most of the responses do follow the bias (82%). The high degree of uncertainty in the condition of a Negative Bias can be again explained by the general observation that a bias for  $-\emptyset$ - can be easily violated.

In sum, we have seen that participants and models tend to be uncertain especially in the condition of a neutral bias. In all experiments, a negative bias reveals higher uncertainty on the side of the participants than on the side of the models. We explained this result by the observation that a bias against using a linker seems to be more easily violated. This finding suggests that an analogical model for predicting human performance needs to weight zero-realizations differently than other realizations.

## The feature hierarchy

The experiments testing the influence of the first constituent, of the suffix, and of the rime have revealed that all three features are effective factors. This does not mean, however, that these factors are equally effective under the same conditions.

Participants may activate the constituent family when it is available. If the first constituent does not have a constituent family, participants base their choice on either the suffix family or on the rime family of the first constituent. In the case of a derived first constituent, the suffix is involved, while in the case of a pseudo-word without any suffix, the rime is crucial. We may be dealing with a fall-back strategy. In the absence of a higher-order unit, the next lower unit determines the analogical set. However, what happens if the information given in the input allows the selection of more than one feature? Are all such features activated simultaneously and do they equally affect the choice of the linker? The different information gains which are provided by TiMBL suggest the hypothesis that the features are ordered in strength. The influence of the constituent might be stronger than that of the suffix, while in turn the influence of the suffix might be stronger than that of the rime. We will test these hypotheses in the following three experiments (Experiment 2–4), and we will use AM and TiMBL to investigate the possible role of different analogical sets.

### Experiments 2 and 3: Constituent preference

Experiments 2 and 3 test whether the first constituent has a stronger influence on the choice of linkers than the suffix (Experiment 2) or the rime (Experiment 3) of the first constituent.

#### Experiment 2: Constituent preference or suffix preference

##### *Method*

*Materials.* For this experiment, we selected a set of 14 derived nouns whose suffixes are mostly combined with the linking *-s-* (mean: 86.8%; *-ing-*: 91.4%; *-ling-*: 80.9%; *-eling-*: 86.7%; *-er-*: 84.1%). At the same time, these derived nouns, when used as left constituents in compounds, tend to occur without the linker *-s-* (mean: 91.7%; range: 75.0% – 100%; 10 had a bias for  $-\emptyset-$  and 4 had a bias for *-en-*). To make sure that the bias for *-en-*, *-s-*, and  $-\emptyset-$  was equal over the list of experimental items, we added 10 monomorphemic nouns with a bias for *-s-* (mean: 98.1%; range: 83.3% – 100%) and 6 monomorphemic nouns with a bias for *-en-* (mean: 91.1%; range: 66.7% – 100%), resulting in 30 left constituents. The 10 monomorphemic nouns with a bias for *-s-* served as experimental items for Experiment 3.

In order to avoid an influence of the right constituent, we combined these 30 left constituents with right constituents that appear with all three linking possibilities (*-s-*, *-en-*, and  $-\emptyset-$ ). Appendix B lists the 16 experimental items. We constructed a separate randomized list for each participant.



*Procedure.* The procedure was identical to the one used in Experiment 1.

*Participants.* Twenty participants, mostly undergraduates at Nijmegen University, were paid to participate in the experiment. All were native speakers of Dutch.

### *Results and discussion*

None of the participants' responses had to be counted as an error. The left bar of Figure 3 shows the number of responses that follow the bias of the constituent, the right bar shows the number of responses that follow the bias of the suffix. The number of responses for the individual compounds are listed in Appendix B. Participants responded most often with the linker that one would expect if they follow the bias of the constituent. Only in 28.6% of all responses was the linker in line with the bias of the suffix. A paired t-test revealed that Constituent Bias reliably overrides Suffix Bias ( $t(13) = 3.04$ ;  $p < .01$ ). We conclude that the first constituent has indeed a stronger effect on the choice of the linker than the suffix of the constituent.

Simulation studies with TiMBL and AM confirm this result. When we train TiMBL and AM on the first constituents of the 22,966 CELEX compounds, they both correctly predict 64.3% of the majority choices for each experimental compound. If the training is based on the suffix, they correctly predict only 21.4%. Training on the rime, the suffix, and the first constituent simultaneously leads to the same results as training on only the first constituent. Therefore, it seems to be mainly the first constituent and its constituent family which is actively used by the participants.

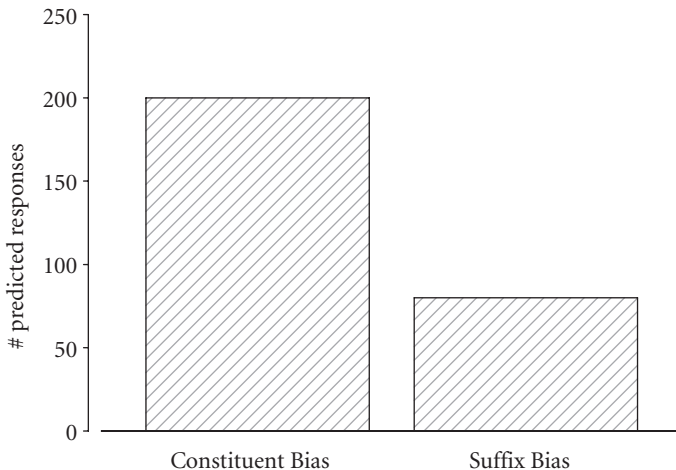


Figure 3. Number of responses predicted by the Constituent Bias and Suffix Bias

In the next section we address the question whether the bias of the constituent family also overrules the bias of the rime family.

### Experiment 3: Constituent preference or rime preference

#### *Method*

*Materials.* We selected from CELEX a set of 10 monomorphemic nouns which tend to occur with a linker (mean: 84.4%; range: 66.7%–100%). At the same time, the rimes of these nouns tend to occur without a linker (mean: 90.6%; *-ee*: 97.1%; *shwa + l*: 87.9%; *-ij*: 90.6%). Six of these nouns had a bias for a combination with the linker *-en-* and four had a bias for *-s-*. To make sure that the bias for *-en-*, *-s-*, and *-Ø-* was equal over the list of experimental items, we added ten derived nouns with a bias against using a linker (mean: 93.9%; range: 63.6%–100%), four derived nouns with a 100% bias for *-en-*, and six monomorphemic nouns with a 100% bias for *-s-*, resulting in 30 left constituents.

In order to avoid an influence of the right constituent, we combined these 30 left constituents with right constituents which appear with all three linkers (*-s-*, *-en-*, and *-Ø-*). Appendix B lists the 10 experimental compounds. We constructed a separate randomized list for each participant.

*Procedure.* The procedure was identical to the one used in Experiments 1 and 2.

*Participants.* Twenty participants, mostly undergraduates at Nijmegen University, were paid to participate in the experiment. All were native speakers of Dutch.

#### *Results and discussion*

None of the participants' responses had to be counted as an error. The left bar of Figure 4 shows the number of responses that follow the bias of the constituent, the right bar shows the number of responses following the bias of the rime. The number of responses of the individual compounds are listed in Appendix B. Figure 4 shows that participants responded mostly with the linker following the bias of the constituent. Only in 11.5% of all responses was the linker in line with the prediction based on the Rime Bias. A paired t-test by items on the number of participants following the bias of the constituent and the number of participants following the bias of the rime confirms that the observed pattern is reliable ( $t(9) = 8.6, p < .001$ ). We can therefore conclude that the influence of the first constituent has a stronger effect on the choice of the linker than the rime of the constituent.

When we train TiMBL and AM on the first constituents of the 22,966 CELEX compounds, both correctly predict 10 out of 10 of the majority choices for each experimental compound. However, when we train on the rime, they correctly pre-

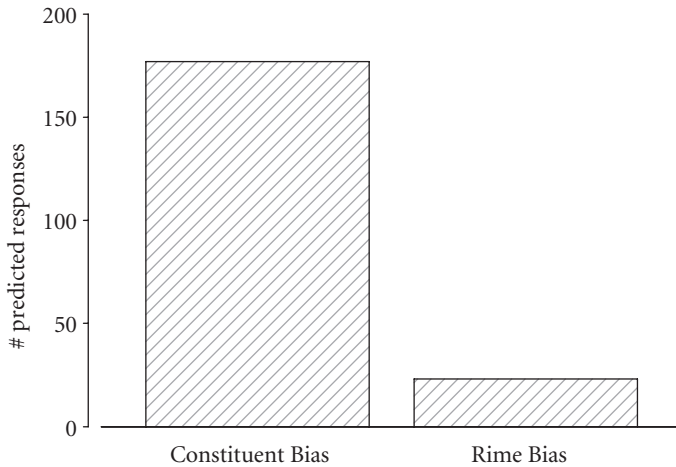


Figure 4. Number of responses predicted by the Constituent Bias and Rime Bias

dict 0 out of 10. Training TiMBL on the rime, the suffix, and the first constituent simultaneously leads to the same results as training on only the first constituent, namely 100% correct predictions. AM's prediction accuracy in this case drops to 90%, which is not significantly lower ( $\chi^2_{(1)} = 0.002$ ,  $p = .96$ ). Clearly, participants base their choices on the constituent family and not on the rime family. In the next section, we will test whether the Suffix Bias is stronger than the Rime Bias.

### Suffix preference

#### Method

*Materials.* We constructed a list of  $4 \times 3 = 12$  phonotactically legal Dutch pseudo-words which ended in 4 different Dutch suffixes that also appear as word-final letter combinations in monomorphemic nouns (*-er*, *-aar*, *-ing*, and *-ist*). When these letter combinations appear in monomorphemic nouns, they are usually not combined with a linker (mean: 72.9%; range: 59.4%–93.5%). In contrast, when they appear as suffixes, they tend to be combined with a linker (mean: 84.0%; *-er*: 84.1% with *-s*; *-aar*: 66.7% with *-s*; *-ing*: 91.4% with *-s*; *-ist*: 93.8% with *-en*).

In order to balance the bias for linkers in the experiment, we also constructed 24 filler constituents. Half of these were phonotactically legal Dutch derived pseudo-words ending in suffixes that appear always without any linker (*-sel*, *-te*, *-atie*, and *-nis*; 3 pseudo-words for each suffix). The other half of the filler items were phonotactically legal Dutch monomorphemic pseudo-words ending in letter combinations that usually appear with a linker (mean: 63.7%; *-eid*: 86.0%, *-ap*: 37.1%, and *-an*: 67.9%; 4 pseudo-words for each combination). For both the

12 experimental items and the 24 fillers, stressed syllables were marked by capital letters.

We constructed two lists of experimental items (List A, List B). Both lists contained the 12 experimental pseudo-words. To List A we added the 12 filler words which usually appear with a linker. To List B we added the 12 fillers which usually appear without a linker. Each pseudo-word was embedded in a sentence constructed to influence the interpretation of the pseudo-word. For the words of List A, the sentences promoted a monomorphemic interpretation of the pseudo-word. For the words of List B, the sentences promoted an affixal interpretation. The following two examples show one of the experimental pseudo-compounds preceded by the two contexts.

#### A. monomorphemic interpretation

*Een 'PLOEver' is een boomsoort. PLOEver\_gried*  
 "A 'PLOEver' is a kind of tree. PLOEver\_gried"

#### B. derived interpretation

*Iemand die graag 'ploeft' is een 'PLOEver'. PLOEver\_gried*  
 "Somebody who likes to 'ploef' is a 'PLOEver'. PLOEver\_gried."

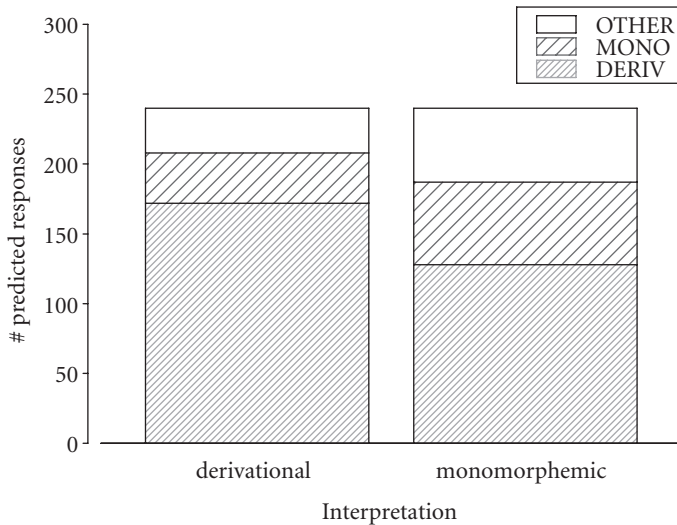
In addition, we constructed  $12 + 24 = 36$  compounds each using a pseudo-word of Lists A and B as a left constituent and combining it with a right phonotactically legal pseudo-word that does not appear in Lists A and B. The compounds with the 12 experimental left constituents were identical in both lists. Appendix A lists all sentences and compounds of both lists. We constructed a separate randomized list for each participant.

*Procedure.* The participants performed a cloze-task. The sentences defining the pseudo-words and the compounds were presented to the participants in written form. Each line presented a sentence and the pair of compound constituents in which the first constituent was identical to the defined pseudo-word. The constituents were separated by two underscores. The participants were instructed to first read the sentence twice in order to understand the meaning of the pseudo-word. Then they had to combine the two constituents into a new compound and to specify the most appropriate linker, if any, at the position of the underscores, using their first intuitions. We told the participants that they were free to use *-en-* or *-e-* as spelling variants of the linker *-en-*. The experiment lasted approximately 5 minutes.

*Participants.* Forty participants, mostly undergraduates at Nijmegen University, were paid to participate in the experiment. All were native speakers of Dutch. List A was presented to one half of the participants, List B to the other half.

### *Results and discussion*

All participants provided a linking choice for all items. The left bar of Figure 5 (derivational interpretation) shows the number of responses when the sentence favors a derivational interpretation. As can be seen from the figure, this condition mainly led to responses as predicted by the bias of the suffix (mean: 71.5%). The right bar of Figure 5 (monomorphemic interpretation) shows the number of responses when the sentence favors a monomorphemic interpretation. The number of responses for the individual compounds are listed in Appendix C. Paired t-tests of the number of responses for the two contexts show that participants responded more often with the predicted linker for a derived first constituent for a sentence favoring a derivational interpretation than for a sentence favoring a monomorphemic interpretation ( $t(11) = 4.5; p < .001$ ). They also responded more often with the predicted linker for a monomorphemic first constituent for a sentence favoring a monomorphemic interpretation than for a sentence favoring a derived interpretation ( $t(11) = 2.9; p = .01$ ). However, even in the case of a sentence



**Figure 5.** Number of responses predicted by the Suffix Bias (DERIV) or the Monomorphemic Bias (MONO), and percentage of other responses (OTHER) in the two experimental conditions of a derivational and monomorphemic interpretation

favoring a monomorphemic interpretation, more responses are predicted by the bias of the suffix than by the bias of the rime ( $t(11) = 3.5$ ;  $p = .004$ ).

These results lead to two conclusions. First, rimes and suffixes of first compound constituents independently influence the choice of linkers. Second, the influence of the suffix is much stronger. It is the prominent factor even when the pseudo-word is introduced contextually as a monomorphemic word.

When we train TiMBL and AM on the suffix of the first constituents of the 22,966 CELEX compounds, they correctly predict 100% of the majority choices for each experimental compound in the case of a preceding sentence favoring a derived interpretation. Their prediction accuracy drops to 83.3% in the case of a preceding sentence favoring a monomorphemic interpretation. When we train the models on the rime instead, they predict only 50% in the case of a sentence favoring a derived interpretation. Their prediction rises to 58.3% in the case of a sentence favoring a monomorphemic interpretation.

These results support the experimental finding that the behavior of the participants is influenced by the context. Participants base their choices more often on the analogical set of the rime instead of the suffix if the preceding sentence favors a monomorphemic interpretation. The results also mirror the stronger influence of the suffix, which seems to easily activate the corresponding suffix family when it is present in the input, even when the monomorphemic interpretation of the pseudo-word should inhibit this activation.

## General discussion

This study aimed for three goals. First, we tried to come to grips with the problem of feature selection in the task of choosing the appropriate linkers in Dutch noun-noun compounds. Second, we tested whether the three main relevant features for this task – Constituent, Suffix, and Rime – are hierarchically ordered. Third, we simulated the choices of participants with AM and TiMBL and compared these models with respect to their classification accuracies and their prediction uncertainty.

The first goal, solving the problem of feature selection, has been addressed by simulation studies focusing on existing compounds in CELEX and experiments with novel compounds. Both kinds of studies have shown that the three analogical sets of the constituent family, the suffix family, and the rime family all influence the choice of linkers in compounds. However, the three factors are not effective to the same extent and under the same conditions. The simulation studies with existing compounds revealed that the constituent family seems to provide the strongest analogical set. The suffix family is as strong as the constituent family, but only for

compounds with first constituents which indeed end in a suffix. Otherwise, it is the least effective one of the three factors. The experiments with novel compounds suggest that the features Constituent, Suffix, and Rime are selected on the basis of a fall-back strategy. If a higher-ranked feature is absent in the input, the next feature down the hierarchy becomes the basis for the analogical choice. This way of feature selection means for AM and TiMBL that we need a component that is dynamically tuned to the information in the input.

At the bottom of the feature hierarchy, the rime family emerges as a rather problematic analogical set. Participants reported extreme difficulties with the experiment testing the influence of the rime. These difficulties were confirmed by the analyses of the uncertainty in the responses of the participants, which revealed a high entropy across all conditions of this experiment. Due to this uncertainty, AM and TiMBL reach a rather low prediction accuracy of maximal 65.3% (AM) and 64.8% (TiMBL), which is less than the accuracies for the experiments testing the influence of the suffix (TiMBL: 92.1%; AM: 75.4%) and constituent (EN-experiment: TiMBL: 75.1%, AM: 82.5%; S-experiment: TiMBL: 82.4%, AM: 82.0%). Apparently, choosing linkers on the basis of the rime of the first constituent is an unusual task. This is not so surprising, considering the fact that for normal compounds there is usually a constituent family or at least a suffix family available which can serve as the analogical set.

The second main question of this study was whether the features Constituent, Suffix, and Rime are hierarchically ordered. We indeed found experimental evidence suggesting that the Constituent Bias overrules both Suffix and Rime Bias. The Suffix Bias in its turn seems to be stronger than the Rime Bias. These results suggest that categories with a lower rank in the hierarchy are only effective in case there is no higher-ranked category available. However, this does not mean that lower-ranked features are not activated. There are two points we have to mention here. First, including a lower-ranked feature into the feature set on which AM and TiMBL was trained in order to simulate participants' choices for linkers never changed the prediction accuracy reliably. Second, recall that in 10-fold cross-validation runs over all CELEX compounds, AM and TiMBL reached the highest classification accuracies when the training was based on all three features simultaneously. The simplest model explaining this finding is an inclusive hierarchy. Whenever there is evidence for a feature in the input, the corresponding analogical set is co-activated. It remains crucial, however, that the highest available feature in the hierarchy is included when training the model.

The third main goal of this study was a comparison of AM and TiMBL with respect to classification accuracy and prediction uncertainty. Comparing the classification and prediction accuracies of AM and TiMBL for existing and novel compounds, we can conclude that, all in all, the models perform equally well. A difference has been found in one case only. Classifying existing compounds taken from

CELEX, including the feature Rime in the feature set, led to significantly lower classification accuracies for AM. In all other cases, the observed differences were not reliable, although we should mention that we found a problem of level-interference with TiMBL. When predicting the linkers chosen by participants in the experiment testing the influence of the Rime Bias, including the feature Suffix reduced the prediction accuracy by approximately 10%.

Analyses of the entropy of the choice-distributions of the participants on the one hand and of the models on the other hand revealed that uncertainty is correlated with the strength of the bias in a family. In the case of a neutral bias, both the models and the participants are significantly more uncertain about the appropriate linker than in the case of a strong bias. The relative high uncertainty of participants in the case of a negative bias can be explained by an overall bias for the specific linker for which an experiment is designed, or by a weaker analogical strength of the bias for  $-\emptyset$ . The mean uncertainty of the two models across items in an experimental condition turned out to be identical in all the investigated experiments. We therefore conclude that the models do not differ in their prediction uncertainty.

In this paper, we have focused on the analogical approach to a partly non-deterministic morphological phenomenon. The standard approach to the analysis of morphological phenomena is to formulate formal rules (e.g., Aronoff 1976; Selkirk 1982; Lieber 1981). In these rule-based approaches, the aim is to capture the generalizations that govern the data. Once a formal rule has been formulated on the basis of inspection of the data, the data themselves become irrelevant, because the rule operates independently of the data to its input. Various researchers (e.g., Clahsen 2000; Marcus, Brinkman, Clahsen, Wiese, & Pinker 1995; Pinker 1991, 1997) argue that these symbolic rules have cognitive reality in the brain.

The standard approach has come under attack from connectionist modelers (e.g., Plunkett & Juola 1999; Seidenberg 1987; Seidenberg & Hoeffner 1998; Rueckl, Mikolinski, Raveh, Miner, & Mars 1997), who exchange symbolic for sub-symbolic representations and merge data instances and rules into the connection weights of multi-layered artificial neural networks (ANN). Probably, ANN models will be able to capture the choice of linkers as well. What our simulation results show, however, is that it is not necessary to give up symbolic representations when the goal is to model non-deterministic data. The analogical approach, moreover, is supported by independent psychological evidence that morphological families play a role in language processing (Schreuder & Baayen 1997; Bertram, Schreuder, & Baayen 2000; De Jong, Schreuder, & Baayen 2000). In addition, a sketch of a psycholinguistic spreading-activation model for the selection of linkers can be found in Krott, Baayen, & Schreuder 2001. We conclude that the analogical approach to morphological rules, in which static symbolic rules abstracted from the data are replaced by dynamic, analogical rules that are linked to and continuously updated by the data, is a fruitful area for future research.



## Notes

\* This study was financially supported by the Dutch National Research Council NWO (PIONIER grant to the third author), the University of Nijmegen (The Netherlands), and the Max Planck Institute for Psycholinguistics (Nijmegen, The Netherlands).

1. The *-en-* has an orthographic variant *-e-* which, in standard Dutch, does not differ in pronunciation.
2. We did not test any semantic rules because CELEX does not provide the required semantic information.
3. For a list of all applicable rules see Appendix D.
4. All results of TiMBL (version 3.0) in this paper are obtained by using the standard IB1 algorithm, the overlap similarity metric with information gain weighting, and one nearest neighbor for extrapolation. In our simulation studies, this set of parameters has been proven to lead to the best results. For AM we excluded ‘=’ as a variable, set the option ‘given’ to ‘exclude’, the option ‘probability’ to unity, and used the option ‘squared’ without specifying any frequency range.
5. Vance (1980) reports similar findings in his study of Lyman’s law which predicts the occurrence of rendaku in Japanese compounds. He concludes that rendaku is psychologically real only for a rather small minority of speakers.
6. Using different parameter settings does not enhance performance. Training on constituent, suffix, and rime while using the IGTREE algorithm leads to 30.1% correctly predicted compounds. With TRIBL we reach 37.0%. If we enhance the number of nearest neighbors for extrapolation to three, both IG and TRIBL reach a prediction accuracy of 30.1%.

## References

- Anshen, Frank, & Mark Aronoff (1988). Producing morphologically complex words. *Linguistics*, 26, 641–655.
- Aronoff, Mark (1976). *Word formation in generative grammar*. Cambridge, MA: MIT Press.
- Baayen, R. Harald, Richard Piepenbrock, & Léon Gulikers (1995). *The CELEX lexical database (CD-ROM)*. University of Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.
- Bertram, Raymond, Robert Schreuder, & R. Harald Baayen (2000). The balance of storage and computation in morphological processing: The role of word formation type, affixal homonymy, and productivity. *Journal of Experimental Psychology: Memory, Learning, and Cognition*, 26, 419–511.
- Clahsen, Harald (1999). Lexical entries and rules of language: a multi-disciplinary study of German inflection. *Behavioral and Brain Sciences*, 22, 991–1060.
- Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, & Antal van den Bosch (2000). TiMBL: Tilburg memory based learner reference guide. Version 3.0. Technical Report ILK 00-01, Computational Linguistics, Tilburg University.
- De Jong, Nivja H., Robert Schreuder, & R. Harald Baayen (2000). The morphological family size effect and morphology. *Language and Cognitive Processes*, 15, 329–365.

- Fienberg, Stephen E. (1980). *The analysis of cross-classified categorical data*. Cambridge, MA: MIT Press.
- Haeseryn, Walter, K. Romijn, Guido Geerts, Jaap de Rooij, & Maarten C. van den Toorn (1997). *Algemene Nederlandse Spraakkunst*. Groningen: Martinus Nijhoff.
- Krott, Andrea, R. Harald Baayen, & Robert Schreuder (2001). Analogy in morphology: modeling the choice of linking morphemes in Dutch. *Linguistics*, 39.
- Lieber, Rochelle (1980). On the organization of the lexicon. Ph.D. thesis. Cambridge, MA: MIT.
- Mattens, W. H. M. (1984). De voorspelbaarheid van tussenklanken in nominale samenstellingen. *De nieuwe taalgids*, 7, 333–343.
- Pinker, Steven (1991). Rules of language. *Science*, 153, 530–535.
- Pinker, Steven (1997). Words and rules in the human brain. *Nature*, 387, 547–548.
- Plunkett, Kim, & Patrick Juola (1999). A connectionist model of English past tense and plural morphology. *Cognitive Science*, 23, 463–490.
- Rietveld, Toni, & Roeland Van Hout (1993). *Statistical techniques for the study of language and language behaviour*. Berlin: Mouton de Gruyter.
- Rueckl, Jay G, Michèle Mikolinski, Michal Raveh, Caroline S. Miner, & Frans Mars (1997). Morphological priming, fragment completion, and connectionist networks. *Journal of Memory and Language*, 36, 382–405.
- Schreuder, Robert, & R. Harald Baayen (1997). How complex simplex words can be. *Journal of Memory and Language*, 37, 118–139.
- Seidenberg, Mark (1987). Sublexical structures in visual word recognition: Access units or orthographic redundancy. In M. Coltheart (Ed.), *Attention and Performance XII* (pp. 245–263). Hove: Lawrence Erlbaum Associates.
- Seidenberg, Mark S., & James H. Hoeffner (1998). Evaluating behavioral and neuroimaging data on past tense processing. *Language*, 74, 104–122.
- Selkirk, Elisabeth O. (1982). *The syntax of words*. Cambridge, MA: MIT Press.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Toorn, Maarten C. van der (1981a). De tussenklank in samenstellingen waarvan het eerste lid een afleiding is. *De nieuwe taalgids*, 74, 197–205.
- Toorn, Maarten C. van der (1981b). De tussenklank in samenstellingen waarvan het eerste lid systematisch uitheems is. *De nieuwe taalgids*, 74, 547–552.
- Toorn, Maarten C. van der (1982a). Tendenzen bij de beregeling van de verbindingsklank in nominale samenstellingen I. *De nieuwe taalgids*, 75, 24–33.
- Toorn, Maarten C. van der (1982b). Tendenzen bij de beregeling van de verbindingsklank in nominale samenstellingen II. *De nieuwe taalgids*, 75, 153–160.
- Vance, Timothy J. (1980). The psychological status of a constraint on Japanese consonant alternations. *Linguistics*, 18, 145–167.

## Appendix A

Materials of Experiment 1: left constituent and right constituent (number of responses with a linker, number of responses without a linker). Capital letters mark word stress.

### L1: Positive Rime Bias

LANTan organisatie (16, 4); VANEid kooi (18, 2); PEUZeid steun (18, 2); KApeid gedrag (17, 3); HORan oord (16, 4); MOEveid voer (18, 2); NOGan plicht (19, 1); GOERan probleem (16, 4); VEEpleid milieu (15, 5); BALan geschiedenis (15, 5); PLAVEid paar (19, 1); LUISan pensioen (18, 2); MIJstan commissie (18, 2); BOELan niveau (15, 5); KOLan controle (12, 8); DAkeid republiek (16, 4); LUCHan conflict (15, 5); BOENEid stam (14, 6); TOpleid gezicht (17, 3); ZAPleid verzameling (16, 4); KEEzeid waarde (17, 3); GROtan aanbod (14, 6); VIJzan dienaar (17, 3); POEkeid hok (18, 2).

### L2: Neutral Rime Bias

Oloe corps (12, 8); MARvoe verzekering (13, 7); TOTroe galerij (8, 12); BODap regeling (16, 4); KIJDap structuur (13, 7); VEUnen pensioen (9, 11); STIEvap karakter (15, 5); DROlen oord (16, 4); TAZoe tak (13, 7); PAGoe toestand (13, 7); BLOstoe hut (8, 12); MIEfap element (14, 6); SCHIJlen middel (10, 10); PLOElen element (10, 10); BIEvap zone (15, 5); BOEdap middel (19, 1); VILnoe vlees (11, 9); POEneen organisatie (6, 14); KRAzen conflict (10, 10); POERgoe vrouw (11, 9); KODap beleid (14, 6); ZOzen zone (9, 11); PUIbap rust (19, 1); DULLen rust (12, 8).

### L2: Negative Rime Bias

NApla bond (7, 13); TUIzem dienaar (15, 5); BIEzel waarde (12, 8); SILda tong (13, 7); KLAVij structuur (9, 11); DRASij regeling (12, 8); WONkel geschiedenis (11, 9); BRANij hulp (13, 7); TIKsem aanbod (14, 6); BISSel probleem (12, 8); PLUIvij karakter (12, 8); PLOdem plicht (16, 4); POEkrij conferentie (10, 10); ARTa vel (11, 9); STIJza kas (10, 10); LIEsem niveau (18, 2); TISSel milieu (6, 14); DUISkra zee (7, 13); STALEm controle (8, 12); DRUImel gedrag (9, 11); SOERkwa kop (10, 10); VOENij beleid (12, 8); VAjel gezicht (15, 5); KROEsem commissie (12, 8).

## Appendix B

Materials of Experiment 2: left constituent and right constituent (number of responses according to the constituent, number of responses according to the suffix).

vluchteling gezicht (20, 0); voorziening regeling (11, 9); belasting kas (13, 7); vreemdeling republiek (20, 0); tiener gedrag (18, 2); tweeling kop (14, 6); kaper hulp (3, 17); woning kooi (17, 3); zuigeling probleem (19, 1); luidspreker hok (7, 13); leerling vel (20, 0); klapper galerij (14, 6); veiling commissie (9, 11); waterleiding aanbod (15, 5).

Materials of Experiment 3: left constituent and right constituent (number of responses according to the constituent, number of responses according to the rime).

handel geschiedenis (20, 0); idee waarde (13, 7); bij controle (13, 7); ezel tong (17, 3); levensmiddel organisatie (19, 1); specerij zee (20, 0); dominee pensioen (16, 4); engel dienaar (19, 1); schilderij paar (20, 0); duivel plicht (20, 0).

## Appendix C

Materials of Experiment 4: List A: definition plus left and right compound constituent (number of responses according to the bias of the suffix, number of responses according to the bias of the letter combination, number of other responses).

Een 'PLOEver' is een boomsoort.	PLOEver_gried (12, 5, 3)
In een glas 'WILter' zit veel alcohol.	WILter_boest (11, 8, 1)
Een 'VIEber' is een blaasinstrument.	VIEber_gedij (5, 12, 3)
Een 'VOEStegaar' is een verdedigingstactiek.	VOEStegaar_sien (9, 7, 4)
Een 'MOEnaar' is een visvergunning.	MOEnaar_gezoel (9, 2, 9)
Iets wat zeldzaam is noemen we een 'BOEzaar'.	BOEzaar_turei (13, 4, 3)
Mediterrane vegetatie heet ook wel 'ROEzing'.	ROEzing_nast (12, 2, 6)
'PRIEling' is een kruidensoort.	PRIEling_faren (14, 3, 3)
Een 'KRONving' is een muziekstuk.	KRONving_doef (11, 1, 8)
'BinTIST' is een Oosters gerecht.	binTIST_zaste (16, 3, 1)
Een 'baraFIST' is een opslagtank.	baraFIST_modee (13, 5, 2)
'GisoFIST' is een Belgisch biermerk.	gisoFIST_buroop (13, 7, 0)

Materials of Experiment 4: List B: definition plus left and right compound constituent (number of responses according to the bias of the suffix, number of responses according to the bias of the letter combination, number of other responses).

Iemand die graag 'ploeft' is een 'PLOEver'.	PLOEver_gried (17, 3, 0)
Iemand die 'wilt' is een 'WILter'.	WILter_boest (14, 5, 1)
Een persoon die goed 'viebt' is een 'VIEber'.	VIEber_gedij (13, 6, 1)
Iemand die graag 'voest' is een 'VOEStegaar'.	VOEStegaar_sien (10, 5, 5)

Degene die ‘moent’ is de ‘MOEnaar’.	MOEnaar_gezoel (16, 2, 2)
De persoon die ‘boest’ is de ‘BOEzaar’.	BOEzaar_turei (9, 4, 7)
Het ‘roezen’ van iets heet de ‘ROEzing’.	ROEzing_nast (12, 3, 5)
Het ‘prielen’ van iets is de ‘PRIEling’.	PRIEling_faren (15, 2, 3)
Het resultaat van het ‘kronven’ is de ‘KRONving’.	KRONving_doef (12, 0, 8)
Iemand die een ‘bint’ bespeelt is de ‘binTIST’.	binTIST_zaste (17, 3, 0)
De ‘baraaf’ wordt bespeeld door de ‘baraFIST’.	baraFIST_modee (18, 2, 0)
De ‘gisoof’ wordt gemaakt door de ‘gisoFIST’.	gisoFIST_buroop (19, 1, 0)

## Appendix D

Rules applied to the CELEX compounds. If first constituent

- ends in shwa plus sonorant, use  $-\emptyset$ .
- ends in a full vowel, use  $-\emptyset$ .
- has the feature  $\langle -\text{human} \rangle$  and ends in *-er*, *-eur*, *-ier*, *-aar*, or *-air*, use *-s*.
- has the feature  $\langle +\text{human} \rangle$  and ends in *-ist*, *-erik*, *-es*, *-in*, *-aan/-iaan*, *-ling/-eling*, *-uur*, *-ant*, *-ent*, *-aat*, *-iet*, *-aal*, *-eel*, *-iel*, *-loog*, or *-graaf*, use *-e(n)*.
- has the feature  $\langle +\text{human} \rangle$  and ends in *-ette*, use  $-\emptyset$ .
- has the feature  $\langle +\text{human} \rangle$  and ends in *-or*, use *-s* or *-e(n)*.
- has the feature  $\langle -\text{human} \rangle$  and ends in *-uur*, in *-ant*, in *-iet*, in *-aal*, in *-eel*, in *-iel*, in *-loog*, in *-graaf*, *-air*, or *-or*, use  $-\emptyset$ .
- has the feature  $\langle -\text{animate} \rangle$  and ends in *-er*, *-eur*, *-ier*, *-ette*, or *-in*, use  $-\emptyset$ .
- has the feature  $\langle -\text{animate} \rangle$  and ends in *-er*, *-eur*, *-ier*, *-ette*, or *-in*, use  $-\emptyset$ .
- has the feature  $\langle -\text{countable} \rangle$  and ends in *-teit/-iteit*, *-schap*, *-ing*, or *-dom*, use *-s*.
- has the feature  $\langle +\text{countable} \rangle$  and ends in *-teit/-iteit*, *-schap*, *-dom*, *-dij/-erij/-arij*, or *-nis*, use *-e(n)*.
- has the feature  $\langle -\text{countable} \rangle$  and ends in *-isme*, *-nis*, *-ij/-erij/-arij*, or *-ade/-ide/-ode*, use *-s*.
- has the feature  $\langle +\text{countable} \rangle$  and ends in *-isme*, *-nis*, or *-ade/-ide/-ode*, use *-n*.
- has the feature  $\langle +\text{countable} \rangle$  and ends in *-ing*, use  $-\emptyset$ .
- ends in *-heid*, or *-(t)je*, use *-s*.
- ends in *-te/-de*, *-sel*, *-sie/-tie*, *-um*, *-theek*, *-aris*, or *-us*, use  $-\emptyset$ .

PART V

## Extending Analogical Modeling



## Expanding $k$ -NN analogy with instance families\*

Antal van den Bosch

### 1. Instance families: An implementation of the analogical set

A marked difference between the  $k$ -nearest-neighbor ( $k$ -NN) rule (Cover & Hart 1967) on the one hand, and analogical modeling (AM) (Skousen 1989) on the other hand, is their rigid ( $k$ -NN) versus dynamic and global (AM) bias in collecting evidence from memorized instances for the classification of a new instance. Especially with  $k = 1$  (a common setting in  $k$ -NN) only the set of minimally- and equally-differing nearest neighbors is used for determining the class of a new instance. Fixing  $k$  ignores the fact that an instance is often surrounded in instance space by a number of instances of the same class that is actually larger or smaller than  $k$ . We refer to such a variable-sized set of same-class nearest neighbors as an *instance family*.

Instance families, when generalized before the actual classification of new instances, can be seen as generalized instances that can be used in the same way as normal instances in regular  $k$ -NN classification. The key difference between  $k$ -NN classification based on instances versus families is that families can match new instances that contain value combinations *that have not been observed in individual memorized instances*. The idea of precompiling an instance base into instance families and then using these families for further  $k$ -NN classification has been implemented in the FAMBL algorithm (van den Bosch 1999).

The precompilation step within FAMBL2 is the main difference with analogical modeling (Skousen 1989); the latter algorithm compiles analogical sets only during classification. Consequently, in AM as many different analogical sets can occur as there are test instances, and they are not in practice stored in memory.

Precompiling generalized instances has a potential advantage of memory compression; less memory is needed when many instances can be summarized by a small number of families. Second, there is a potential classification speed advantage, since less memory items need to be traversed in  $k$ -NN classification. Never-



theless, our major interest lies in the effects that generalizing instances may have on generalization accuracy. The central question addressed in this contribution is whether FAMBL2 benefits from its strategy in that respect, when applied to natural language processing tasks. From the results obtained in a range of experiments, we conclude that generalizing instances has the prospected effects of memory compression and bringing, implicitly, more than a rigid number of instances in the nearest-neighbor set. However, the results show that the net effects in generalization accuracy are generally small and not significantly different when compared to standard  $k$ -NN classification with information-gain-ratio feature weighting as implemented in 1B1-IG (Daelemans & van den Bosch 1992; Daelemans, van den Bosch, & Weijters 1997).

In this contribution, we start with a description of the FAMBL2 algorithm in Section 2. We then report in Section 3 on experiments on a range of natural language processing tasks with FAMBL2 and standard weighted  $k$ -NN. We summarize our findings and discuss the relation between FAMBL2 and analogical modeling in Section 4.

## 2. FAMBL2: Description of algorithm

Memory-based learning, also known as instance-based, example-based, lazy, case-based, exemplar-based, locally weighted, and analogical learning (Stanfill & Waltz 1986; Aha, Kibler, & Albert 1991; Salzberg 1991; Kolodner 1993; Aha 1997; Atkeson, Moore, & Schaal 1997), is a class of supervised inductive learning algorithms for learning classification tasks (Shavlik & Dietterich 1990). Memory-based learning treats a set of labeled (pre-classified) training instances as points in a multi-dimensional feature space, and stores them as such in an *instance base* in memory (rather than performing some abstraction over them).

An instance consists of a fixed-length vector of  $n$  feature-value pairs, and an information field containing the classification of that particular feature-value vector. After the instance base is built, new (test) instances are classified by matching them to all instances in the instance base, and by calculating with each match the *distance*, given by a distance function  $\Delta(X, Y)$  between the new instance  $X$  and the memory instance  $Y$ . The memory instances with the smallest distances are collected, and the classifications associated with these nearest neighbors are merged and extrapolated to assign a classification to the test instance.

The most basic distance function for patterns with symbolic features is the *overlap metric*  $\Delta(X, Y) = \sum_{i=1}^n \delta(x_i, y_i)$ , where  $\Delta(X, Y)$  is the distance between patterns  $X$  and  $Y$ , represented by  $n$  features, and  $\delta$  is the distance between feature values,  $\delta(x_i, y_i) = 0$  if  $x_i = y_i$ , else 1. Classification in memory-based learning

systems is basically performed by the  $k$ -nearest neighbor ( $k$ -NN) classifier (Cover & Hart 1967; Devijver & Kittler 1982), with  $k$  usually set to 1.

Early work on the  $k$ -NN classifier pointed at advantageous properties of the classifier in terms of generalization accuracies, under certain assumptions, because of its reliance on full memory (Fix & Hodges 1951; Cover & Hart 1967). However, the trade-off downside of full memory is the resulting computational inefficiency of the classification process, as compared to parametric classifiers that do abstract from the learning material. Therefore, several early investigations proposed *editing* methods: namely, finding criteria for the removal of instances from memory (Hart 1968; Gates 1972) without harming classification accuracy. Other studies on editing also explored the possibilities of detecting and removing noise from the learned data, so that classification accuracy might even improve (Wilson 1972; Devijver & Kittler 1980). The renewed interest in the  $k$ -NN classifier from the late 1980s onwards in the  $\Delta 1$ -subfield of machine learning (Stanfill & Waltz 1986; Stanfill 1987; Aha, Kibler, & Albert 1991; Salzberg 1991) resulted in several new implementations for editing, but also other approaches to abstraction in memory-based learning emerged. We identify three types:

1. **Editing** (Hart 1968; Wilson 1972; Aha, Kibler, & Albert 1991): removing instances (according to a classification-related utility) that do not reach a given threshold. Editing is not careful in principle, but the approaches that are discussed here and that are included in the empirical comparison – i.e.  $\mathbb{B}_2$  and  $\mathbb{B}_3$  (Aha, Kibler, & Albert 1991) – collect statistical support for the conclusion that an editing operation can be harmless.
2. **Oblivious (partial) decision-tree abstraction** (Daelemans, van den Bosch, & Weijters 1997): compressing (parts of) instances in the instance base into (parts of) decision-trees. Part of the motivation to perform top-down induction of decision trees ( $\tau$ DIDT) is the presence of clear differences in the relative importance of instance features, allowing features to be strictly ordered in matching (Quinlan 1986). The approach is dependent on the use of a feature-weighting metric.
3. **Carefully merging instances** (Salzberg 1991; Wettschereck & Dietterich 1995; Domingos 1996): merging multiple instances in single generalized instances. Generalized instances can be represented by conjunctions of *disjunctions* of feature values, which is equivalent to rules with wild-cards.

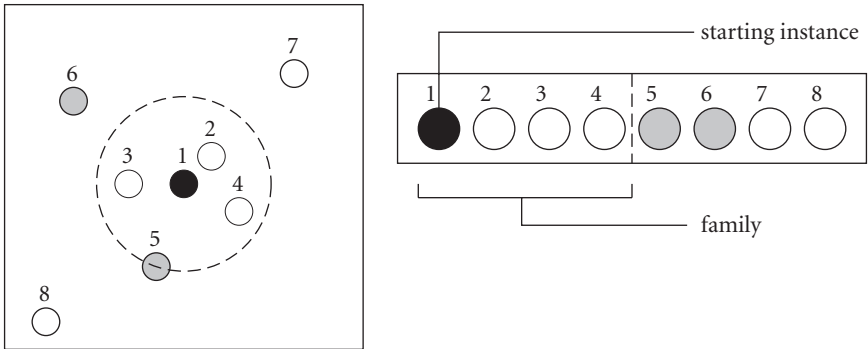
Here we describe FAMBL2, belonging to the third group of carefully-abstracting memory-based learning algorithms. FAMBL2 merges groups of very similar instances (called families) into family expressions. The core idea of FAMBL2 is to transform an instance base into a set of *instance family expressions*. First, we outline the ideas and assumptions underlying FAMBL2. We then give a procedural description of the learning algorithm.

2.1 Instance families: Definition

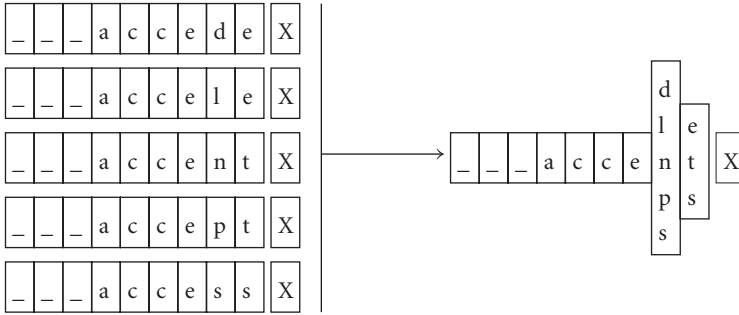
Classification of a new instance in memory-based learning involves a search for the nearest neighbors of that instance. The value of  $k$  in  $k$ -NN determines how many of these neighbors are used for extrapolating their (majority) classification to the new instance. A fixed  $k$  ignores the fact that an instance is often surrounded in instance space by a number of instances of the same class that is actually larger or smaller than  $k$ . We refer to a varying set of same-class nearest neighbors as an instance's *family*. The extreme cases are, on the one hand, instances that have a nearest neighbor of a different class – i.e. they have no family members and are a family on their own – and on the other hand, instances that have as nearest neighbors all other instances of the same class.

Thus families are class clusters, and the number and sizes of families in a data set reflect the *disjunctivity* of the data set – that is, the degree of scatteredness of classes into clusters. In real-world data sets, the situation is generally somewhere between the extremes of total disjunctivity (one instance per cluster) and no disjunctivity (one cluster per class). Many types of language data appear to be quite disjunct (Daelemans, van den Bosch, & Zavrel 1999). In highly disjunct data, classes are scattered among many small clusters, which means that instances have few nearest neighbors of the same class (on average).

Figure 1 illustrates how FAMBL2 determines the family of an instance in a simple two-dimensional instance space. All nearest neighbors of a starting instance (marked by the black dot) are searched and ranked in the order of their distance to



**Figure 1.** An example of a family in a two-dimensional instance space (left). The family, at the inside of the dotted circle, spans the focus instance (black) and the three nearest neighbors labeled with the same class (white). When ranked in the order of distance (right), the family boundary is put immediately before the first instance of a different class (grey).



**Figure 2.** An example of family creation in FAMBL2. Four grapheme-phoneme instances, along with their token occurrence counts (left), are merged into a family expression (right).

the starting instance. Although there are five instances of the same class in the example space, the family of the starting instance contains only three instances, since its fourth-nearest instance is of a different class.

Families are converted in FAMBL2 to *family expressions* (which are hyperrectangles) by merging all instances belonging to that family simultaneously. Figure 2 illustrates the creation of a family expression from an instance family. The general modus of operation of FAMBL2 is that it picks instances from an instance base one by one from the set of instances that are not already part of a family. For each newly-picked instance, FAMBL2 determines its family, generates a family expression from this set of instances, and then marks all involved instances as belonging to a family (so that they will not be picked as starting point or member of another family). FAMBL2 continues determining families until all instances are marked as belonging to a family.

The ordering of instances to be picked as centers of new families is important. Whenever instances are encapsulated in a family, they cannot be the starting point of another. However, one of these instances could have been a better starting point (e.g., because it is the central same-class nearest neighbor of a larger group of instances). Intuitively, it would be best to start building families with those instances that are the middle instances of the largest families. Although this may appear circular, it is possible to estimate the appropriateness of an instance to be a starting point for family generation, by computing its *class-prediction strength* (CPS), which expresses the success of that instance in predicting the class of its surrounding nearest-neighbor instances. Instances in the middle of large families will have high class-prediction strengths; when used for classification, these instances will serve as correct nearest neighbors to a large number of instances surrounding them.

Class-prediction strength (CPS) of an instance  $i$  is typically defined (Salzberg 1990; Domingos 1995) as the number of times the instance is a nearest neighbor of a training instance regardless of its class ( $N$ ), minus the number of these nearest neighbors that are of a different class ( $N\delta_i$ ), divided by  $N$  to express a portion between 0.0 and 1.0:  $e_i = \frac{N_i - N\delta_i}{N_i}$ . An instance with class-prediction strength  $e = 1.0$  is a perfect predictor of its own class; an  $e$  near or at 0.0 indicates that the family is a bad predictor. As argued in Domingos 1995, this “raw” class prediction strength has a bias towards low-frequent instances that is sometimes unwanted: it assigns a maximal score of 1.0 to an instance when it is used correctly as a nearest neighbor only once. The Laplace correction is a common operation that favors high-frequent over low-frequent instances with the same raw score. Laplace correction introduces the number of classes  $c$  into the equation:

$$e_i = \frac{(N_i - N\delta_i) + 1}{N_i + c}.$$

To compute CPS, we perform an auto-classification test with standard  $k$ -NN plus (by default) information-gain-ratio feature weighting as implemented in the IB1-IG algorithm (Daelemans & van den Bosch 1992; Daelemans, van den Bosch, & Weijters 1997).<sup>1</sup> In this experiment,  $k$  is set to 3. This means that not only all nearest (equidistant) neighbors of an instance that differ in one or two features are taken into account, but also the instance itself. This ensures that all instances receive some baseline non-null score, reflecting the intuition that in language data, low-frequent events may also reoccur and thus be a nearest neighbor to a new occurrence of themselves unless they are true noise (Daelemans, van den Bosch, & Zavrel 1999). The same  $k = 3$  limit is used when searching for nearest neighbors in family creation. This means that family members are allowed to differ in two features maximally.

In the FAMBL2 algorithm, instances are ordered by their CPS, and are picked as starting points for new instances beginning with the instance with the highest CPS. This is the key difference with the original FAMBL algorithm, in which starting points were selected randomly (van den Bosch 1999). To summarize, a pseudocode description of the learning phase is given in Figure 3.

After learning, the original instance base is discarded, and further classification is based only on the set of family expressions yielded by the family-extraction phase. Classification in FAMBL2 works analogously to classification in pure memory-based learning: a match is made between a new test instance and all stored family expressions. When a family expression records a disjunction of values for a certain feature, matching is perfect when one of the disjunctive values matches the value at that feature in the new instance. When two or more family expressions of different classes match equally well with the new instance, the class is selected with the highest occurrence summed over the matching expressions. When the tie

---

Procedure FAMBL LEARNING PHASE:

Input: A training set  $TS$  of instances  $I_{1\dots m}$ , each instance being labeled with a family-membership flag set to  $FALSE$

Output: A family set  $FS$  of family expressions  $F_{1\dots m}$ ,  $m \leq n$

$i = f = 0$

1. Determine the class-prediction strength of all instances  $I_{1\dots m}$  in  $TS$  and order them (largest CPS first)
  2. While not all family-membership flags are  $TRUE$ , Do
    - While the family-membership flag of  $I_i$  is  $TRUE$  Do increase  $i$
    - Compute  $NS$ , a ranked set of nearest neighbors to  $I_i$  with the same class as  $I_i$ , among all instances with family-membership flag  $FALSE$ . Nearest-neighbor instances of a different class with family-membership flag  $TRUE$  are still used for marking the boundaries of the family.
    - Select all members in  $NS$  that fall within the 3 closest  $k$  buckets ( $k = 3$ ) and remove all other instances from  $NS$
    - Set the membership flags of  $I_i$  and all remaining instances in  $NS$  to  $TRUE$
    - Merge  $I_i$  and all instances in  $NS$  into the family expression  $F_f$  and store this expression along with a count of the number of instance merged in it
    - $f = f + 1$
- 

**Figure 3.** Schematized overview of the learning (family-extraction) phase in FAMBL.

remains, the class is selected that occurs the most frequently in the complete family expression set.

We conclude our description of the FAMBL2 algorithm by noting that FAMBL2 allows for the inclusion of informational abstraction in the form of feature-weighting, instance-weighting and value-difference metrics. For comparison with IB1-IG, as described in the next section, we have included information-gain-ratio feature weighting in FAMBL2. Weighting metrics are likely to have a profound effect on family extraction. For example, a study by van den Bosch (1997) suggests that using information-gain feature weighting (Quinlan 1986) in pure memory-based learning (viz. IB1-IG, in Daelemans & van den Bosch 1992), can yield considerably bigger families.

### 3. Effects of family generalization

In our comparative experiments between FAMBL2 and standard IB1-IG, we are interested in the differences caused by FAMBL2's family generalization stage. Given a test instance, FAMBL2 and IB1-IG may assign the same correct or false classification

based on different classifications (e.g. one family in FAMBL2 and three instances in IB1-IG), but they may also disagree – in which case one of them may be right, or they may be both wrong.

An illustration of a difference that actually occurs between FAMBL2 and IB1-IG trained on a dataset of English word pronunciation is the following. Consider a small family generated by FAMBL2, from two instances representing the /l/ pronunciation of the “l” in “singularize” and “angularity”, supposing that instances represent one focus letter and four left and right neighboring letters to represent the context. This family is generalized in FAMBL2 as [i or a][ngulari][z or t]. Upon presentation of the instance representing the unseen word “singularity”, the generalized family expression offers a complete match with the new instance, due to the disjunction in the [i or a] and [z or t] parts of the expression, producing the correct /l/ pronunciation. Given the same test word, IB1-IG would yield two best-matching nearest neighbors with each one mismatching on one feature, while producing the same correct classification. In general, using family expressions for classification strengthens the class votes of the instances generalized in families: it can move their class votes up in the *k*-ranking (but never down).

To investigate the occurring differences in detail, we have collected results on datasets representing English grapheme-phoneme conversion, Dutch diminutive noun formation, German plural noun formation, English part-of-speech tagging, English base-noun-phrase chunking, and English prepositional-phrase attachment. We briefly describe these six datasets here.

**English grapheme-phoneme conversion** (henceforth referred to as GP) is the mapping of English words to their phonemic counterparts, where the classification occurs at the letter level: mappings are made between letters in context and their appropriate phonemes. The grapheme-phoneme conversion data used in the experiments described here is derived from the CELEX lexical data base (Baayen, Piepenbrock, & van Rijn 1993). We have used the first of the ten partitionings from the 10% 10-fold cross-validation experiment described in van den Bosch 1999.

**Dutch diminutive formation** (henceforth DIM) selects the correct diminutive inflection to Dutch nouns out of five possibilities (*je*, *tje*, *pje*, *kje*, and *etje*) on the basis of phonemic word transcriptions segmented at the level of syllable onset, nuclei, and coda for the final three syllables of the word. The data stems from a study described in Daelemans, Berck, and Gillis 1997.

**German plural formation** (henceforth PLU) predicts the correct plural inflection (with possible umlaut) out of 8 possibilities, on the basis of singular nouns represented by their phonemic representation segmented at the level of syllable onset, nuclei, and coda for the final three syllables of the word. The data is also described (and tested on) in Daelemans’s article (in this volume).

**English part-of-speech tagging** (henceforth  $\text{POS}$ ) involves the disambiguation of syntactic classes of words for particular contexts. We assume a tagger architecture that processes a sentence from a disambiguated left to an ambiguous right context, as described in Daelemans, Zavrel, Berck, & Gillis 1996. The original data set for the part-of-speech tagging task, extracted from the LOB corpus, contains 1,046,151 instances; we have used a randomly-extracted 10% of this data.

**English base-NP chunking** (henceforth  $\text{NP}$ ) predicts the segmentation of sentences into non-recursive NPs. Veenstra (1998) used the Base-NP tag set as presented by Ramshaw and Marcus (1995): *I* for inside a Base-NP, *O* for outside a Base-NP, and *B* for the first word in a Base-NP following another Base-NP. See Veenstra 1998 for more details, and Daelemans, van den Bosch, & Zavrel 1999 for a series of experiments on the original data set from which we have used a randomly-extracted 10%.

**English PP attachment** (henceforth  $\text{PP}$ ) is the attachment of a prepositional phrase  $\text{PP}$  in the sequence  $\text{VP NP PP}$  ( $\text{VP}$  = verb phrase,  $\text{NP}$  = noun phrase,  $\text{PP}$  = prepositional phrase). The data consists of four-tuples of words, extracted from the Wall Street Journal Treebank. From the original data set (Ratnaparkhi, Reynar, & Roukos 1994; Collins & Brooks 1995; Zavrel, Daelemans, & Veenstra 1997), Daelemans, van den Bosch, and Zavrel (1999) took the train and test set together to form the particular data used here.

In all experiments, both  $\text{FAMBL2}$  and  $\text{IB1-IG}$  use information-gain-ratio feature weighting (Quinlan 1986), which appears crucial in producing adequate  $k$  rankings and is an important difference with standard  $\text{AM}$  (Daelemans, van den Bosch, & Weijters 1997). For each data set we generated a single random partitioning into a 90% training set and a 10% test set. Classification was done by both algorithms using  $k = 1$ ;  $\text{IB1-IG}$  finds the set of closest nearest neighbors that all differ in the same zero or more features, and  $\text{FAMBL2}$  does the same for families.

Table 1 displays the overall generalization accuracies yielded by the two algorithms on the six test sets. Accuracy differences are small. Yet, reasonable compression is obtained in the number of families produced by  $\text{FAMBL2}$  when compared to the number of instance types (i.e. the number of unique instances, without duplicates) maintained in  $\text{IB1-IG}$ 's memory. These results are in line with the findings reported in van den Bosch 1999:  $\text{FAMBL}(2)$  compresses, but does not improve generalization accuracy as compared to  $\text{IB1-IG}$ .

There are more differences in the classifications made by the two algorithms when focusing on the instance level and when looking for differences in the sense of the “singularize” – “angularity” – “singularity” example given above. In the first case, Table 2 lists the average distance between a test instance and its nearest neighbors in  $\text{IB1-IG}$  and  $\text{FAMBL2}$ . These results indicate that classifications by  $\text{FAMBL2}$  are



**Table 1.** Generalization accuracies in percentages and absolute numbers of correctly classified test instances for **IB1-IG** and **FAMBL2** on single partitionings of the six tasks; numbers of families; and compression rates of **FAMBL2** versus **IB1-IG** in terms of numbers of families versus numbers of instance types.

task	generalization accuracy		number of families	fam. vs inst. compression
	IB1-IG	FAMBL2		
GP	88.1% (5975/6781)	87.9% (5962/6781)	31862	41.2%
DIM	95.4% (377/395)	96.2% (380/385)	1893	46.3%
PLU	94.8% (2385/2517)	94.6% (2377/2517)	4742	61.5%
POS	96.6% (10105/10462)	96.6% (10102/10462)	21802	71.0%
NP	97.5% (2448/2512)	97.5% (2448/2512)	17386	22.2%
PP	80.8% (1932/2390)	79.9% (1909/2390)	6980	65.9%

**Table 2.** Average distance between test instances and their nearest neighbors for **IB1-IG** and **FAMBL2**, plus the percentage of distance decrease obtained by **FAMBL2**, measured on the six tasks.

task	average distance to nearest neighbor		% closer
	IB1-IG	FAMBL2	
GP	0.119	0.115	3.3%
DIM	0.075	0.071	5.3%
PLU	0.028	0.026	5.8%
POS	0.153	0.141	7.8%
NP	0.151	0.151	0.1%
PP	0.054	0.052	3.3%

**Table 3.** Test classification disagreement statistics

task	total # disagree	both wrong	IB1-IG right		FAMBL2 right	
			further	distr.	closer	distr.
GP	102	15	24	26	17	20
DIM	3	0	0	0	0	3
PLU	48	8	9	15	1	15
POS	76	9	10	25	5	27
NP	0	0	0	0	0	0
PP	131	0	21	56	11	43
<b>total</b>	<b>360</b>	<b>32</b>	<b>64</b>	<b>122</b>	<b>34</b>	<b>108</b>

indeed based on nearest neighbors at closer distances. As with the “singularity” example, at least some nearest neighbors are merged and appear as closer families in FAMBL2 classification. On average, FAMBL2 finds nearest neighbors at about 5% closer distance.

In the second case, Table 3 displays detailed counts of cases in which the two algorithms disagree on the classification of a test instance. We discern five possible situations involving disagreements:

1. both algorithms are wrong;
2. IB1-IG is right, although FAMBL2 found one or more nearest families at a closer distance;
3. IB1-IG is right, finding nearest neighbors at the same distance as FAMBL2, but having a class distribution in which the correct class is the most frequent, while FAMBL2 has an incorrect class as the most frequent class (because one or more families entered the  $k = 1$  nearest-neighbor set, carrying incorrect classes with them);
4. FAMBL2 is right, finding one or more nearest families at a closer distance than the nearest neighbors found by IB1-IG;
5. FAMBL2 is right, finding nearest neighbors at the same distance as IB1-IG, but having a class distribution in which the correct class is the most frequent class, while IB1-IG has an incorrect class as the most frequent class.

The five columns in Table 3 display the absolute numbers within these five outcome types yielded by the two algorithms on the six tasks. FAMBL2 and IB1-IG agree completely on the NP task. In the majority of the disagreements, one of the two algorithms has the better class distribution, while both have nearest neighbors at the same close distance (tasks PLU and NP). Overall, the results confirm that FAMBL2 does classify differently from IB1-IG, but the net effect as compared to the accuracies yielded by IB1-IG is slightly negative, but close to zero.

## 4. Discussion

First, we summarize the findings reported in the previous section and draw conclusions from them. We then discuss the relation between memory-based learning, careful abstraction in FAMBL2, and analogical modeling.

### 4.1 Summary of findings and conclusions

Merging instances to form families, and then using these precompiled families to base classifications on, is a close alternative to standard memory-based learning

as implemented in the IBI-IG algorithm. FAMBL2 is able to reach comparable levels of generalization accuracy, while obtaining memory compression rates ranging from 20 to 80%. On the other hand, FAMBL2's learning phase is a computationally costly procedure (though not exponential) as it involves a complete memory-based classification of the training set.

The effect of merging instances to families (thereby opening up the possibility that they jointly end up in a less distant  $k$ -level of nearest neighbors) has a negligible net effect on generalization accuracy as compared to standard memory-based learning. When a family moves its class up a  $k$ -level, results show that this movement improves and deteriorates class distributions roughly equally often.

In sum, pure memory-based learning remains a recommendable choice due to its simplicity. Unlike FAMBL2 and AM, it bases its classifications only on nearest neighbors. With natural language processing tasks, this seems to be the best overall strategy available. No reported results with FAMBL (van den Bosch 1999) or FAMBL2, or results obtained in comparisons between standard memory-based learning and AM (Daelemans, in this volume) show a trend towards an advantage of using instances further away (i.e. beyond a low, fixed  $k$  of nearest neighbors) in classification.

#### 4.2 Relation with analogical modeling

In terms of the original AM algorithm (Skousen 1989), standard  $k$ -NN classification takes only the instances in no more than the  $k$  most specific supracontexts that contain observations, whether these are homogeneous or not, as the basis for extrapolating the output class. The lack of a check on homogeneity may be a cause for the general finding that increasing  $k$  is often detrimental to generalization accuracy with standard  $k$ -NN on language processing tasks (Daelemans, van den Bosch, & Zavrel 1999). In FAMBL2, with the same  $k$ , generating a disjunction of values in the generalized family expression entails an explicit *union* of adjacent homogeneous supracontexts that have the same class label. In classification, these joined supracontexts henceforth act as one, representing a potentially higher number of instance pointers and hence a higher analogical effect in further classification. A major difference with AM is that FAMBL precompiles its families, usually down to a set of family expressions that is (considerably) smaller than the original instance set. In AM, generating the analogical set is done for each test instance. There can be many more analogical sets than training instances. It would, however, be interesting to explore the possibilities of precompiling within AM a limited set of analogical sets on the basis of the training set, before classification.

To conclude, AM strongly suggests that more instances should be encapsulated in analogical sets than just the  $k$  closest matches. Implementing this general idea

in a strictly limited manner as family generalization appears to be a valid step in bridging the gap between the  $k$ -NN and  $\Delta M$  approaches and finding the general class of algorithms that combines the best of both worlds.

## Notes

\* The author wishes to thank Walter Daelemans, Jakub Zavrel, and the other members of the ILK (Tilburg) and CNTS (Antwerp) research groups for fruitful discussions and criticisms. This research has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences.

1. Auto-classification with 1B1-IG is performed using the TiMBL software package, version 3.0.2 (Daelemans, Zavrel, van der Sloot, & van den Bosch 1999).

## References

- Aha, David W. (1997). Lazy learning: Special issue editorial. *Artificial Intelligence Review*, 11, 7–10.
- Aha, David W., Dennis Kibler, & Marc K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Atkeson, Chris, Andrew Moore, & Stefan Schaal (1997). Locally weighted learning. *Artificial Intelligence Review*, 11, 11–73.
- Baayen, R. Harald, Richard Piepenbrock, & Hedderik van Rijn (1993). *The CELEX lexical data base on CD-ROM*. Philadelphia, PA: Linguistic Data Consortium.
- Collins, Michael, & James Brooks (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of third workshop on very large corpora*. Cambridge.
- Cover, Thomas M., & Peter E. Hart (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13, 21–27.
- Daelemans, Walter, Peter Berck, & Steven Gillis (1997). Data mining as a method for linguistic analysis: Dutch diminutives. *Folia Linguistica*, 31, 57–75.
- Daelemans, Walter, & Antal van den Bosch (1992). Generalisation performance of back-propagation learning on a syllabification task. In M. F. J. Drossaers & A. Nijholt (Eds.), *Proceedings of TWLT3: Connectionism and natural language processing* (pp. 27–37). Enschede: Twente University.
- Daelemans, Walter, Antal van den Bosch, & A. J. M. M. Weijters (1997). 1Gtree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11, 407–423.
- Daelemans, Walter, Antal van den Bosch, & Jakub Zavrel (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 34, 11–43.
- Daelemans, Walter, Jakub Zavrel, Peter Berck, & Steven Gillis (1996). MBT: A memory-based part of speech tagger generator. In E. Ejerhed & I. Dagan (Eds.), *Proceedings of fourth workshop on very large corpora* (pp. 14–27). ACL SIGDAT.

- Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, & Antal van den Bosch (1999). TiMBL: Tilburg Memory Based Learner, version 3.0, reference manual. Technical Report ILK-0001, ILK, Tilburg University.
- Devijver, Pierre A., & Josef Kittler (1980). On the edited nearest neighbor rule. In *Proceedings of the fifth international conference on pattern recognition*. The Institute of Electrical and Electronics Engineers.
- Devijver, Pierre A., & Josef Kittler (1982). *Pattern recognition, a statistical approach*. London, UK: Prentice-Hall.
- Domingos, Pedro (1995). The RISE 2.0 system: A case study in multistrategy learning. Technical Report 95-2, University of California at Irvine, Department of Information and Computer Science, Irvine, CA.
- Domingos, Pedro (1996). Unifying instance-based and rule-based induction. *Machine Learning*, 24, 141–168.
- Fix, Evelyn, & J. L. Hodges, Jr (1951). Discriminatory analysis – nonparametric discrimination; consistency properties. Technical Report Project 21-49-004, Report No. 4, USAF School of Aviation Medicine.
- Gates, Geoffrey W. (1972). The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18, 431–433.
- Hart, Peter E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14, 515–516.
- Kolodner, Janet L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann.
- Quinlan, J. Ross (1986). Induction of Decision Trees. *Machine Learning*, 1, 81–206.
- Ramshaw, Lance A., & Mitchell P. Marcus (1995). Text chunking using transformation-based learning. In *Proceedings of third workshop on very large corpora* (pp. 82–94).
- Ratnaparkhi, Adwait, Jeff Reynar, & Salim Roukos (1994). A maximum entropy model for prepositional phrase attachment. In *Workshop on human language technology*. Plainsboro, NJ: ARPA.
- Salzberg, Steven (1990). *Learning with nested generalised exemplars*. Norwell, MA: Kluwer Academic Publishers.
- Salzberg, Steven (1991). A nearest hyperrectangle learning method. *Machine Learning*, 6, 277–309.
- Shavlik, Jude W., & Thomas G. Dietterich (Eds.). (1990). *Readings in Machine Learning*. San Mateo, CA: Morgan Kaufmann.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Stanfill, Craig (1987). Memory-based reasoning applied to English pronunciation. In *Proceedings of the sixth national conference on artificial intelligence (AAAI-87)* (pp. 577–581). Los Altos, CA: Morgan Kaufmann.
- Stanfill, Craig, & David L. Waltz (1986). Toward memory-based reasoning. *Communications of the ACM*, 29, 1213–1228.
- van den Bosch, Antal (1997). Learning to pronounce written words: A study in inductive language learning. Ph.D. thesis, Universiteit Maastricht.
- van den Bosch, Antal (1999). Careful abstraction from instance families in memory-based language learning. *Journal for Experimental and Theoretical Artificial Intelligence*, 11, 339–368.

- Veenstra, Jorn B. (1998). Fast NP chunking using memory-based learning techniques. In *Proceedings of benelearn'98*. Wageningen, The Netherlands.
- Wettschereck, Dietrich, & Thomas G. Dietterich (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19, 1–25.
- Wilson, Dennis L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *Institute of Electrical and Electronic Engineers Transactions on Systems, Man and Cybernetics*, 2, 408–421.
- Zavrel, Jakub, Walter Daelemans, & Jorn Veenstra (1997). Resolving PP attachment ambiguities with memory-based learning. In M. Ellison (Ed.), *Proceedings of the workshop on computational language learning (CoNLL'97)*. Madrid: ACL.



# Version spaces, neural networks, and Analogical Modeling

Mike Mudrow

## 1. Introduction

While much of the literature on Skousen's Analogical Modeling algorithm has been devoted to showing how this model (Skousen 1989, 1992) is different from other possible cognitive models, it will be my intention in this paper to bridge some of the proposed gaps between AM and a few other well established cognitive models. In doing so, I hope to show that this work is very much in line with mainstream research in cognitive science, while at the same time pointing out some of the unique advantages of the AM model.

In a way, the recent renaissance of analogical theories is a return to the foundations of linguistic studies in western society. It was, after all, the Greek Analogists who first supported the idea that languages were regular by nature. This seems almost ironic, given the way analogy has been viewed in more recent linguistic movements, but make no mistake: analogy is back. Already in the mid-seventies, linguists like Ohala (1974) and Anttila (1977) were laying out specific plans for reintroducing analogy into mainstream linguistics. The mid-eighties saw important publications in the area of morphology, such as Bybee (1985) and Rumelhart and McClelland (1986), whose work has been the inspiration for a new generation of linguists exploring non-rule alternatives to language modeling.

1989 brought the publication of Skousen's *Analogical Modeling of Language*, which introduced arguably the first new mathematical formalization of pure analogy since the advent of four-part analogy and which decisively challenged the long-standing notion that analogy could not be constrained sufficiently to be taken seriously as a model for linguistic behavior (cf. Kiparsky 1974, 1978). Around this same time Spencer (1988) proposed using four-part analogy to handle the generative problem of bracketing paradoxes, and Becker (1990) also published his monograph, *Analogie und morphologische Theorie*, in which he successfully united the worlds of Priscian and Aronoff (1976) by showing that input and output structures



which are directly related to the bases of a proportional analogy could be thought of as algebraically formulated word formation rules.

The last decade of the twentieth century has, of course, experienced a literal explosion in the number of new models based on analogy, either direct or indirect. Many of these are connectionist models of one kind or another (cf. Daugherty & Seidenberg 1994; Holyoak & Thagard 1996; Gasser 1997). These also include many, such as Kruschke (1992), which could be called exemplar-based (cf. Shanks 1995 for an overview). Other such models might include (but are not limited to) Lazy Learning (Aha et al. 1991), Case-Based Reasoning (Riesbeck & Schank 1989) and Data-Oriented Parsing (Bod 1998). What all of these models share is a fundamentally non-declarative approach (cf. Chandler 1995) to linguistic analysis. Many of the exemplar-based models also tend to yield very similar predictions concerning linguistic behavior, and it is our job to try to understand why this might be the case and what each model can teach us about the nature of cognition.

## 2. Version spaces

One powerful conceptual learning mechanism based on generalization from examples involves “version spaces” (Mitchell 1978, 1982). The version space approach was originally developed to improve the efficiency of heuristic searches<sup>1</sup> and can, in principle, be applied to any induction problem. In fact, VanLehn and Ball (1987) have demonstrated that a variation of Mitchell’s version space algorithm is even capable of learning context-free grammars of the sort proposed by Chomsky (1957, 1965). Though very powerful, the concept of a version space is actually quite simple. Basically, version spaces are sets of structured concepts which are related to each other by increasing order of generalization. They are most easily illustrated using a directed acyclic graph such as the one in Figure 1.

This figure shows a graph which represents the sample version space  $V$ . Each node in the graph represents a possible concept that can be described using a number of variables, much like a word can be described using a sequential list of phonemes. The total number of possible variables ( $m$ ) will determine how many elements (nodes) will be in the version space. In this case there are three variables, so the total number of elements will be  $2^m = 8$ . Each of the three variables can take on the following values: 1, 2, 3 or – (unspecified). Every version space graph has a maximal element (the most specific concept) and a minimal element (the most general concept) and the nodes between them are ordered in the following manner: each successive row contains concepts with one more unspecified variable than those in the row above it and there is a line drawn upwards from each concept connecting it to only those concepts in the next highest row which are proper

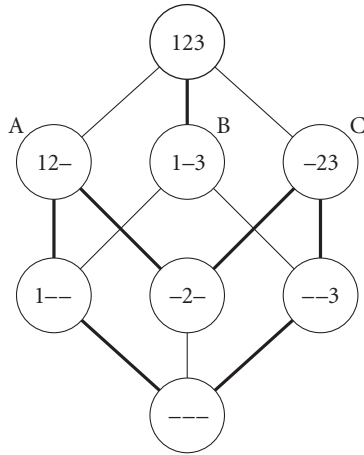


Figure 1. The sample version space  $V$

subsets of it.<sup>2</sup> Thus, the version space  $V$  contains all of the possible supracontexts of the specific concept “123”, given in order of increasing generality going from top to bottom.

One of the major advantages of using a version space when doing a heuristic search is that, unlike with many other inductive search methods, one must not assume that all of the given concepts are available prior to the start of the induction process, but rather can be added one at a time as they become available, thus allowing for natural development in the absence of hard-wired knowledge. Version space algorithms will “create intermediate hypotheses, and they are capable of updating these hypotheses to account for each new instance” (Genesereth & Nilsson 1987: 175). Each time a new exemplar is encountered, another set of possible goal concepts is automatically created (i.e., the nodes of the new version space graph, bounded minimally by a completely unspecified concept and maximally by the new exemplar itself). The list of possible goal concepts is then pared down as positive and negative examples are declared (or discovered) through the process of candidate elimination. Simply stated, this means that the goal concept cannot be more specific than any positive example, nor can it be a generalization of any negative example.

In order to illustrate how this works, let’s put some teeth into our sample version space  $V$ . Although it is tempting to follow the obvious metaphor and treat variables like phonemes, they can in fact represent any attribute of any concept. However, since the realm of phonology is a familiar one to most linguists, let’s set up our sample version space in terms of phonological variables. For instance, the first variable could represent the manner of articulation (stop, fricative, nasal),<sup>3</sup>

the second the place of articulation (labial, alveolar, velar) and the third the voicing (voiceless, voiceless aspirated, voiced) of a given consonant, with the parameters in the parentheses assigned the values 1, 2 and 3 respectively. On this interpretation, the given concept “123” would represent the phoneme /d/, a voiced alveolar stop.

Now look at the three nodes of the version space graph labeled A, B and C. Let’s say that the phoneme /t/ is given as a negative example of our goal concept. This phoneme corresponds to the node labeled A in the graph: an alveolar stop (which is not necessarily voiced). This would allow us to rule out all of the nodes which are below this node and connected to it via bold lines as possible goal concepts, because they are simply generalizations of the negative example. At this point we have gone from a set of eight possible goal concepts down to four, a fifty percent reduction. If the phoneme /g/ is now given as a positive example (node B in the graph: a voiced velar stop), we can further eliminate the maximal node of the set (connected once again with a bold line) as it is more specific than the positive example. Finally, we are given that the phoneme /z/ is also a negative example (node C in the graph: a voiced alveolar fricative), which means we can discount both this node and the node below, since it is a generalization of it, and now we have whittled our version space graph down to only one node (node B) which must represent the goal concept: a voiced stop.

As mentioned before, this is a very simple example of a version space. In practice, the nodes of a version space graph need not symbolize single concepts and could even represent complex rules and formulas.

### 3. Analogical Modeling

At the same time that Mitchell was formulating his Version Spaces model, Skousen (1992) was developing his own model for predicting behavior analogically. Interestingly, Skousen also used this same partial order of generalizations as the basis for his test for supracontextual homogeneity, which is the linchpin of his model. In fact, one could say that there is no essential difference between the supracontextual spaces used in AM and Version Spaces (as defined by Mitchell). However, Skousen’s model is corpus-based, whereas Mitchell’s model relies on the continuous introduction of (labeled) positive and negative examples. In other words, Skousen decided to relate each supracontextual space in AM to a fixed dataset (which may be assumed to be gathered through experience).

One (rather obvious, in hindsight) consequence of this decision was that the space of possible analogical models can be reduced significantly and without any calculations by simply eliminating those supracontexts which are empty (i.e., those which have no specific subcontexts actually occurring in the dataset). In practice,

such reductions can be quite large, given the rather sparse nature of linguistic data. (The proportion of actual words to possible words in a given language is usually quite small.) Unfortunately, this still leaves any actually occurring word in a given dataset which has anything in common with a given context as a possible analogical model, which implies a vague notion of analogy similar to the one which has been the focus of so much criticism over the last three decades.

In order to refine this notion and further pare down these supracontextual spaces, Skousen proposed an algorithm with “three important properties [which] affect the probability of selecting a particular example as an analogical model” (1995:217):

- proximity:** the more similar the example is to the given context, the greater the chances of that example being selected as the analogical model;
- gang effect:** if the example is surrounded by other examples having the same behavior, then the probability of selecting [one of] these similarly behaving examples is substantially increased;
- heterogeneity:** an example cannot be selected as the analogical model if there are intervening examples, with different behavior, closer to the given context.

All three of these properties are derived from Skousen’s (1992) psychologically plausible measure of uncertainty, but only the third property is unique to the AM algorithm. This property could also arguably be the most important of the three, since not only can it affect the probability of selecting a given example from a dataset, it can also eliminate certain examples completely in a manner which drastically affects overall selection probabilities.

Consider the supracontextual space in Figure 2 (adapted from the second chapter of Skousen’s 1989 book). As was the case for the Version Space discussed earlier, there are three variables, each of which can take on the values 0, 1, 2 or 3. The sparse dataset used in this example consists of five of the 64 possible occurrences: 310e, 032r, 210r, 212r and 311r. The “e” and “r” labels associated with each of the five occurrences simply stand for exceptional and regular behavior, but as Skousen is quick to point out, these are used for clarity’s sake and could just as well have been “x” and “y” or any other arbitrary labels. In fact, one of the strengths of AM (and other analogy-based approaches) is that it can predict both regular and exceptional behavior using a single algorithm. Even idiosyncratic data and data which are otherwise noisy or not completely specified are no problem for the model. On the other hand, AM does crucially rely on each occurrence in a given dataset being labeled according to some specific behavior, and this can be shown to result in some undesirable consequences relating to the model’s psychological plausibility, as will be discussed in more detail below.

In this graph homogeneous supracontexts (those which can contribute sub-contexts to the pool of possible analogical models) are surrounded by bold circles.

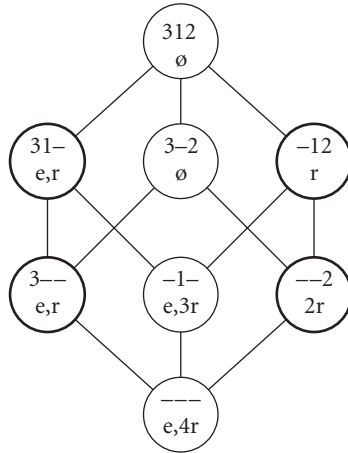


Figure 2. Supracontextual space for the context “312”

Two of the supracontexts can be eliminated right away (before heterogeneity is even considered). This is because the supracontext “3-2” is empty, since none of its corresponding subcontexts (302, 312, 322, 332) are present in the dataset. This is a reduction of 2 supracontexts but it does not really represent a reduction in the number of possible analogical models. This number is, however, cut in half by virtue of the fact that the supracontext “-1-” shows heterogeneous behavior (thus eliminating 9 of the 16 possible subcontexts which could potentially serve as analogical models).

In a small artificial example such as this, testing for homogeneity can be accomplished by a simple inspection of the graph. Skousen’s strict definition of heterogeneity will eliminate any supracontextual node which contains more disagreements than its immediate subcontextual nodes (those situated above it and connected to it by lines), and since the number of disagreements is simply  $2n_e n_r$  (Skousen 1989:29), where  $n_e$  is the number of occurrences labeled “e” and  $n_r$  the number of occurrences labeled “r”, it should not be difficult to identify heterogeneous supracontexts. In fact, testing for homogeneity is usually even easier than that. For instance, a supracontext containing more than one outcome will always be ruled out if one of its occurring subcontexts contains only a single outcome (e.g. -12), since the number of disagreements in such subcontexts must be zero. Likewise, if the product of  $n_x$  and  $n_y$  for a given supracontext is larger than the same product for any of its occurring subcontexts (e.g. 31-), the same will be true, since the number of disagreements is simply this product times two.

The minimal node on the graph, representing the most general (completely unspecified) supracontext, can further be eliminated using what Skousen calls

inclusive heterogeneity. This works exactly the same way as Mitchell's negative candidate elimination: rule out those nodes which are more general than one which has already been ruled out (in this case "-1-"). In practice, this supracontext is almost never homogeneous anyway and will probably always be eliminated, since it also contains more subcontexts than any other supracontext and therefore it entails more calculations while not really contributing much in the way of information content.

#### 4. Neural networks

In some of his earlier work on analogical modeling, Skousen would simply have selected (usually at random) one of the fully specified subcontexts whose supracontexts were not eliminated by his homogeneity constraint, thus making the probability of selecting a given data occurrence linearly proportional to its frequency in the dataset (Skousen 1992:8). Later a conceptually simpler basis for selection was proposed which involved utilizing a network of pointers (which were being used to measure uncertainty in the model anyway).

In this case, the analogical set (from which a model is chosen) is simply the group of pointers in the network originating from and leading to the tokens in the dataset which are contained within each of the non-empty homogeneous supracontexts. Using this as the basis for selection has the effect of making the probability of selecting a given data token proportional to the square of its frequency in the dataset.

Those familiar with connectionist literature will recognize the network in Figure 3 as also representing what is called an auto-associative network. This simple design is the basis for some of the most sophisticated neural networks which have been developed, yet it is actually "the most general architecture for a connectionist system; all other architectures are more restricted subsets of this architecture" (McClelland & Rumelhart 1989:161). Thus it would be possible to implement the AM algorithm using neural networks in the following manner: have one network whose sole purpose it is to determine which supracontexts are non-empty and homogeneous, and then pass this information on through a buffer to a cleanup auto-associator such as the one in Figure 3. Next, for each acceptable supracontext assign an activation strength of +1 to each of its subcontextual nodes and +0 to every other node in the network. Finally, send a single activation pulse into the network and record the resulting activation values. If we assume a Hebbian update rule, then this should, in theory, yield the same results as the standard AM algorithm.

There are two major problems with the connectionist implementation just described, however. For one, it is uninteresting as a connectionist model. By the time

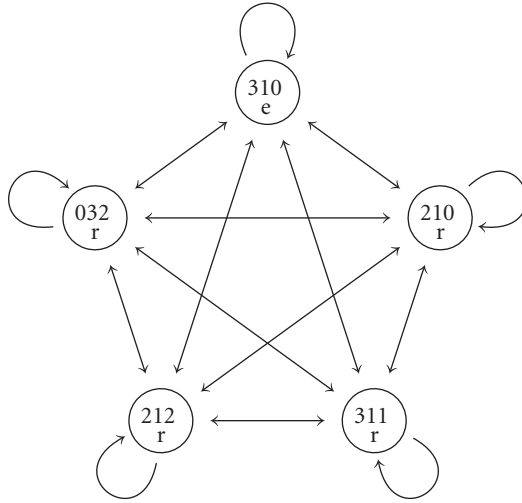
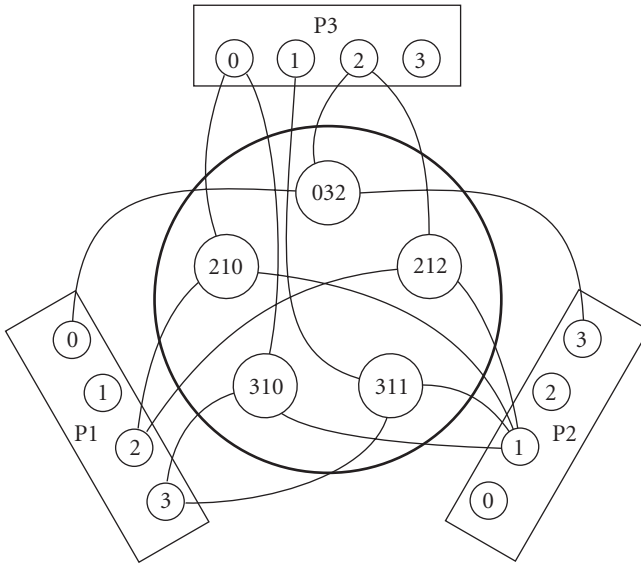


Figure 3. Dataset with network of pointers

we insure that the network will behave exactly like the AM algorithm, it no longer looks much like a typical neural network, since there is practically no learning going on and the information content of the actual weight matrix is minimized. Secondly, this network would not necessarily be any faster than AM, if for no other reason than because the cleanup auto-associator would have to be activated sequentially for each acceptable supracontext and then zeroed before each new activation cycle. This is necessary, since each node in the dataset can be (and often is) associated with more than one supracontext and radically different analogical effects are predicted when these multiple influences are summed.

But is it necessary to bend over backwards to show that a neural network model can implement the AM algorithm? As it turns out the answer is no, since it can be demonstrated that a very simple (and well understood) connectionist model is capable of making analogical predictions which are already similar to those of Skousen’s model. Figure 4 represents an Interactive Activation and Competition (IAC) model designed to illustrate the artificial problem described in the previous section.

The network consists of four subgroups of units. Those subgroups outlined in rectangles contain visible units which are amenable to input from outside the network. These three subgroups represent the three possible positions of the variables in the data cues and each contains four units: one for each possible variant. The subgroup outlined in a circle contains so-called “hidden units” whose activations cannot be directly affected from outside of the network and which represent the individual members of the given dataset. Each unit in every subgroup has neg-



**Figure 4.** An Interactive Activation and Competition Network

active connections with every other unit in the same subgroup, whereas mutually excitatory connections exist between the hidden units and the visible units which correspond to their composition. For instance, the hidden unit “310” will excite the “3” unit in position one, the “1” unit in position two, and the “0” unit in position three – and vice versa. Such models are discussed in great detail in Grossberg 1978 and McClelland & Rumelhart 1989.

An interesting thing happens when the units corresponding to a given context in Skousen’s model are exposed to external input: the resulting stable activation values for the hidden units (those representing the members of the dataset) are virtually equal to the analogical effect that would be predicted for these data members by the AM algorithm without the homogeneity constraint.<sup>4</sup> Recall that Skousen’s analogical set (the pool from which the actual analogical model is selected) consists of all of the pointers originating from and leading to tokens in the analogical network associated with any of the homogeneous supracontexts for a given context. If the homogeneity constraint were lifted, then this set would consist of all of the pointers originating from and pointing to tokens in the analogical network associated with any non-empty supracontext. Nothing would prevent the model from essentially determining contextual similarity and this is what the IAC model does best.

Other similarity-based models, such as TiMBL (Daelemans et al. 1999), also employ contextual similarity (or nearest neighbor) algorithms sometimes in order



to select analogical behavioral models when no model identical to the target is stored in memory. For instance, although the TiMBL model can utilize several different similarity algorithms, its predictions under such circumstances will often closely resemble those of an IAC network or the AM model described above. (See Daelemans et al. 1994 for a comparison of AM and the Lazy Learning model of Aha et al. 1991, which TiMBL is based upon.)

## 5. Analogy without homogeneity

One problem with contextual similarity is that when either random selection or selection by plurality (choosing the behavior associated with the highest number of pointers in the analogical set) is used without the homogeneity constraint, there is no way to guarantee that a member of the dataset will be classified properly if it is presented as the given context, especially if that member exhibits exceptional behavior. For instance, if the context “310e” were presented in the above problem, the number of pointers leading to its node in the analogical set would be greater than that of any other single occurrence in the set, but because of the large amount of leakage the number of pointers leading to occurrences exhibiting regular behavior would be in the majority. Thus, selecting a pointer at random is actually unlikely to result in predicting the appropriate behavior in this case.

In order to rectify this situation, we could adopt a third rule of usage which might be called *selection by majority*: select as a model the single node in the analogical set with the most pointers leading to it. This selection rule is arguably more psychologically plausible than selection by plurality anyway, since it would not have to entail any kind of statistical sampling in order to determine what the most frequent outcome is. When it is applied to analogical sets in a version of AM based on contextual similarity (versus supracontextual homogeneity), the resulting predictions are remarkably close to those of the intact AM algorithm.

In fact, when the predictions of this alternative model were compared to those of Skousen’s original model for the artificial problem illustrated in Figures 2 and 4 above, an average correlation coefficient  $r$  of 0.99 was obtained over the set of 64 possible given contexts, and this result was retained when the identity of the exceptionally behaving occurrence(s) was allowed to vary within the dataset. For 54 of the 64 (84.4%) of the contexts the predictions (in terms of the probability of selecting the exceptional behavior) were identical ( $r = 1.0$ ). The remaining 10 contexts were also assigned very similar predictions by the two models (the standard deviation being only 0.06). Not surprisingly, these numbers were unchanged when the predictions of the IAC network were substituted for those of the alternative version of AM (again assuming selection by majority).

There are also some important advantages to being able to select analogical models without first having to determine homogeneity. First of all, this would eliminate the need for labeling data. Assuming that we do retain individual occurrences of behavior in the brain, it is difficult to imagine how each of these occurrences might be labeled according to an associated outcome, since such labels will always be task-dependent. That is to say, any single form can be associated with a large number of different kinds of behavior depending on a given context. Which behaviors will be remembered and which forgotten, or are all possible outcomes labeled at the time of storage? If the latter is the case, how does the algorithm know which labels to use in determining homogeneity? Until a mechanism is proposed for retrieving behavioral labels and assigning them to appropriate data occurrences, we are left with a serious ambiguity with regard to which behaviors are to be associated with which data and when.

Removing the test for homogeneity and employing selection by majority would also eliminate the need for short-term storage of separate analogical sets and make the number of supracontexts associated with each occurrence in the dataset directly proportional (anticorrelated) to its distance (in terms of variables) from the given context. There would still be significant gang effects and the “correct” analogical model would still retain the highest probability of being selected (relative to other prospective models), but more importantly, such an alternative model would not have the exponential explosion problem which has so severely restricted the application of AM in the past. This is because the number of supracontexts which would have to be considered could be reduced to simply the number of variables ( $m$  instead of  $2^m$ ). In the general case, it can be shown that when determining the analogical set for a given context [a b c], evaluating only the supracontexts [a - -], [- b -] and [- - c] will result in almost no loss of precision over evaluating all of the supracontexts [a b c], [a b -], [a - c], [- b c], [a - -], [- b -], and [- - c]. This may not seem like much of a reduction when there are only three variables, but even with only twenty variables, this would result in a savings of over one million supracontexts.

To demonstrate this, the same artificial problem from Skousen (1989) was presented twice to the alternative AM model without considering homogeneity, once evaluating  $2^m - 1$  supracontexts (all of the possible supracontexts except the most general one) and once evaluating only  $m$  (in this case 3) supracontexts. The results were not identical, but the average correlation coefficient  $r$  was once again 0.99 over the 64 possible given contexts. In summary then, it appears that either an IAC network or a version of Skousen’s AM algorithm which does not test for homogeneity but employs a different selection rule is capable of making predictions which are very similar to those made by the intact AM algorithm (always assuming the random selection rule of usage), but with some important simplifications which may affect the plausibility and usefulness of the general model.

## 6. The SimNet model

Having said all that, I would now like to argue that Skousen's formulation for AM is still superior to the alternative formulation just described. Thus far, when comparing these two formulations, I have been explicitly assuming that one would use random selection as the only rule of usage, much as others have done when comparing AM to alternative models (cf. Baayen 1995). However, as Skousen quite correctly argues (1989:82–85), there is evidence that people can and do learn to make different predictions according to their particular motivation (Messick & Solley 1957), and this can be translated into the ability to apply alternate rules of usage. Without homogeneity, the AM model would be unable to account for this finding, since it could only produce acceptable predictions using a single rule of usage, namely the proposed selection by majority. This predicts behavior which roughly corresponds to standard AM using random selection, but there would be no equivalent for selection by plurality. This applies to the IAC model as well.

Skousen also points out (1989:85–86) that people can also vacillate between forms while speaking. This finding would also be difficult to explain using a model in which the same occurrence (the one with the most pointers in the analogical set) is predetermined to be selected every time (assuming the dataset remains unchanged). In the original model one could simply choose among the various occurrences in the analogical set, which seems satisfying at first glance, but Skousen only describes two rules of usage (random selection and selection by plurality), neither of which can preclude (with any degree of certainty) the selection of the same model twice in a row, unless we assume that forms can be thrown in and out of the dataset (or analogical set) at will. In a connectionist model, on the other hand, one could simply suppress unwanted targets by providing top-down inhibition (or inhibition from the inside out, if you will) and then feed the given context back into the network. Since real-time activation would be measured in milliseconds, this would not preclude speakers from vacillating between targets.

In order to account for both kinds of behavior while still avoiding some of the pitfalls associated with Skousen's homogeneity constraint, I developed a somewhat different neural network model. SimNet is an exemplar-based connectionist model based roughly on a modified version on Grossberg's (1978) IAC model. Unlike many of the more common pattern associator models discussed in the connectionist literature, SimNet employs local representations of actual language (or other behavioral) data in its hidden layer and does not involve any form of extensive training regimen, yet it is still capable of responding to sequences of input and can readily incorporate newly acquired data. As its name implies, SimNet selects analogical models based on their contextual similarity to a given set of input, but unlike many other similarity-based models it can also make probabilistic predictions which appear to employ multiple rules of usage.

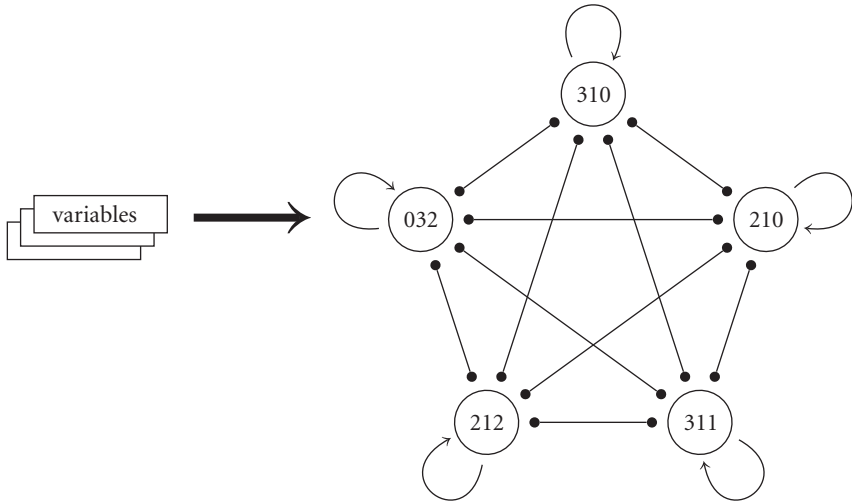


Figure 5. Input to the hidden unit subgroup representing the dataset in SimNet

The model only differs from the IAC model discussed above in that a variation of a simple circuit, also originally proposed by Grossberg (1976) in his detailed discussion of competitive learning mechanisms, is employed on each cycle inside the subgroup of “hidden units” representing the data. This circuit was designed to select a single winner from a pool of partially activated units – essentially the same task presented to someone seeking an analogical target to model behavior after. The basic idea is this: allow every active unit in a given subgroup to excite itself and at the same time inhibit every other unit in the same subgroup.<sup>5</sup> In the limit this circuit will continually drive the activation of the unit(s) with the highest initial activation upwards while driving the activation of all other units in the subgroup down until it reaches its minimum value (selection by majority). But this architecture also yields some interesting results after a single activation pass.

Figure 5 is a graphic representation of the SimNet architecture using the artificial dataset discussed at length above. By convention, lines which terminate in an arrowhead represent excitatory connections while those terminating in a point represent inhibitory connections. The input from the visible units is shown here being presented in sequential order, but because of the particular activation function used, it makes no difference whether the “visible” variable units are activated in sequence or in parallel. Either way, the activation of each hidden unit will be the same after all of the variable input has been received. After the activation inside the hidden unit subgroup has spread, those units whose activation levels are above a certain threshold will then pass along their newly acquired activation, scaled by a measure of attention strength, to all of their associated visible units. Then the whole

process starts over again. Importantly, the set of associated visible units (which represent characteristics of the data) is by no means restricted to those receiving external input. For instance, if the data represent words, then activating phonological variables associated with one word will not only partially activate other words (temporarily) but also the semantic and syntactic characteristics associated with them (and vice versa), in some cases supplying default values for certain kinds of behavior.

## 7. Three types of behavior

The remainder of this chapter will be devoted to comparing the predictions of the SimNet model with those of the AM algorithm on various groups of experiments. The first of these is another artificial problem which was designed to demonstrate the ability of Skousen's model to handle three diverse types of linguistic data: categorical, exceptional/regular and idiosyncratic.<sup>6</sup> This problem also involves data tokens comprised of three variables, except this time the number of possible contexts will be only 32 as the first variable is restricted to the smaller set of variants {0,1}. As before the second and third variables will each have four variants, {2,3,4,5} and {6,7,8,9} respectively. Three distinct datasets will be extracted from this set of possible tokens, each one representing a different type of basic linguistic behavior. To test his model in an environment reflecting categorical behavior, Skousen presented the following dataset:

027x 039x 046x 047x 048x 058x  
126o 137o 147o 148o 157o 159o

In this case, the obvious generalization would be that any token beginning with a "0" should be classified as "x" and any token beginning with a "1" should be classified as "o" (a simple binary rule). Notice that he intentionally stacked the deck against his model by using "x" and "o" data tokens which resemble each other in terms of the remaining two variables. Nonetheless, his model was able to correctly categorize all 32 contexts after the presentation of only nine of the twelve data tokens<sup>7</sup> with a 99.4% degree of accuracy (Skousen 1989:41), assuming that the categorical behavior should be extended to all of the possible contexts.

The second dataset looks very much like the first one, except that only one of the twelve data tokens is labeled "x", representing an instance of exceptional behavior surrounded by regularity:

027o 039o 046o 047x 048o 058o  
126o 137o 147o 148o 157o 159o

A rule-based approach would simply memorize the exceptional behavior and assume that the regular behavior applies elsewhere, but we have seen that this is not an accurate reflection of human categorization (cf. Bybee & Pardo 1981; Johnson & Venezky 1976). Once again, Skousen's model is able to "discover" the exceptional behavior by the time the twelfth data token is presented, but for three of the given contexts close to the exceptional "047" context, there is about a 20% chance that the exceptional behavior will be carried over. Logically speaking, the analogy model is only 98.3% accurate, but realistically it is this fuzziness near the exceptional behavior that lends the most support.

The final environment consisted of a dataset with only two members: 047x and 126o. Here we have two idiosyncratically labeled tokens and precious little in terms of any pattern to model behavior after. Accordingly, we would expect some kind of a smooth transition in predicted behavior moving from contexts close to the first data token to those close to the second (Labov 1973), and this is exactly what happens. The probability of a predicted behavior {x,o} for each context turns out to be a function of that context's variable similarity to the labeled data. Because there are only three variables with which to measure similarity in this example, the transition curve is nearly linear, but as the number of variables is increased, this function begins to look more and more like a threshold (Skousen 1989:49). A strictly rule-based model would have no basis upon which to classify any of the 32 given contexts, save the two in the dataset.

The graphs in Figures 6 through 8 clearly show how close the predictions of the two models (AM and SimNet) are to each other, no matter which type of behavior is presented. Notice particularly that in Figures 6 and 7 all of the leakage occurs in exactly the same environments. The amount of leakage is higher for SimNet (after

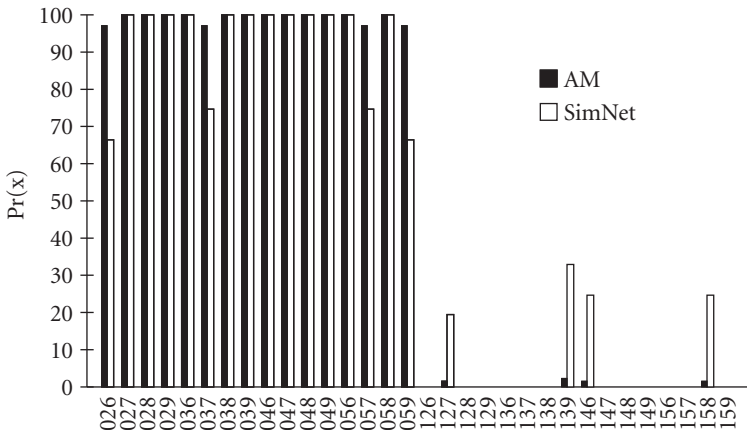


Figure 6. Predictions based on an environment of categorical behavior

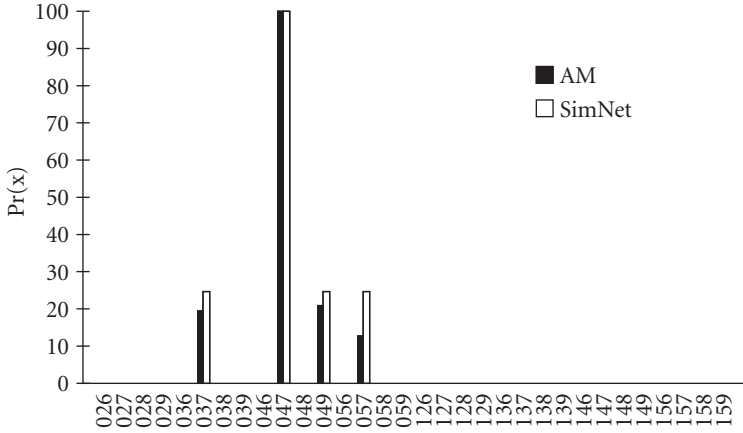


Figure 7. Predictions based on an environment of exceptional behavior

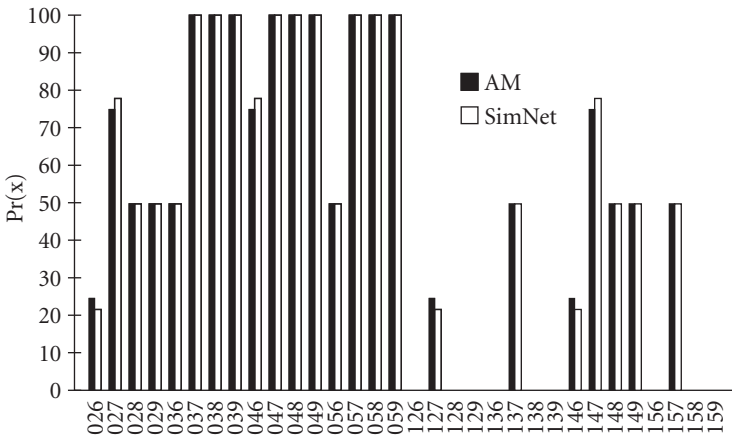


Figure 8. Predictions based on an environment of idiosyncratic behavior

a single activation pass), but may in fact be more realistic, given the relatively small dataset and the fact that each occurrence was only described in terms of three variables. The probability of getting this kind of leakage in a categorical environment will decrease exponentially as the size of the dataset and the number of variables is increased.

The SimNet model also allows for an attention strength bias on each of the variable inputs which was not considered here (all biases were fixed at 1.0). However, there is ample evidence that human beings can and do employ selective attention strategies when attempting to classify information (Shepard, Hovland, &

Jenkins 1961; Nosofsky 1984). For example, in the artificial categorical environment described above, it is not difficult to figure out that the principal consideration for classifying new data should be the value of the first variable. When the attention bias for the first variable is doubled in this environment, there is a significant drop in the amount of leakage across the  $x / o$  boundary (from 6.9% down to 0.5%). In fact, this would allow the predicted behavior to be perfectly segregated according to the value of the first variable at asymptote. The time it takes the network to settle could be offered as a partial explanation for why more mistakes are recorded in timed classification experiments than when people have more time to consider their answers, or for why frequency effects sometimes show up in speeded lexical decision tasks but not when making acceptability judgments (Pinker 1999). Also, although these biases must be set by hand in this simple model, Kruschke (1992) has shown that a more sophisticated neural network employing back propagation of error is capable of learning attention strengths in the course of classification.

What about when the data is noisy or incomplete? Just like AM, the SimNet model is able to “recognize” tokens from the dataset when the first variable has been masked. When the context “#37” was activated, the model behaved exactly as if the context “137” had been activated. Likewise, when the context #47 was activated, the two hidden units representing the contexts “047” and “147” became active and the prediction was a toss between the two. On the other hand, when noisy contexts are presented, the model is forced to lower its affective filter and treat the distortions as if they were idiosyncratic.<sup>8</sup> When the context “037” is presented, for instance, all twelve hidden units become active, the degree of their activation being dependent upon their similarity to the distorted context. The initial predictions (in terms of probability  $Pr$ ) for this context are  $Pr(o)$  55.1% and  $Pr(x)$  44.9% (compared to Skousen’s  $Pr(o)$  53.9% and  $Pr(x)$  46.1%). Due to the lower inhibition levels, the network will have to cycle a long time before eventually selecting the “o” outcome associated with the token “137”. When the similar context “047” is presented, all of the hidden units are again activated, but this time the initial predictions are  $Pr(o)$  50% and  $Pr(x)$  50% and this result will not change as the network settles. Both models yield similar predictions for the “#47” and “047” contexts, but for very different reasons.

## 8. Finnish verbs revisited

Perhaps one of the most well-known applications of the AM algorithm is Skousen’s treatment of past tense verbal morphology in standard Finnish (1989: 101–136). This subject provides an abundance of data for those interested in linguistic variation, and this analysis, more than any other, went a long way toward proving the



utility of his purely analogical approach. I will not attempt to duplicate the detail of this analysis here, but rather simply refer the reader interested in a more in-depth description to Skousen's book.

By way of introduction, past tense verbs in Finnish are generally marked by a word-final /i/, but the segments immediately preceding this final vowel differ widely. Skousen (1989: 102) provides a list of six environments which are more or less amenable to a rule-like description, but there are also many verbs which defy such description. The group used in the study consisted of two-syllable Finnish verbs whose second syllable consists of a consonant followed by a single short unstressed non-high unround vowel (*e*, *ä*, or *a*). In general, this final vowel would simply be replaced by /i/ in the past tense form of these verbs, but there are two somewhat cohesive subgroups within this larger one. For instance, if the final vowel is immediately preceded by a sequence consisting of a sonorant followed by the consonant /t/, then the final consonant-vowel sequence tends to be replaced by /si/. Likewise, if the final vowel is /a/ and the first vowel of the first syllable is an unround vowel, then the final vowel is most often replaced by /oi/ in the past tense form. Sound convoluted? It is – and what's more, there is a considerable amount of overlap between these three outcomes.

The dataset for this analysis was constructed by extracting 173 two-syllable verbs ending in a short unstressed non-high unrounded vowel from the Saukkonen and *Suomen Kuvalehti* textual studies. Any such verb which had at least one past-tense occurrence in either of the databases was included in the dataset. Of the verbs in the dataset, 117 used the *V-i* past tense form, while 36 used *a-oi* and only 20 the *tV-si* form. All of the 173 verbs were then analyzed in terms of ten variables reflecting each verb's phonemic and syllabic identity.

Skousen was limited by implementation considerations to using only ten variables when doing this analysis in the late 1980s. Although the relatively rapid advance of computer technology would allow us to describe this data using twice as many variables today, I used a similar set of ten variables for these simulations in an attempt to keep the comparison as fair as possible.<sup>9</sup> The main point is that these variables are sufficient to distinguish all of the tokens in the respective datasets and, as Baayen (1995: 394) has pointed out, they do not represent an "informed – 'structuralist' – selection of the relevant dimensions of variation".

I will only compare the predictions of the two models for test verbs which were not in the dataset. Most of these verbs did occur in the two textual studies, but since they did not occur in the past tense they were not included among the 173 verbs making up the dataset (Skousen 1989: 114).<sup>10</sup> Using the *Nykysuomen sanakirja* (NSSK) as a reference, Skousen divides these verbs into four subgroups according to their possible past tense forms and their syllable makeup. First, I will compare the predictions for three test verbs in each of the first three subgroups, and then go on to discuss the fourth subgroup in more detail.

For verbs ending in the syllable *-ta* and having at least two vowels in the initial syllable, the first of which is /a/, we find that all three past tense forms are possible:

		Pr( <i>V-i</i> )	Pr( <i>a-oi</i> )	Pr( <i>tV-si</i> )	
kaarta-	'swerve'	0.0	31.4	68.6	AM
		0.0	55.0	45.0	SimNet
saarta-	'surround'	0.1	23.7	76.2	AM
		5.0	53.6	41.4	SimNet
raata-	'toil'	0.0	99.6	0.4	AM
		0.0	92.1	7.9	SimNet

All of the predictions for AM were made using the random selection rule of usage. All of SimNet's predictions were recorded after a single pass through the network.<sup>11</sup> Of course, if the selection by plurality rule of usage were employed in AM, then only one outcome would be selected. The same would be true for SimNet if the network were allowed to settle. In fact, there appears to be a high statistical correlation between the predictions of AM using random selection and those of SimNet after a single activation pass, and another correlation between the predictions of AM using selection by plurality and those of SimNet at asymptote. This is due to the fact that in the large majority of cases, the nearest neighbor selected by SimNet (the token with the highest initial activation) will also exhibit the most frequent behavior. But not always. In this case the two models would make very different predictions for the first two verbs using these alternate selection methods. It turns out that the NSSK lists both the *a-oi* and the *tV-si* past tense forms for these two verbs, but only the *a-oi* form for *raata*. It seems to be advantageous for the models to be able to distinguish those verbs having two vowels plus a sonorant consonant (VVS) in the first syllable from those which do not.

If the onset of the second syllable is not /t/, then the *tV-si* outcome is highly unlikely to be a possibility and we are left with only the *a-oi* and *V-i* past tense forms. Including verbs with a single stressed vowel in the first syllable further decreases the statistical probability that verbs in this subgroup will use anything but the *a-oi* and *V-i* forms:

		Pr( <i>V-i</i> )	Pr( <i>a-oi</i> )	
kalva-	'prey on'	14.6	85.4	AM
		15.2	84.8	SimNet
lappa-	'haul'	2.1	97.9	AM
		9.9	90.1	SimNet
paukka-	'crack'	2.6	97.4	AM
		0.0	100.0	SimNet

Once again, in all three verbs the conditions are met for the general rule which would select the *a-oi* outcome and once again both models are able to predict this behavior without having explicit knowledge of these conditions or the rule itself.

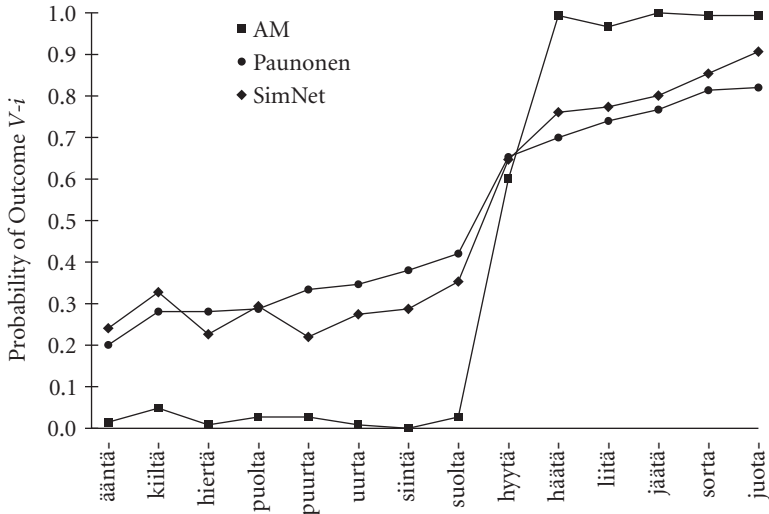
The third subgroup consists entirely of verbs which have only a single stressed vowel in the first syllable and whose final vowel is something other than /a/. Under these conditions, the *V-i* past tense form is highly favored above all others:

		Pr( <i>V-i</i> )	Pr( <i>tV-si</i> )	
kute-	'spawn'	99.8	0.2	AM
		96.4	3.6	SimNet
päte-	'be valid'	100.0	0.0	AM
		100.0	0.0	SimNet
syte-	'chip'	100.0	0.0	AM
		100.0	0.0	SimNet

Note that the probability of selecting the *V-i* form is very high for both models. It is especially unusual for the SimNet model to make such one-sided predictions after only a single activation pass. Skousen (1989: 121) lists gang effects of approximately 4.0 for other verbs in this subgroup, and Paunonen (1973: 291) suggests that alternate past tense forms for these verbs are all but impossible, "even as slips of the tongue", whereas alternate past tense forms for other verbs did not seem so abhorrent. This makes sense if verbs associated with alternate past tense forms never receive any significant degree of activation, and it would follow then that where such slips of the tongue are possible we should also expect higher levels of leakage during early stages of activation spread.

The fourth subgroup is basically a miscellaneous category. It consists of verbs which have heavier first syllables, containing any combination of two or three sonorants (vowel, liquid or nasal), the first of which was not /a/. Some of them had obstruent codas and some did not. This subgroup always has a final syllable which has as its onset the consonant /t/ plus any short unstressed non-high unrounded vowel. Verbs in this subgroup can take only the *V-i* and the *tV-si* past tense forms.

Paunonen (1973) describes an experiment in which a list of forty-one of these verbs was presented to a group of thirty-six university students. The task was to decide for each verb whether the two alternative past tense forms (*V-i* and *tV-si*) were (1) neither natural nor possible, (2) not very natural but possible or (3) completely natural and possible. In an effort to account for the intuitions reflected in Paunonen's experiment, Skousen (1989: 118) compared the predictions of his model for the 14 verbs from this subgroup which did not occur in the dataset with the results obtained by Paunonen. This comparison is shown graphically in Figure 9. The estimated linear correlation between Skousen's results and Paunonen's



**Figure 9.** Comparison of the predictions of AM and SimNet with the acceptability ratings obtained in Paunonen (1973) for 14 Finnish verbs

composite measure of acceptability was an impressive 0.97, but I believe that the flexibility of the SimNet model may allow it to account for these intuitions even better.

The estimated linear correlation between the results obtained using the SimNet model and the acceptability ratings from Paunonen's experiment was 0.98. This same correlation coefficient was obtained when comparing the SimNet results with those obtained using AM but, as Figure 9 clearly demonstrates, this is not always the best measure of similarity between two sets of results. The predictions obtained using the SimNet model are visibly closer to Paunonen's results<sup>12</sup> (linear matched for comparison), even though its correlation coefficient is only one point higher. Likewise, although the SimNet predictions tend to respond to similarity relations in the data in roughly the same way as AM, they are clearly not as close to Skousen's results despite the fact that the correlation coefficient is identical for both sets.

Remember that the SimNet results reported here were recorded after only a single activation pass through the network. In the limit the predictions of this same model would approximate a threshold function with a sharp (but not vertical) transition in the behavior curve between the verbs "suolta-" and "häätä-" and no leakage on either side of this transition. In other words, the predictions recorded after the network is allowed to settle are much closer to those obtained using AM. This is true regardless of whether the rule of usage employed is random selection or selection by plurality.

This is interesting for a number of reasons. For one, it implies that AM's predictions using random selection are too close to being categorical to account for the large amount of variation observed in the acceptability ratings of Finnish speakers. More importantly, it suggests a possible explanation for Paunonen's own judgment that the disfavored past tense forms of these verbs might slip out accidentally, whereas the disfavored forms for the verbs in the third subgroup could not. It seems at least possible that in rapid speech there would be less time spent on deciding which past tense form to use for verbs which have not been committed to memory. The SimNet model would in fact predict that the more time a person spends making this decision (allowing this decision to take place), the less chance there will be that a disfavored form will be vocalized. Paunonen's judgments and the ratings observed in his study both indicate a difference in the proportional strength of the predictions for these two subgroups, but this is missing in Skousen's model.

## 9. Variation in Danish compounds

Predicting the form of compounds in Germanic languages has historically been perceived to be a very difficult task (cf. Krott et al., in this volume), but I would agree with Becker (1992) that this is largely a consequence of the fact that linguists have confined themselves to a syntagmatic approach to compositional morphology, rather than allowing for new formations to be derived analogically from existing compounds. The experiments described in this section were designed to show that the latter approach can not only account for the attested formations better than one based on concatenation with inflection or allomorphy, but it is also capable of accounting for the variation which does occur in a straightforward manner.

Most Danish words have what is called a standard combination form which is used when the word occurs as the first element of a compound. Augst (1975:134) put the number at approximately 90% for German, and based on my own limited research, this seems to be about right for Danish as well. The problem is that there is no systematic way to relate these forms to other inflected forms of the same words, and even where systems of rules and subrules for the formation of these combination forms have been suggested (cf. Hansen 1967; Kōneke 1986), there are always large numbers of exceptions. This can also not be considered a peripheral phenomenon, since the number of new (recorded) compound formations in Danish over a twenty year period was shown to be more than twice as high as all other new word formations put together (Riber Petersen 1984).

While previous studies have attempted to describe the synchronic distribution of these combination forms based solely on phonological and/or morphological information about the first constituent, a number of factors could be proposed which may influence them:

1. the semantics of the first constituent
2. the semantics of the entire compound
3. the syntactic function of both constituents
4. the structure of the last syllable in the first constituent
5. the structure of the first syllable in the head constituent
6. the phonological structure of both constituents
7. the presence of a derivational suffix on the first constituent
8. the inflectional class of the first constituent
9. the position of stress in and the length of the first constituent:

<b>dag-vagt</b> ‘day shift’	vs.	<b>hverdags-tøj</b> ‘everyday clothes’
<b>fart-grænse</b> ‘speed limit’	vs.	<b>overfarts-sted</b> ‘ferry’
<b>gang-bro</b> ‘footbridge’	vs.	<b>foregangs-mand</b> ‘pioneer’
<b>gård-mand</b> ‘farmer’	vs.	<b>urtegårds-mand</b> ‘gardener’
<b>snit-mønster</b> ‘pattern’	vs.	<b>gennemsnits-X</b> ‘average X’
<b>tøj-klemme</b> ‘clothes pin’	vs.	<b>legetøjs-butik</b> ‘toy store’

An ideal representation (or list of constraints in Optimality Theory) would take each of these factors into consideration. While hardware limitations and a lack of fully specified databases still prevented me from doing this, it was nonetheless possible to find a representation scheme which addressed many of these possible influences.

Following other work in connectionist modeling (cf. MacWhinney & Leinbach 1991; Daugherty & Seidenberg 1994), I adopted a symmetrical CCCVCCC syllabic representation for the five syllables closest to the constituent boundary of each compound<sup>13</sup> as indicated in Figure 10. The total number of variable inputs was thus 35. However, these representations differ from those of previous researchers in a number of important ways; each merit some discussion, since they provide a useful and easily implemented general measure of edit distance for modeling various linguistic phenomena in other languages as well.

First of all, as can be seen in the figure, the syllables carrying a main stress are fixed to the initial slot of each constituent’s subrepresentation. This has the effect of anchoring the entire compound according to its metrical structure (see Gentner & Markman 1997 for a discussion of the importance of alignment in measuring sim-

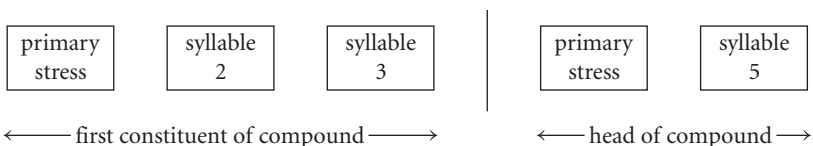


Figure 10. Overview of variable representation for compound experiments

ilarity, and Stemberger 1990 for other psycholinguistic evidence concerning the storage of stress patterns in words). This type of alignment strategy is not only useful for representing Danish compounds. It also provides a simple explanation for why the much-discussed denominal verbs derived from existing English compounds like *grandstand* and *highstick* are not inflected like the simplex verbs *stand* and *stick* in the past tense (*\*grandstood*, *\*highstuck*): if these stressed monosyllabic verbs are not mapped onto their unstressed counterparts in the compound, then it is unlikely that they will provide a useful analogical model.

I also used orthographic (rather than phonemic) features, partially for logistical reasons and partially to rule out effects of dialect. However, these features were distributed within each syllable according to their level of sonority rather than simply being left or right aligned. Thus, the first consonant position was reserved for sibilants, the second for stops and fricatives, the third for liquids, nasals and semivowels, etc. This further alignment of features allowed the model to make more fine-tuned similarity judgments in the absence of unlimited variables, especially when blank feature slots were considered to be meaningful.<sup>14</sup> In any event, this 35 dimensional space was large enough to provide unique feature vectors for each compound.

The data for these simulations were extracted from three sources: the electronic version of the Nudansk Ordbog (1986) and two Danish-English dictionaries (Vinterberg & Bodelsen 1966 and Vinterberg & Axelsen 1979). In all, three datasets were extracted for each compound tested: one containing all compounds with an identical head constituent, one containing all compounds with a matching first constituent, and a third which was a combination of the first two sets. Each dataset was then further screened to eliminate any homophony or metaphorical extensions of the target constituent which would probably not be activated when the target is viewed or when searching for an analogical model. For example, the Danish word *kort* can mean both “map” and “card”, the word *klippe* can mean both “to cut” and “rock”, whereas *jorde* can be the first constituent for compounds concerning “funerals”, “the earth”, and by extension “soil or property”. In larger datasets these might be considered noise, but in the present case, their inclusion would have been detrimental, since semantics can affect a compound’s morphology. Finally, following Booij 1994 and Baayen, Dijkstra, & Schreuder 1997, plural constituents were considered to have their own (i.e. opaque) representations, since pluralization can (at least in some cases) involve concept formation (as opposed to verbal inflection), and evidence from psycholinguistic studies indicates that plurals also have unique lemma representations.

The first set of targets was taken from the book *Babettes Gæstebud* (‘Babette’s Feast’) by Karen Blixen. 245 compounds were extracted from the book, all of which had a noun head constituent and a first constituent which was listed in the Nudansk Ordbog as an independent word. It would be difficult to be more specific than

this when using Danish data, due to the high degree of overlap between infinitive, substantive, and adjectival first constituents. The linking options themselves (+zero, +s, +(C)e and -(C)e) can further complicate matters, for example, by creating forms which resemble an infinitive but are not listed independently (bøn<sub>N</sub> ‘prayer’ → bønne-bog ‘prayer book’) or by truncating intuitively verbal forms so that they are indistinguishable from existing substantives (krybe<sub>V</sub> ‘creep/crawl’ → kryb-dyr ‘reptile’). This means that adverbs were also admitted as first constituents (comprising only 13.9% of the targets), all of which have 100% static combination forms. Nonetheless there was quite a bit of variation: 10% of first constituents in this group showed nondeterministic behavior with regard to their combination form. Wherever possible, the representations used were based on listed words and those which semantically made the compound easiest to interpret.

First, I did a cross validation test for the 245 head-match-only datasets.<sup>15</sup> These results were already quite good: 79.1% correct for AM and 83.9% correct for SimNet. Both models generally chose analogical models which had the same metrical structure and number of syllables. They also recognized derivational suffixes and were able to pick up on the generalization, noted in the literature, that the suffixes *-dom*, *-else*, *-hed*, *-ing*, *-ion*, *-skap* and *-tet* all have combination forms which use the linking element +s (99% correct for compounds with these suffixes or the suffix *-er*). More than that, however, they were able to distinguish between bisyllabic words ending in *-ng* (which normally take the +s linking element) and those ending in *-ig* (which often do not take a linking element). First constituents which ended in a vowel were very unlikely to add anything in their combination form (never when the final vowel was stressed, yet always after *-else*), and in general, monosyllabic constituents were less likely to add anything in their combination forms as well.

In order to account for the observation (Becker 1992) that the semantics of the entire compound can influence the the form of the first constituent, I then attempted to split the five head-match-only datasets which had the highest number of first constituents up into smaller ad hoc classes based upon some salient semantic notion which was observed to bind them together somehow. These five datasets had the head constituents *-bog* ‘book’, *-dag* ‘day’, *-mand* ‘man’, *-mester* ‘master’ and *-tid* ‘time’. The average size of these datasets before splitting them up was about 145 words. The average size of the ad hoc datasets was about 20.

For example, it was observed that the percentage of correct predictions for the “man”-compounds (76.7% with AM originally) were much higher when those compounds referring to naval posts (90%) or to violent/criminal people (100%) were considered separately. The “day”-compounds (originally 63.6% with AM) were likewise very consistent when those days referring to various types of weather (100%) or a typical day in a given month (100%) were considered separately. Similarly, “day”-compounds denoting a celebration, holiday or commemoration, many



of which were incorrectly classified when lost in the crowd, were correctly predicted 93.5% of the time by both models in isolation.<sup>16</sup> These ad hoc categories were also found to exist for a good many other Danish and German compounds. They are not necessarily disjunctive and could be as specific as “sports titles” or as vague as “animals”, but it seems that if no previous compound with the same first constituent exists to model behavior after, then this kind of semantic information must be crucial for selecting the correct combination form for the novel compound. The average percentage of correct predictions for this group was 93.1% for AM and 93.6% for SimNet.

Next, I tested the actual compounds which were extracted from the book using their combined datasets (all listed compounds with either a matching head or first constituent). On the first run, I allowed both models to remember all of the data and, not surprisingly, both did a very good job of predicting the combination form for each first constituent (99.2% for AM and 99.6% for SimNet). While this may in fact reflect something close to what a native speaker of Danish would do under identical circumstances, it is nonetheless not very interesting from a modeling point of view. This percentage is a direct result of the fact that 78% of the target compounds were listed in the sources from which the databases were extracted. Thus 191 of the targets were predestined to be correctly predicted in both models, and these same data tokens also severely restricted the predictions for the remaining targets as well.

Therefore, during the second run it was assumed that the target compound was not available as a possible analogical model. This time the predictions were more “interesting”, but also still very impressive. On the second run, AM correctly predicted the form for 95.9% of the target compounds, while SimNet predicted 97.5% of the forms correctly. The differences between the two models are instructive in this case.

Both models missed on the compounds *klipfisk*, *øllebrød*, *aftensmåltid*, *landsby* and *provstedatter*. *Klipfisk*, as it turns out, was probably borrowed directly from the Norwegian word *klepfisk* and has nothing to do with the verb *klippe* ‘to cut’ at all, even though this is an almost automatic interpretation for the compound (which means “split and dried codfish”). The word *øllebrød*, on the other hand, does not mean “beer bread” as a non-native speaker would be inclined to believe, but rather a soup-like dish made with (usually non-alcoholic) beer and pieces of bread. Its most likely etymology is *øl+og+brød* (‘beer and bread’). This is a semi-productive pattern in Modern Scandinavian and includes other words like *smørrebrød* ‘butter and bread’ and *saftevand* ‘juice and water’. Both *aftensmåltid* ‘evening meal’ and *landsby* ‘rural town’ are examples of an exceptional linking element being retained through high frequency while surrounded by different behavior. The normal combination form for *aften* is +zero, except when referring to food (see above), while the word *land* is used almost exclusively without a linking element

when it means “rural”. In fact, the +s linking element is strongly associated with the meaning “national” in Danish. All four of these words are very common in spoken Danish.

The frequency argument will not work for the next compound, *provstedatter* ‘rural dean’s daughter’, however. This appears to be the expected form for this compound formation, but it is not predicted by either model. This linking element could very well be an epenthetical schwa between the hard consonant cluster at the end of the first constituent and the stop onset of the head constituent which matches it in place and manner of articulation. As will be discussed below, this compound would have been predicted correctly if it had had more support from other compounds sharing this first constituent or if only first constituent matches had been used in the dataset.

Of the few compounds that SimNet classified “better” than AM, the first three are remnants of completely unproductive linking patterns. They would not have been selected had activation been stopped early on during processing, and quite simply they receive the correct combination form at asymptote only because their nearest neighbor in the dataset happens to use the same combination form: *bygmester* < *brygmester*, *spørgsmål* < *søgsmaal*, *bønnebog* < *børnebog*. In the case of the first two of these compounds, they also happen to be remnants of the same unproductive linking pattern as the target.

Another two which SimNet predicted correctly were *Norgeskort* ‘map of Norway’ and *jordelivet* ‘the earthly life’. Both words are part of small but productive patterns which received very little support in their respective datasets. Maps of countries always take the +s linking element, but such compounds only accounted for 2 out of 17 tokens in that dataset. The word *jord*, when used metaphorically or when it can be translated with “earthly” (vs. “earth”), consistently takes the +(C)e linking element, but this time only 3 of 74 data tokens supported this (sub)pattern. Again, it is a coincidence that the nearest neighbors (*verdenskort* and *jorderige*, respectively) are also a part of these ad hoc groupings. In other words, it is an open question whether or not we would want a model of language to predict these particular forms correctly under these circumstances. SimNet took a long time (relatively speaking) to predict both of these forms and it also missed another target at asymptote, *måltid*, which despite its age should have been easy to predict from its phonology/orthography. If selection had been made after a single activation pass, there would have been a good chance that a correct model would have been chosen, in fact.

Intuitively, we would expect the first constituent by itself to be a more reliable indicator of what the combination form of new compounds should look like, so the same group of 245 targets were once again tested, this time using the first-constituent-match-only datasets. All else being equal, AM was able to correctly predict 97.1% of the targets this time. This is only a 1.2% increase, but a Pear-

son's chi-squared test performed on these results indicates that it is a significant one ( $\chi^2(1) = 12.64$ ,  $p = 0.0004$ ). Unfortunately, SimNet's predictions were no better using these datasets than they were using the combined datasets (97.5% for both). This is because 7 of the 245 targets (including some which were predicted correctly) could not be tested in this way, since there was not a single listed compound with the same first constituent in the sources used. SimNet also lost 2 other correct predictions when the head-match tokens were removed from the datasets (for *bygmester* 'master builder' and *spørgsmål* 'question'). In other words, under this condition the two models functioned almost identically.

I would like to mention just a few examples of more rampant variation involving Danish compounds now before moving on to the psycholinguistic experiments. Recall that about 10% of Danish words are not consistent in their combination forms. In fact, many of them are attested with more than one form within a single compound type:

*svingdør* / *svingedør* 'swinging or revolving door'  
*dødedag* / *dødsdag* 'date of someone's death'  
*værtedyr* / *værtsdyr* 'host animal'

Most often such pairs are functional equivalents, but sometimes they are not:

*landmand* 'farmer' vs. *landsmand* 'compatriot'  
*natravn* 'nighthawk' vs. *natteravn* 'night owl' (person)

When there is a semantic distinction, then it is possible for ad hoc semantic groupings to explain the variation (as in the above examples). Logically, if no such semantic difference is present, it should be more difficult to explain this variation in this manner. To test this assumption, I looked at 154 compounds whose first constituent was *nat* 'night'. 50 of these (about 1/3) are listed with the combination form *natte-* and the rest with the form *nat-* (none with the form *nats-*). As expected, many of these compounds can be grouped into ad hoc semantic categories which turn out to be quite regular internally. This was attempted for all 154 compounds in this set; however only 77 of them (50%) were given categorical predictions by AM and SimNet, thus indicating that we should indeed expect some overlap. In fact, I found three "night"-compounds listed with both combination forms and no apparent difference in meaning. The predictions for those three compounds is given below:

compound		AM predictions	SimNet predictions
<i>nat(te)kvarter</i>	'accommodations'	55.1% / 44.9%	55.4% / 42.0%
<i>nat(te)himmel</i>	'night sky'	56.6% / 43.4%	55.8% / 41.1%
<i>nat(te)runde</i>	'night beat'	40.9% / 59.1%	37.4% / 62.6%

Another kind of variation involves deverbal compounds in Danish. There is a generally accepted rule that deverbal first constituents with simplex morphology should use the +(C)*e* (infinitive) combination form, unless a nonconcatenative substantivized form exists, in which case the latter form should be used. If a deverbal first constituent shows complex morphology, on the other hand, then a (longer) substantivized combination form should be used (cf. Køneke 1986). In other words, a deverbal first constituent should look as much like a noun as possible and these are the accepted ways of accomplishing this.

Of course, there are many exceptions to both of these generalizations:

*be-skære-saks, op-vaske-maskine, und-vige-manøvre* (complex +(C)*e*)  
*bygning-måde, åbnings-fest, skærings-dag* (simplex +(C)*ings*)

We also find examples of first constituents following both of these combination patterns with no apparent rhyme or reason, but just as in the previous examples, this variation is nontrivial and cannot be attributed to dialect.

One compound family with a particularly high amount of this kind of variation has a first constituent based on the verb *bygge* ‘to build’. There were 124 such compounds listed in my sources, approximately half of which occur with the combination form *bygge-* and the other half with *bygning-*.<sup>17</sup> There were also three remnant forms with the unproductive *byg-*. There were also 18 of these compounds listed with nondeterministic behavior. Two of these are regularizations of the unproductive remnant forms, while the remaining 16 are attested with both the +(C)*e* and the +(C)*ings* forms. Despite this rampant variation, both AM and SimNet were able to predict the correct behavior (or at least one of them) for each of the 124 compounds in this simulation. Given the above discussion, this is not so surprising. More interesting are the average results for those 16 compounds listed with both a +(C)*e* and a +(C)*ings* combination form by AM (using random selection) and SimNet (after a single activation pass):

AM (random selection)	SimNet (single pass)
-nings 49.7%	-nings 49.9%
-ge 46.7%	-ge 45.7%
+zero 03.6%	+zero 04.4%

While rule-based approaches to language explanation can sometimes predict conflicting outcomes, I do not believe that they can provide quantitative results like this or predict the amount of variation which is actually attested in the data to this extent. The simulations using the compounds extracted from the book together with these results show quite clearly the degree to which very different models of analogy can approximate each other’s predictions, even using alternate rules of usage.

## 10. Human classification strategies

Some of the most commonly encountered criticisms of language models based on pure analogy are that they are not restrictive enough or not reliable enough as classifiers in a statistical sense. Thus, from a language evolution point of view they should also predict rates of change across languages which are simply not attested. Certainly the research reported in this volume and elsewhere involving AM should go a long way towards dispelling the myth that analogy cannot be restrictive. Daelemans et al. (1999) have also put forth solid evidence that retention of specific examples and avoidance of generalization will lead to more realistic results when modeling language learning phenomena. In general, I believe that such criticism also betrays an underestimation of the role of convention in language (and perhaps in human cognition), which is also common in linguistics.

We can easily test quantitative analogical models to see how they classify various data, but how do human beings go about classifying data to which they have not yet been exposed. In an effort to help answer this question, I conducted two simple psycholinguistic experiments which were designed to test competing hypotheses about this important process.

### Experiment 1: Numeric data

#### *Materials*

The same set of 64 three-digit numbers used in Skousen 1992 and Baayen 1995 was used as the data for this experiment. These numbers each consisted of three variables which could take on the values 0, 1, 2 or 3. Those whose second variable had a value of 0 or 1 were assigned the outcome A; those with 2 or 3 were assigned the outcome B. A group of 16 of these 64 labeled numbers was extracted and used as a presentation set. The remaining 48 numbers were then randomly divided into 3 groups of 16 test sets. Half of the presentation sets were presented in numerical order and the other half were presented to a control group in random order. A list of these presentation and test sets can be found in the Appendix.

#### *Procedures*

The participants in the experiment were asked to perform a simple labeling task. Each participant was given a sheet of paper which contained one of the two labeled presentation sets at the top and one of the three test sets at the bottom. There were underscores next to each of the numbers in the test sets and participants were asked to study the labeled numbers at the top of the paper and then label the numbers at the bottom of the paper based on what they had seen. These instructions were

given in written form and verbally at the beginning of the experiment, which lasted approximately five minutes from the time I finished giving the instructions.

### *Participants*

Forty-six volunteers participated in this experiment. All were undergraduate students at Utah State University.

### *Results*

The group which was shown the presentation data in numeric order and the control group both classified the unmarked tokens equally accurately, the adjusted average misclassification rate for both groups being 19.3 percent.<sup>18</sup> This rate of misclassification is certainly much better than chance, but it is also a far cry from the perfect performance of many models which are claimed to be more optimal in terms of modeling human classification. The analysis of the responses indicated that only about 20% of the students were able to discover the relatively simple rule used when originally classifying the presentation set. Many of them were clearly using analogy based on individual data tokens in the presentation set and not looking for a rule to apply at all. This analysis was verified by informal questioning after the task was completed during which some participants reported spending the bulk of their time looking for similarities between the items in the presentation set and those in the test set (which is the last thing a rule induction engine would do).

## Experiment 2: Language data

### *Materials*

For this experiment another set of 16 data tokens was used for presentation, but this time they consisted of actual English words. Instead of numbers, the three variables consisted of graphemes: the first and third variables were all orthographic variants of the four English voiceless stops /ʃ/, /p/, /t/ and /k/,<sup>19</sup> whereas the second variable was allowed to take on any of the four English vowels “i”, “a”, “o” and “u”. Thus the dataset was comprised of closed one-syllable English words each containing a combination of two voiceless stops. Words containing the vowels “i” and “a” (unrounded front vowels) were assigned the outcome 1; those containing “o” or “u” (mid or back vowels) were assigned the outcome 2. Each vowel was directly correlated with one of the numbers (0, 1, 2, 3) from the first experiment and their numbers within the dataset were kept constant as well in an attempt to minimize the effect of the make-up of the dataset on the resulting predictions. The emphasis was on using numeric data versus actual language data. Two test sets were used. The first (T1) consisted of a second group of 16 closed one-syllable English words similar to those in the presentation set. The second test set (T2) differed from the

first in that voiced stop consonants were allowed in the first position. Half of the presentation sets were presented in alphabetical order and the other half were again presented to a control group in random order. A list of these presentation and test sets can be found in the Appendix.

### *Procedures*

The procedures for this experiment were identical to those of the first experiment.

### *Participants*

Thirty-eight volunteers participated in this experiment. All were undergraduate students at Utah State University. All were native speakers of English.

### *Results*

As in the first experiment, the control group classified the test items equally accurately, but this time the adjusted average misclassification rate was about one quarter of that obtained for the numeric data, namely 5.9% for the T1 condition and 4.5% for the T2 condition. The presentation and test sets (especially T2) were constructed in such a way that the participants were likely to concentrate on the rime of each syllable and ignore the onset – a very common task for most speakers of English, yet an analysis of the responses indicates that half of them still did not pick up on the categorical rule used to classify the original data. Interestingly, almost half of the misclassifications were due to the four words in the two test sets which did not have a rhyme in their dataset: “pip”, “tip”, “bock” and “dock”. On average, these words were three times more likely to be misclassified than the average word containing a rhyme in its dataset. This is a strong indicator that once again analogy to individual items was being used as opposed to the categorical vowel rule.

### General discussion

It makes sense that human subjects would perform better on the linguistic labeling task than on the numeric labeling task. First of all, we have more practice analyzing words than random strings of numbers. The words also have the additional advantage that they can be associated with sounds which allowed the subjects to compare the presented data on a dimension which was not available during the numeric labeling task. Finally, the task was conceptually simpler as well, since there was an additional clue as to which variable was likely to be the salient one for the purposes of this particular classification task: two types of variables (consonants and vowels) were used in the second experiment, whereas only one type (numbers) was used in the first one. That the results were slightly better for the T2 condition compared to the T1 condition also follows from the fact that there were no exact matches in the

T2 test set for the initial consonant, which was yet another clue that the information needed for classification was to be found somewhere in the rime. Yet after all of these clues, a full half of the participants were still unable to perfectly classify the test items according to the proposed rule.

How about analogical models like AM and SimNet? It turns out that these types of models can very closely approximate the behavior of the subjects in these two experiments. These same data were presented to both models and their respective predictions are given below:

	Misclassification rates		
	AM	SimNet	Human data
Numeric data	9.0	13.1	19.3
Language data 1	6.0	5.7	5.9
Language data 2	4.7	4.7	4.5
Combined 1 & 2	5.35	5.2	5.2

In this case, we know that the models are basing their predictions on pure analogy to the presented data. It is interesting to note that both models not only show similar overall rates of misclassification (especially for the linguistic tasks), but they also stumble in similar places. For the T2 test set, both models misclassified only those items which did not contain a rhyme in the presentation set. Recall that the participants were also three times more likely to misclassify these same items, relative to other items in this test set.

## 11. Conclusion

In this paper we have examined several different types of models based on analogy which all appear to be related to each other to various degrees. The second half of the paper was devoted to an in-depth comparison of two specific models, AM and the localist connectionist model, SimNet. We have seen that both models are capable of predicting language behavior in a way which seems to be very consistent with the available evidence from actual language users. SimNet is particularly good at capitalizing on similarities in a given population of exemplars. It has the advantage of being able to store information about frequency in terms of weights and associations and can also retrieve default characteristics of activated hidden units which are not present in the input. We have also discussed how the settling of activation in this model closely approximates the different rules of usage posited for AM.



On the other hand, AM is clearly better than other exemplar-based models at learning to predict probabilistic behavior at its attested frequency of occurrence (Skousen 1989:81). The other models discussed can approximate these predictions but are rarely as accurate in terms of mirroring the actual level of variation to which they have been exposed. Even SimNet and TiMBL, which make predictions which are very close to those of AM, must make use of information weight gain measures or adjust some parameters in order to account for idiosyncratic or otherwise deviant data to the same extent. It may be possible to systematically set such parameters based on the size and make-up of a given dataset, but this would nevertheless not be as conceptually attractive as not having to adjust any parameters at all. Another apparent advantage of AM is that its predictions are less drastically affected by imperfect memory than those of other similarity-based models.

Some choose to view these various analogical models as competitors, but I prefer to think of them as being complementary to each other. Becker's (1990) analogy model could be viewed as a bridge between formal transformational grammars and the more functionally oriented morphological/phonological models of Ohala (1974) and Bybee (1985), both of which also share many characteristics with Skousen's model. In a similar way, I think AM (and TiMBL and SimNet) could be viewed as a bridge between more logically oriented models like that of Mitchell (1982) and more elaborate distributed connectionist models. Neither one of these model types resembles the other very much, but both have many important features in common with the AM model. In a sense, this allows the latter to enjoy the best of both worlds: better interpretability plus the ability to make useful generalizations on novel input and, as we have seen, to make human-like classifications, especially in the domain of language.

## Notes

1. Mitchell's 1978 thesis essentially addressed one of the major shortcomings of perhaps the first viable concept-formation program (Winston 1975), namely that it only maintained one nondisjunctive hypothesis at any given time (Genesereth & Nilsson 1987:174), and many of the subsequent concept-formation models can be shown to be special cases of Mitchell's original model (Mitchell 1982).
2. This formula insures that each concept in the graph will be a generalization of the concepts which are both located above and connected to it. In logic this relationship is referred to as a partial order of generalizations.
3. Of course nasals and stops are not mutually exclusive, but for the sake of this example, we will pretend that they are. This pretense will have no impact on the theoretical import of the example whatsoever.

4. All 64 of the possible contexts in the example were presented to both models and the probability (in percent) of selecting the exceptional outcome (e) was determined for each. The average correlation coefficient  $r$  between the two sets of results is equal to 0.99, but only when the general context (i.e. “- -”) is left out in Skousen’s model. This should not be a problem though, since this supracontext is trivially non-empty and does not reveal anything about the similarity of the data tokens to the given stimulus. This supracontext would almost always be heterogeneous anyway in the standard model.
5. In my formulation, every hidden unit in the network will excite itself and inhibit every other hidden unit, with the excitatory force from each hidden unit being directly related to that unit’s net positive input and the inhibition from each hidden unit relative to a fraction of that value.
6. These experiments were originally reported in Skousen 1989:40–49.
7. Skousen uses a strict (and consistent) definition of behavior discovery throughout his 1989 book: a behavior is considered to be discovered when the overall leakage (probability of predicting the wrong outcome) permanently drops below 2%. For practical purposes, however, his model will discover categorical behavior as soon as an equal number of differently labeled tokens are presented.
8. This involves lowering one parameter in the model, the inhibition level within the hidden unit subgroup, to one third of its original value. It is a virtue of AM that no such adjustments are ever necessary to get similar results, but they do not seem to be completely unmotivated. It is also important to note that this is also the only parameter adjustment which was ever necessary during the course of the reported experiments.
9. These variables are almost identical to the ones used by Skousen in his analysis, except that instead of counting the sonorants in the first syllable he specified whether or not there was a second “vowel” in the first syllable and whether or not there was a liquid or nasal anywhere after the first vowel. After reviewing more extensive lists of Finnish verbs, it appears as though the information most relevant to the palatalization of the onset of the final syllable (when this syllable is open, has a /t/ onset and a short unstressed non-high unround vowel) is the relative length of the word in terms of the number of sonorants, but this was not represented in the original description. All such two-syllable Finnish verbs with three sonorants in the first syllable show a preference for palatalization in the past tense, and interestingly, this preference also holds for most three-syllable Finnish verbs having similar final syllables, where the sonorants are spread out over two preceding syllables. For most verbs these variables yield the same results as those used in Skousen’s original analysis and those shifts in predicted probabilities which do occur are relatively minor.
10. If the test verbs were part of the dataset, then the predictions would be dictated by the outcomes associated with these verbs and thus would be uninteresting, assuming perfect memory. Although this was not tested specifically, it would also be possible to run these simulations under conditions of imperfect memory and exclusion of the given context, even if that context were part of the dataset.
11. For all of the tests conducted using the Finnish data, the same parameter settings were used in the model as before, except that the gamma value was set slightly higher at 1.65 $\alpha$ .

12. The results of Paunonen's experiment are adapted in this figure assuming a linear relationship between his acceptability measure and probabilities. It is also possible that this functional relationship could be non-linear. It would be interesting to explore this possibility and also to see how the predictions of both models would be affected by using imperfect memory.
13. Actually, I only used two consonant positions in the coda, since there were only 3 instances in the Danish data which required the use of all three positions. The 7th feature of each syllabic representation was used to give information about the number of syllables preceding (or following) a given syllable.
14. This is done in AM by setting the "null" parameter to "include". When empty features were not considered meaningful, then in general only 10–20 dimensions were evaluated and the algorithm was much faster but the predictions were not as tight.
15. Because the heads were the same within each dataset, only the 21 variables defining the first constituent of each compound were used in these simulations. The extra variables would be ignored in both models anyway.
16. The two compounds which were missed were *Juledag* and *Påskedag* (Christmas and Easter). Needless to say these are both extremely high-frequency first constituents with entrenched forms which would be remembered.
17. Note that there is a default rule which places an /n/ before the *-ing* suffix in these formations. Analogy models like AM, SimNet and TiMBL readily provide an intuitive explanation for this phenomenon (both diachronically and synchronically), since a chosen analogical model will almost always have this same (otherwise meaningless) infix in those environments where it should be expected.
18. The adjusted averages were calculated by averaging the raw misclassification rates and then discarding those which deviated from this average by more than one standard deviation. The very few results which were discarded included individuals who didn't finish the task or classified all items the same. In this case the raw misclassification rate was only 2% higher than the adjusted rate. This procedure also reduced the variance within the responses for the second experiment by a factor of 5 or 6.
19. The /ʃ/ and /p/ graphemes had only one variant, "Ø" and "p" respectively. The variants for /t/ were "t" and "tt" while those for /k/ included "c", "k" and "ck".

## References

- Aha, David W., Dennis Kibler, & Marc Albert (1991). Instance-based learning algorithms. *Machine Learning*, 7, 37–66.
- Anttila, Raimo (1977). *Analogy*. The Hague: Mouton.
- Aronoff, Mark (1976). *Word Formation in Generative Grammar*. Cambridge: MIT Press.
- Augst, Gerhard (1975). Über das Fugenmorphem bei Zusammensetzungen. In G. Augst (Ed.), *Untersuchungen zum Morpheminventar der deutschen Gegenwartssprache*. Tübingen: Narr.

- Baayen, R. Harald (1995). Review of R. Skousen: *Analogy and structure*. *Language*, 71, 390–396.
- Baayen, R. Harald, Ton Dijkstra, & Robert Schreuder (1997). Singulars and plurals in Dutch: evidence for a parallel dual route model. *Journal of Memory and Language*, 36, 94–117.
- Becker, Thomas (1990). *Analogie und morphologische Theorie*. München: Wilhelm Fink.
- Becker, Thomas (1992). Compounding in German. *Rivista di Linguistica*, 4, 5–36.
- Bod, Rens (1998). *Beyond grammar: an experience-based theory of language*. Cambridge: Cambridge University Press.
- Booij, Geert E. (1994). Against split morphology. In G. E. Booij & J. van Marle (Eds.) *Yearbook of morphology 1993* (pp. 27–50). Dordrecht: Kluwer Academic Publishers.
- Bybee, Joan L., & Elly Pardo (1981). Morphological and lexical conditioning of rules: experimental evidence from Spanish. *Linguistics*, 19, 937–968.
- Bybee, Joan L. (1985). *Morphology: a study of the relation between meaning and form*. Amsterdam: Benjamins.
- Chandler, Steve (1995). Non-declarative linguistics: some neuropsychological perspectives. *Rivista di Linguistica*, 7, 233–247.
- Chomsky, Noam (1957). *Syntactic structures*. The Hague: Mouton.
- Chomsky, Noam (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1994). Skousen's analogical modeling algorithm: a comparison with lazy learning. In D. Jones (Ed.), *Proceedings of the International Conference on New Methods in Language Processing* (pp. 1–7). Manchester: UMIST.
- Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, & Antal van den Bosch (1999). TiMBL: Tilburg memory based learner reference guide 2.0. Technical Report 99–01. ILK Research Group, Tilburg University.
- Daelemans, Walter, Antal van den Bosch, & Jakub Zavrel (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 11, 11–43.
- Daugherty, Kim, & Mark Seidenberg (1994). Beyond rules and exceptions: a connectionist approach to inflectional morphology. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The Reality of Linguistic Rules*. Amsterdam: Benjamins.
- Gasser, Michael (1997). Transfer in a connectionist model of the acquisition of morphology. In H. Baayen & R. Schroeder (Eds.), *Yearbook of Morphology 1996* (pp. 97–116).
- Genesereth, Michael R., & Nils J. Nilsson (1987). *Logical foundations of artificial intelligence*. Los Altos: Morgan Kaufmann Publishers.
- Gentner, Dedre, & Arthur B. Markman. (1997). Structure mapping in analogy and similarity. *American Psychologist*, 52, 45–56.
- Grossberg, Stephen (1976). Adaptive pattern classification and universal recoding: Part I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23, 121–134.
- Grossberg, Stephen (1978). A theory of visual coding, memory and development. In E. L. J. Leeuwenberg & H. F. J. M. Buffart (Eds.), *Formal Theories of Visual Perception*. New York: Wiley.
- Hansen, Aage (1967). *Moderne Dansk*. Copenhagen: Gyldendal.
- Holyoak, Keith J., & Paul Thagard (1996). *Mental leaps: analogy in creative thought*. Cambridge, MA: MIT Press.

- Johnson, Dale D., & Richard L. Venezky (1976). Models for predicting how adults pronounce vowel digraph spellings in unfamiliar words. *Visible Language*, 10, 257–268.
- Kiparsky, Paul (1974). Remarks on analogical change. In J. M. Anderson & C. Jones (Eds.), *Historical Linguistics. Proceedings of the First International Conference on Historical Linguistics*. Amsterdam: North-Holland.
- Kiparsky, Paul (1978). Analogical change as a problem for linguistic theory. Reprinted in P. Kiparsky (Ed.), *Explanation in Phonology*. Dordrecht: Foris Publications.
- Køneke, M. (1986). *Danske Substantiviske Kompositas Natur, Funktion og Semantik*. Speciale i dansk, Institut for nordisk filologi, Københavns Universitet Amager.
- Kruschke, John K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, 99, 22–44.
- Labov, William (1973). The boundaries of words and their meanings. In C. N. Bailey & R. Shuy (Eds.), *New ways of analyzing variation in English*. Washington, DC: Georgetown University Press.
- MacWhinney, Brian, & Jared Leinbach (1991). Implementations are not conceptualization: revising the verb learning model. *Cognition*, 40, 121–157.
- McClelland, James L., & David E. Rumelhart (1989). *Explorations in parallel distributed processing: a handbook of models, programs, and exercises*. Cambridge, MA: MIT Press.
- Messick, Samuel J., & Charles M. Solley (1957). Probability learning in children: some exploratory studies. *Journal of Genetic Psychology*, 90, 23–32.
- Mitchell, T. M. (1978). Version spaces: an approach to concept learning. Doctoral dissertation, Stanford University.
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18, 203–226.
- Nosofsky, Robert M. (1984). Choice, similarity and the context theory of classification. *Journal of Experimental Psychology. Learning, Memory and Cognition*, 10, 104–114.
- Nudansk Ordbog (1986). 13th Edition. Copenhagen: Politikens Forlag.
- Ohala, John J. (1974). Experimental historical phonology. In J. M. Anderson & C. Jones (Eds.), *Historical linguistics. Proceedings of the first international conference on historical linguistics*. Amsterdam: North-Holland.
- Paunonen, Heikki (1973). On free variation. *Suomalais-Ugrilaisen Seuran Aikakauskirja*, 72, 285–300.
- Pinker, Steven (1999). *Words and rules: the ingredients of language*. London: Weidenfeld & Nicolson.
- Riber Petersen, Pia (1984). *Nye Ord i Dansk 1955–1975*. Copenhagen: Gyldendal.
- Riesbeck, Christopher K., & Roger S. Schank (1989). *Inside case-based reasoning*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Rumelhart, David E., James L. McClelland, & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press.
- Shanks, David R. (1995). *The psychology of associative learning*. Cambridge: Cambridge University Press.
- Shepard, Roger N., Carl L. Hovland, & Herbert M. Jenkins (1961). *Learning and memorization of classifications* (Psychological Monographs 75). Washington, DC: American Psychological Association.

- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1995). Analogy: a non-rule alternative to neural networks. *Rivista di Linguistica*, 7, 213–232.
- Spencer, Andrew (1988). Bracketing paradoxes and the English lexicon. *Language*, 64, 663–682.
- Stemberger, Joseph P. (1990). Wordshape errors in language production. *Cognition*, 35, 123–157.
- VanLehn, Kurt, & William Ball (1987). A version space approach to learning context-free grammars. *Machine Learning*, 2, 39–74.
- Vinterberg, Hermann, & C. A. Bodelsen (1966). *Dansk-Engelsk Ordbog* (2nd ed.). Copenhagen: Gyldendal.
- Vinterberg, Hermann, & Jens Axelsen (1979). *Dansk-Engelsk Ordbog* (3rd ed.). Copenhagen: Gyldendal.
- Winston, Patrick Henry (1975). Learning structural descriptions from examples. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill.

## Appendix

### A. Numeric presentation set used with human subjects in Experiment 1:

002-A 011-A 022-B 031-B 103-A 110-A 120-B 201-A  
210-A 213-A 230-B 231-B 233-B 300-A 313-A 322-B

### B. Test items used with human subjects in Experiment 1:

001\_\_ 130\_\_ 003\_\_ 122\_\_ 013\_\_ 023\_\_ 100\_\_ 131\_\_  
111\_\_ 020\_\_ 112\_\_ 320\_\_ 101\_\_ 333\_\_ 301\_\_ 133\_\_  
331\_\_ 311\_\_ 232\_\_ 012\_\_ 332\_\_ 202\_\_ 221\_\_ 021\_\_  
223\_\_ 200\_\_ 323\_\_ 010\_\_ 220\_\_ 113\_\_ 030\_\_ 303\_\_

312\_\_ 121\_\_ 310\_\_ 033\_\_  
102\_\_ 330\_\_ 203\_\_ 123\_\_  
222\_\_ 000\_\_ 321\_\_ 302\_\_  
132\_\_ 211\_\_ 032\_\_ 212\_\_

### C. Language presentation set used with human subjects in Experiment 2 (T1):\*

at-1 cap-1 cop-2 kick-1 pack-1 pap-1 pat-1 pick-1  
pit-1 pock-2 pop-2 pot-2 puck-2 putt-2 tap-1 up-2

---

\* All words were taken from the *American Heritage Dictionary of the English Language*, New College Edition. Boston: Houghton Mifflin, 1979.

D. Test set used with human subjects in Experiment 2 (T1):

it \_\_ tot \_\_ tat \_\_ tut \_\_ tack \_\_ cot \_\_ tick \_\_ top \_\_  
 cut \_\_ cat \_\_ cup \_\_ pip \_\_ tuck \_\_ kit \_\_ tock \_\_ tip \_\_

E. Language presentation set used with human subjects in Experiment 2 (T2):

cap-1 cat-1 cop-2 cup-2 cut-2 kit-1 pack-1 pick-1  
 pit-1 pop-2 tack-1 tap-1 tip-1 top-2 tot-2 tuck-2

F. Test set used with human subjects in Experiment 2 (T2):

dot \_\_ bit \_\_ gut \_\_ bock \_\_ got \_\_ bat \_\_ bop \_\_ duck \_\_  
 gap \_\_ dock \_\_ back \_\_ buck \_\_ dip \_\_ but \_\_ gat \_\_ guck \_\_

	number of samples	average misclassified	standard deviation	variance
Numeric data				
raw	46	.213	.17	.029
adjusted	44	<b>.193</b>	.14	.020
Language 1				
raw	19	.112	.17	.030
adjusted	17	<b>.059</b>	.07	.005
Language 2				
raw	19	.076	.15	.021
adjusted	18	<b>.045</b>	.06	.004

## Exemplar-driven analogy in Optimality Theory

James Myers

### 1. Introduction

The term “analogy” may be something of a dirty word for most theoretical linguists, but it shouldn’t be forgotten that it was theoretical linguists who first coined the term as it applies to language. Of course, when the Neogrammarians wrote about paradigm leveling or four-part proportional analogy, it was often just in passing on the way to what really interested them, namely regular rules. The same has been true for their structuralist and generative descendants, with a major excuse usually being that analogy was too vague a notion to deal with in a formal model. While this excuse is no longer valid, a sharp divide nevertheless remains between the mostly positive attitude of computer modelers and psycholinguists towards analogical approaches, and the mostly negative attitude of generative linguists.

Recently this has begun to change. Optimality Theory (or OT, to use its standard abbreviation; Prince & Smolensky 1993, 1997) is a formal generative model of language that has certain properties that make it capable of handling true exemplar-driven analogy (as opposed to earlier generative reanalyses of analogy using general rules, e.g. Kiparsky 1978, 1988). Recognition of this fact is gradually filtering through the “mainstream” OT literature, with prominent researchers such as Kenstowicz (1995, 1997), Steriade (1999a, 1999b, 2000), Burzio (1997a, 1997b, 1999, 2000, 2002), and Hayes (1999b) beginning to peek out of the analogical closet, along with newer scholars trained in the generative tradition, including Benua (1995, 1997a, 1997b), Alderete (1999), Kirchner (1999), and Albright (2002). My own contribution has been to try to push the analogical approach as far as it can go in an OT formalism, in the hope that generative linguists and nongenerativists working on analogy can better share insights.

To this end, I set myself the goal of building a completely explicit formal model of the traditional linguistic notion of four-part proportional analogy (focussing on



phonological analogy), using nothing but devices already found in the OT literature. In this paper I first review some of this literature to show that my model is not extremely radical by current generative standards, and then I describe how my model actually works. Next I prove that it is equivalent to the simplest possible kind of connectionist network, a linear associator, which has well-known strengths and weaknesses (see e.g. Anderson 1995). I then compare the explicit quantitative predictions made by the OT model with those made by Analogical Modeling (AM; Skousen 1989, 1992). In general, the OT model of analogy performs much worse than AM, but the fact that it makes quantitative predictions at all, and that these predictions are far more accurate than chance, convinces me that it is in principle possible to build a bridge between generative and nongenerative approaches to analogy. Moreover, I show how insights from AM and connectionism may be used to improve the quantitative accuracy of the model (though this requires going beyond OT formalism). Finally, like traditional generative theories of language, and unlike AM, my OT model represents both inputs and outputs with features, and it is also capable of incorporating non-analogical factors. These two properties seem to give it an advantage in handling certain empirical phenomena, and so I hope that in building the bridge between OT and AM, the exchange of insights will run both ways.

## 2. Analogy in Optimality Theory

In this section I discuss some of the properties that make OT more similar to analogical approaches than previous generative models and show how explicitly analogical analyses are becoming more common in the OT literature. This discussion will then lead to the fully analogical OT model described in the following section.

The most obvious property that makes OT analogy-friendly is that it is non-derivational and surface-based. This results from its being a descendent both of standard generative theories of linguistic constraints and of so-called constraint-satisfaction connectionist networks (see especially Prince & Smolensky 1997).

Another important property of OT is that it makes a foundational distinction between two kinds of linguistic constraints. So-called *Structure constraints* include those like the famous NoCoDA, which require output forms to conform to universal structural principles that may or may not be motivated by extra-linguistic factors. Such constraints are the clear descendents of generations of generative constraints, including the syntactic principles of Government and Binding theory. However, OT also posits so-called *Faithfulness constraints*, whose sole job it is to require forms to be “faithful” to themselves or to other forms – that is, to prevent the Structure constraints from doing anything. If OT grammars had only

Structure constraints, all languages would be reduced to the maximally unmarked form, which is clearly not the case. With Faithfulness constraints, OT thus makes it explicit, perhaps for the first time in generative linguistics, that at least half of the human language faculty involves brute memorization of forms as they are, regardless of how inelegant, costly, or marked they may be. A purely analogical OT model, then, would be one that is built solely out of Faithfulness constraints.

The fact that exemplar-driven analogy is driven by exemplars may seem to pose an impossible challenge for OT, since OT constraints are usually described as completely general, even universal or innate. From the very beginning, however, it has been recognized that it is often necessary to posit constraints that are specific to specific classes, or even to particular lexical items. For example, in perhaps the most famous application of OT, McCarthy and Prince's (1993a) analysis of Tagalog *um* infixation, just such a constraint plays a crucial role. The claim of this analysis is that the distribution of *um* can be explained if one thinks of it as being affixed as close to the beginning of the word as possible without creating a new syllable coda. Disallowing the coda is the responsibility of the universal Structure Constraint NoCoDA, but clearly there is no universal principle requiring affixes to appear towards the beginning of a word. To account for this fact, McCarthy and Prince (1993a) propose a universal Faithfulness constraint EDGEMOST which is parameterized by word edge (in this case, the left one) and by morpheme (in this case, *um*). Hence EDGEMOST(Left, *um*) requires the morpheme *um* to appear at the left edge of a word, meaning that the further away *um* is from this edge, the more it violates this constraint. McCarthy and Prince (1993b) later re-analyzed EDGEMOST(Left, *um*) within the Generalized Alignment approach, now calling it ALIGN(*um*, Left, Stem, Left), but it still has to refer specifically to the morpheme *um*.

Universal constraints parameterized by lexical item are sometimes called *Parochial constraints* (e.g. Hammond 1995), and they are ubiquitous in the OT literature. For example, to deal with the different phonological behaviors of the two major classes of English derivational morphology (e.g. the  $\emptyset \sim [n]$  alternation in *condemn-condemnation* versus no alternation in *condemn-condemnable*), Benua (1997a, 1997b) uses parochial constraints parameterized to each class, which then allows her to rank the constraints separately and derive the phonological differences (this analysis will be described in more detail below). Some OT researchers (e.g. Russell 1995, 1999; Hammond 1995, 1997; Golston 1996) have gone much further, proposing models in which morphemes or words are themselves (sets of) parochial constraints.

Thus to make analogy exemplar-driven in an OT model, we need parochial Faithfulness constraints. For the purposes of my model of analogy described in the next section, I maintain the standard OT assumption that distinguishes *inputs* (roughly equivalent to the underlying representations of earlier generative theories

of phonology) from *outputs* (i.e. surface forms), and the familiar set of Faithfulness constraints called IDENT-IO which require input (I) and output (O) forms to be identical in some feature (McCarthy & Prince 1995). Somewhat new is my assumption that IDENT-IO is parochial rather than general, and that it operates over whole words, not individual morphemes. The general form of this parameterized constraint is IDENT-IO(*W*;F), where *W* represents a word, and F a feature. For example, the constraint IDENT-IO(*bat*;[labial]) would mean that the word *bat* cannot change its value of [labial] from input to output. Translated into more theory-neutral terminology, this kind of constraint has the job of preventing analogy (or other factors) from affecting one particular phonological property in one particular word.

The use of the parameters *W* and F require some brief comments. As is the case for any model of analogy, the particular representation used may have enormous consequences for how it works (see e.g. Baayen's 1995 comments on Skousen 1992, or Pinker & Prince's 1988 criticisms of Rumelhart & McClelland 1986). By calling F a "feature" I don't necessarily adopt the standard distinctive features of generative phonology; setting F to /b/ or VOT = 20 msec or even [bæt] may prove to work better. Likewise, I don't necessarily follow the linguist's traditional focus on types rather than tokens. *W* thus may be taken to represent a particular token of a word (as spoken or heard by some individual). Token-based approaches to phonology are becoming more common (see e.g. Bybee 2000; Kirchner 1999), and I will also adopt this assumption in this paper, since as we will see, it allows my OT model to handle lexical frequency effects in a natural way.

Nevertheless, IDENT-IO is not the sort of Faithfulness constraint that can itself give rise to analogy, which of course involves relations *between* words. Fortunately, here is where recent developments in OT theorizing become particularly useful for analogical purposes. Starting with McCarthy and Prince (1995), Faithfulness has been generalized from involving only inputs and outputs, to involving any pairs of representations. McCarthy and Prince (1995) applied this new theory (called *correspondence theory*) to two parts of a single output (stem and reduplicant in reduplicated forms), and soon thereafter Kenstowicz (1995, 1997) and Benua (1995, 1997a, 1997b) applied it to pairs of morphologically related output forms.

Output-output (OO) correspondence allows for analyses that are strikingly different from anything that had previously been allowed in generative theory, and strikingly similar to traditional theories of analogy. For example, a blatant use of paradigm leveling forms the basis of Benua's (1997a, 1997b) analysis of *condemn-condemnation/condemnable* alluded to earlier. In essence, her analysis suggests that while *condemn* may lose its (supposedly) underlying /n/ due to a Structure constraint against syllable-final [mn] sequences, the loss of the /n/ in *condemnable* is by analogy: a parochial Faithfulness constraint, specific to the class of morphology that includes *-able*, requires *condemn* and *condemnable* to share the property


of [n]-lessness. Technically this is handled by ranking the anti-[mn] constraint at the top (it is never violated), then ranking the OO-constraint above the IO-constraint (i.e. it's better for *condemnable* to become similar to *condemn* than to keep its underlying /n/). Another parochial OO-constraint for morphology like *-ation* is ranked below the IO-constraint (i.e. it's better for *condemnation* to keep its original /n/ than to become similar to *condemn*).

Without necessarily condoning the particular application of analogy here, it's worth noting the important sociological development that such analyses represent. First, while there are some grumblings about them in the OT literature (e.g. Booij 1997; Hale, Kissock & Reiss 1998), they are becoming more common; other examples include Burzio (1997a, 1997b, 1999, 2000, 2002) and Steriade (2000). Second, these authors openly acknowledge that what they are doing should be called analogy; Kenstowicz (1995, 1997) makes this particularly explicit. Third, analogical analyses of this sort have been accepted so rapidly that one has to conclude that they are filling a need that has long been felt but could never before be expressed.


For example, the standard generative phonology textbook Kenstowicz (1994) (written just before OT came to dominate phonological theory) argues that the vowel-length differences many speakers show before the flaps in *writer* and *rider* must be due to ordered rules (i.e. vowel-lengthening before flapping), just as argued in Chomsky and Halle (1968). Ironically, Kenstowicz (1994:71–72) does consider an alternative analysis in which *writer* contains a short vowel by analogy with *write* (more precisely, *writer* contains the short-vowel allomorph of *write*), but then rejects it. With output-output correspondence (developed partly with the help of Kenstowicz himself), the analogical analysis can now be formalized by positing a Faithfulness constraint IDENT-OO([vowel length]) that outranks the Structure constraints requiring vowels to be long before voiced consonants. Although I don't know of any work in the OT literature that actually presents this analysis, it's not difficult to flesh out the details. To illustrate this, and to give readers less familiar with OT notation a chance to practice before things get more technical later on, I provide the details here.

First, a Structure constraint requiring consonants to be flapped in certain intervocalic environments (call it FLAP) must be ranked higher than the Faithfulness constraint IDENT-OO([vowel length]), which is in turn ranked above the Structure constraint requiring long vowels only before voiced consonants (call it LONG). Tableaux 1 and 2 (as they are called) then illustrate what happens in the pairs *ride-rider* and *write-writer*.

(1) Why both *ride* and *rider* have long vowels (no analogy)

Input:	FLAP	IDENT-OO ([vowel length])	LONG	IDENT-IO ([vowel length])
raid-raidr	*		**	
ra:ιδ-raidr	*	*	*	*
raid-ra:ιδr	*	*	*	*
ra:ιδ-ra:ιδr	*			**
raid-raiDr			**	
ra:ιδ-raiDr		*	*	*
raid-ra:iDr		*	*	*
 ra:ιδ-ra:iDr				**

(2) Why both *write* and *writer* have short vowels (paradigmatic leveling)

Input:	FLAP	IDENT-OO ([vowel length])	LONG	IDENT-IO ([vowel length])
rait-raitr	*			
ra:it-raitr	*	*	*	*
rait-ra:itr	*	*	*	*
ra:it-ra:itr	*		**	**
 rait-raiDr			*	
ra:it-raiDr		*	**	*
rait-ra:iDr		*		*
ra:it-ra:iDr			*	**

As is usual in the OT literature, I list possible outputs in the first column (here, all possible combinations of vowel length with flapping). Constraints are listed left to right from highest to lowest rank (an OT grammar is defined by a constraint ranking). Stars indicate violations of a given candidate output by a given constraint; multiple stars mean multiple violations by the same candidate. The optimal candidate (i.e. the one predicted to be grammatical, marked with a pointing finger) is in the subset of candidates that least violate the highest-ranked constraint, and in this subset, it is in the subset of candidates that least violate the second-highest-ranked constraint, and so on. Perhaps a quicker way to spot the optimal candidate is to mentally translate the stars into digits (\* = 1, \*\* = 2, “ ” = 0, etc.), and the row of stars for a given candidate into a number (e.g. 1020 for the first row in (1)). The optimal candidate is then the output associated with the lowest number (e.g. in (1), the candidate marked with the pointing finger is associated with the number 0002).

Note that in (1), no analogy occurs. The optimal candidate here is simply the one that obeys both Structure constraints (FLAP and LONG). By contrast, in (2), the structurally best candidate is *rait-rai:Dr* (second from bottom), but that is not the

one chosen. Instead the optimal candidate is one in which *write* and *writer* have vowels with the same duration, since IDENT-OO([vowel length]) outranks LONG.

The main point to take away is that by using output-output correspondence, the traditional generative rule-ordering analysis can be replaced with an analogical one (specifically, paradigmatic leveling), and this analogical analysis is formally precise. My proposed OT model of analogy, however, goes much further than the examples just sketched.

### 3. Four-part proportional analogy in Optimality Theory

To the best of my knowledge, nothing in the OT literature has taken the logical next step, which is to try to build an OT model of four-part proportional analogy. This more general form of analogy subsumes paradigm leveling as a special case, and it is far more powerful. Moreover, as I noted in the introduction, it is something with a long tradition in linguistics, and thus I hope less threatening to unconditioned generative linguists than more sophisticated models of analogy like AM. In this section I show how to bring this kind of analogy into OT, focusing on technical issues (see Myers 2000a for discussion of the applications of the model to linguistic data that pose serious problems for traditional generative models without analogy; also Green 2001).

The first thing to do, it should be clear, is to make output-output correspondence completely parochial, rather than requiring that it only apply within paradigms. Otherwise we can't describe the irregularization of *dive* (past tense *dove*) as analogy with *drive-drove*. Thus I posit OO-constraints of the form IDENT-OO( $W_i, W_j; F_k$ ), where  $W_i$  and  $W_j$  are words (or word tokens) and  $F_k$  is some feature (in the extended sense of "feature" discussed earlier).

But of course analogy does not work to make any random pair of words similar to each other. To constrain the IDENT-OO constraints, we have to go somewhat beyond the OT mainstream, but only somewhat. The problem is this. In a proportional analogy, there are four items ( $a, b, c, d$ ) standing in the relation  $a : b :: c : d$ . This is standardly taken to mean that if  $a$  shares feature  $F$  with  $c$ , then  $b$  shares feature  $G$  with  $d$ . In terms of parochial IDENT-OO constraints, this says: if IDENT-OO( $a, c; F$ ) then IDENT-OO( $b, d; G$ ). Is there any way of creating a new constraint that is violated if and only if this logical implication is false?

As it happens, there is. In the grab bag of OT innovations is the notion of *constraint conjunction*, which creates new constraints with Boolean operators (see Smolensky 1995; Crowhurst & Hewitt 1997; and Balari, Marín, & Vallverdú 2000 for non-analogical applications). It turns out that the constraint we need has the

form given below (conjoined with the AND operator), which is violated if and only if at least one of the two component constraints is violated.

- (3) IDENT-OO( $a, c; F$ )  $\wedge$  IDENT-OO( $b, d; G$ )  
 [abbreviation: OO $\wedge$ OO-( $a, c; F$ )( $b, d; G$ )]

That this constraint has the desired behavior can be seen if we consider a toy lexicon containing four items  $a, b, c, d$ . If  $a$  and  $c$  are already similar, as in (4a below),  $d$  will change its form to conform to  $b$ . However, if  $a$  and  $c$  aren't already similar, as in (4b below),  $d$  won't change. (The conjoined constraint is violated in both candidates since the first component constraint is violated, and hence it has no effect on the choice of optimal output.)

- (4a)  $a$  and  $c$  are similar

$a = [+F], b = [+G],$ $c = [+F], d = [-G]$	OO $\wedge$ OO-( $a, c; F$ )( $b, d; G$ )	IO-( $d; G$ )
$d = [-G]$	*	
$\textcircled{\text{e}} d = [+G]$		*

- (4b)  $a$  and  $c$  are not similar

$a = [-F], b = [+G],$ $c = [+F], d = [-G]$	OO $\wedge$ OO-( $a, c; F$ )( $b, d; G$ )	IO-( $d; G$ )
$\textcircled{\text{e}} d = [-G]$	*	
$d = [+G]$	*	*

While this makes analogical change in  $d$  contingent on the properties of  $a, b$ , and  $c$ , there is still nothing preventing us from bringing a random quartet of words together into a spurious proportion. To deal with this, I fall back on the time-honored generative tradition of positing a universal principle. This principle also explicitly disallows IDENT-OO constraints acting on their own outside of proportions.

- (5) PROPORTION PRINCIPLE

Given the items  $a, b, c, d$  in a language and the features  $F$  and  $G$ , the conjoined constraint IDENT-OO( $a, c; F$ ) $\wedge$ IDENT-OO( $b, d; G$ ) is generated if and only if there exists a single outcome function  $o$  such that  $o(a) = b$  and  $o(c) = d$ . IDENT-OO constraints do not exist outside of such conjoined constraints.

In justification of this move, I point out that all other models of analogy (including traditional notions, AM, and connectionism) tacitly assume something very much like this principle. For example, if one runs an AM simulation on data points associated randomly with outcomes (e.g. *drive-ate, strive-banana*), one shouldn't expect to get particularly insightful results. Any theory of analogy thus

presupposes a theory of “relatedness”; the Proportion Principle merely makes this presupposition explicit.

This completes the set of supplemental devices needed for the OT model of analogy. For the remainder of its powers the model relies on nothing more than the central OT notion of extrinsic constraint ranking. This is all that is needed to deal with the notoriously capricious nature of analogy (which often fails to apply in one language in precisely the environment where it readily applies in another). For example, we can assume that all English dialects have constraints like the following, which requires the past tense forms of *drive* and *dive* to have the same vowel since the present tense forms have the same rime.

$$(6) \text{ IDENT-OO}(\textit{drive}, \textit{dive}; [\textit{ayv}]) \wedge \text{ IDENT-OO}(\textit{drive}_{\textit{PAST}}, \textit{dive}_{\textit{PAST}}; [\textit{o}])$$

In a dialect where *dive* is regular, this constraint (whose existence is required by the Proportion Principle) is stripped of all power by being extrinsically ranked below the IO-constraint that keeps the past tense form of *dive* in its original form, as in (7a below). By contrast, in a dialect where *dive* is irregular, these constraints are ranked in the reverse order, as in (7b below), and the past tense of *dive* becomes *dove* by analogy with *drive-drove*.

(7a) a *dive-dived* dialect

[drayv], [drov], [dayv], [dayvd]	IO-( <i>dive</i> <sub>PAST</sub> ; [ay])	OO^OO-( <i>drive, dive</i> ; [ayv]) ( <i>drive</i> <sub>PAST</sub> , <i>dive</i> <sub>PAST</sub> ; [o])
☞ [dayvd]		*
[dov]	*	

(7b) a *dive-dove* dialect

[drayv], [drov], [dayv], [dayvd]	OO^OO-( <i>drive, dive</i> ; [ayv]) ( <i>drive</i> <sub>PAST</sub> , <i>dive</i> <sub>PAST</sub> ; [o])	IO-( <i>dive</i> <sub>PAST</sub> ; [ay])
[dayvd]	*	
☞ [dov]		*

Paradoxically, extrinsic constraint ranking also turns out to provide a neat account of universal properties of analogy, such as gradient similarity effects, gang effects, and frequency effects. The explanation for this is that under the null hypothesis, OT constraints can be extrinsically ranked in every possible way cross-linguistically. If we examine the quantitative predictions of the completely random ranking of analogical conjoined constraints, the probability that a given form will be changed by a given analogy is determined entirely by the number of triggering analogical constraints. For example, the more similar a target form is to an analogical trigger, the more features they will share, and thus the more analogical constraints there will be that are parochial with respect to those words (i.e. one such constraint per



shared feature). Likewise, the larger the gang of analogical triggers  $\{W_1, \dots, W_n\}$  that are similar to a given target form  $W_{n+1}$ , the more analogical constraints there will be that are parochial with respect to those words (namely constraints referring to  $W_1$  and  $W_{n+1}$ ,  $W_2$  and  $W_{n+1}$ , and so on).

The same argument works for frequency effects. To make this completely explicit, consider the following toy lexicon containing three word types  $a, b, c$ , where  $a$  and  $b$  are both equally similar to  $c$ , but  $a$  is twice as frequent as  $b$ . The question concerns which result the OT model predicts to be more likely:  $c$  (in its form for  $o(c)$ ) analogizing to  $a$  or  $c$  analogizing to  $b$ .

- (8) Lexicon:  $a = [+F], o(a) = [+G],$   
 $b = [+F], o(b) = [-G],$   
 $c = [+F]$   
 Data set:  $\{a, a, b\}$

Using constraints that are parochial with respect to tokens rather than types, the Proportion Principle generates the analogies given in the following tableau (analogies between  $a$  and  $b$  are left out, since we're focusing on the behavior of  $c$ ). Note that there are two constraints enforcing similarity between  $a$  and  $c$ , and only one enforcing similarity between  $b$  and  $c$ . Note also that there is no claim that these constraints are extrinsically ranked in any particular way; following the convention in the OT literature, I indicate the lack of ranking by separating the constraint columns with dashed lines.

(9)

$a = [+F], o(a) = [+G],$ $b = [+F], o(b) = [-G],$ $c = [+F]$	OO^OO- ( $a, c;F$ ) $(o(a), o(c);G)$	OO^OO- ( $a, c;F$ ) $(o(a), o(c);G)$	OO^OO- ( $b, c;F$ ) $(o(b), o(c);G)$
$o(c) = [+G]$			*
$o(c) = [-G]$	*	*	

The question then becomes a mathematical one: given completely random constraint ranking, what is the probability that the candidate output  $o(c) = [+G]$  will be chosen as optimal? While the analogical flavor of this question is new, the issue of variable constraint ranking in OT is not. Going back to Kiparsky (1993), OT researchers have used variable ranking to deal with variable linguistic phenomena. (Other applications of variable constraint ranking in OT include Anttila 1997; Anttila & Cho 1998; Nagy & Reynolds 1997; Hayes & MacEachern 1998; Boersma 1998; Boersma & Hayes 2001; and Myers 2000b.) Most useful for our purposes here, Myers (2000b) proves several theorems for calculating precise probabilities without having to face the factorial explosion that occurs when all  $n!$  rankings of  $n$  constraints are examined. The central result is what Myers (2000b) calls Anttila's Theorem (after Anttila 1997), stated here as follows:

(10) ANTILA'S THEOREM

If there are only two competing candidates  $X_1$  and  $X_2$ , the probability that candidate  $X_1$  will be chosen as optimal under completely random constraint ranking is

$$P(X_1) = |C_{X_1}| / [ |C_{X_1}| + |C_{X_2}| ],$$

where  $|C_{X_i}|$  = number of constraints that evaluate  $X_i$  over the alternative candidate

In other words, if there are only two candidates to consider, the probability that one will be optimal is just the proportion of constraints that favor it out of all constraints that favor either candidate. (Constraints that treat all candidates the same way can be entirely ignored, according to a theorem that Myers 2000b calls Noncommittal Constraint Irrelevance.)

Specifically, what we find with the analysis in (9) are the following probabilities:  $P(o(c) = [+G]) = 2/3$ ,  $P(o(c) = [-G]) = 1/3$ . (Readers wanting to get a hands-on feel for Anttila's Theorem may write out all six (= 3!) constraint rankings implied by the tableau in (9) to confirm that it does indeed work.) Thus  $c$  is twice as likely to conform to the analogy with  $a$  as with  $b$ . This demonstrates that this OT model shows one major kind of frequency effect: the more frequent the analogical trigger, the stronger its analogical force.

The model is also capable of handling the flip side of frequency effects, namely the more frequent a potential analogical target, the less likely it is to undergo analogy (as in the blocking of regularization in high-frequency English verbs). To represent target frequency, we use token-parameterized IDENT-IO constraints. Continuing with the above example, we give word  $o(c)$  an initial value of  $[-G]$  and a token frequency of 2, resulting in the following tableau. (The first three constraints are the same as in (9) above.) Anttila's Theorem now predicts the probabilities  $P(o(c) = [+G]) = 2/5$ ,  $P(o(c) = [-G]) = 3/5$ . Thus an increase in the frequency of an analogical target decreases its likelihood of undergoing an analogy (here, a drop in  $P(+G)$  from 0.667 to 0.400).

(11)

$a = [+F], o(a) = [+G],$	OO^OO-	OO^OO-	OO^OO-	IO-( $o(c);G$ )	IO-( $o(c);G$ )
$b = [+F], o(b) = [-G],$	( $a, c;F$ )	( $a, c;F$ )	( $b, c;F$ )		
$c = [+F], o(c) = [-G]$	( $o(a), o(c);G$ )!	( $o(a), o(c);G$ )!	( $o(b), o(c);G$ )!		
$o(c) = [+G]$			*	*	*
$o(c) = [-G]$	*	*			

This, then, is an OT model of true exemplar-driven analogy. It assumes virtually nothing that has not already been discussed in the OT literature, and its major technical devices (output-output correspondence and constraint ranking) are en-

tirely mainstream. To understand precisely where the OT model stands among nongenerative models of analogy, however, we need to examine the nature of its quantitative behavior more closely. This is the subject of the following section.

#### 4. Analogy in Optimality Theory and connectionism

In this paper I have been using the term “analogy” to refer to an empirical fact that has been recognized by linguists for almost two hundred years, not just the particular theory of it provided by AM. Thus I have no qualms in listing connectionism as an alternative model of analogy. For example, Rumelhart & McClelland 1986, using a connectionist model, is one possible analogical analysis of English inflection; Derwing & Skousen 1994, using AM, is another. In this section I show that the OT model of analogy sits squarely in the connectionist tradition. In fact, under a reasonable representational assumption (also made in AM), it is exactly equivalent to the simplest kind of connectionist network, a linear associator. If a more complex representational scheme is used, its behavior is somewhat more complex, but is still essentially connectionist-like.

The representational assumption just alluded to involves supposing that the outcomes (i.e. the forms that the function  $o$  maps to) are atomic units, rather than composite forms built out of the same features that compose the data points. AM makes this assumption quite clearly (as do nearest-neighbor approaches; see elsewhere in this volume). For example, in the example in Skousen 1989:23–37, the basic forms are built out of three four-valued features, giving representations like 310 and 032, but the outcomes are the two distinct atoms  $e$  and  $r$ . It is not even immediately obvious how AM could be modified so that the outcomes themselves could be built out of features in any meaningful way (though I make an explicit suggestion in this direction in a later section). The atomic nature of the outcomes in AM makes it eminently suitable for morphological analogy, which involves choosing among a fixed set of distinct morphemes, but it may cause problems for certain kinds of phonological analogy, which may affect only part of a form at a time. This possible weakness of AM will be discussed further below, but first I will adopt the atomicity assumption and see what consequences it has for the OT model of analogy.

The general situation is as follows. We have a set of words (or word tokens)  $W_1, \dots, W_n$ , represented with features  $F_1, \dots, F_m$ , and an outcome function  $o$  mapping the words onto a set of atomic outcomes  $X_1, \dots, X_a$ . We want to know what analogy will do with a new word  $W_{n+1}$  given all possible rankings of all conjoined analogical constraints conforming to the Proportion Principle. How can we calculate the relative probabilities  $P(o(W_{n+1}) = X_1), \dots, P(o(W_{n+1}) = X_a)$ ?

At first this may seem like a very difficult problem. Since more than two candidate outputs are being considered, Anttila’s Theorem does not apply. Moreover, the behavior of the constraints may possibly vary quite unpredictably. This would leave us with the computationally irritating factorial problem of checking all possible constraint rankings. As it happens, however, the assumption of atomic outcomes makes the constraints so well behaved that a slight extension of Anttila’s Theorem can be used.

First, we can completely ignore all constraints that make no reference to  $W_{n+1}$  (e.g. those that require identity between  $W_1$  and  $W_2$ ). These will be vacuously obeyed by all possible outputs for  $W_{n+1}$ , and as stated by the theorem of Noncommittal Constraint Irrelevance mentioned earlier, constraints that don’t choose among any candidates can be removed without affecting probabilities under variable ranking. Now, all analogical constraints that do refer to  $W_{n+1}$  must have the following form (see 12 below) if they are to conform to the Proportion Principle. Note that in accordance with the atomicity assumption, the outputs  $o(W_i)$  and  $o(W_{n+1})$  in the second component of the conjoined constraint are completely identical, rather than merely sharing the value for a single feature.

$$(12) \text{ IDENT-OO}(W_i, W_{n+1};F_j) \wedge \text{IDENT-OO}(o(W_i), o(W_{n+1}))$$

Logically there are only four possible behaviors of this constraint. These are represented schematically in (13), where the stars indicate under what conditions the constraint is violated.

(13)

	$\text{IDENT-OO}(W_i, W_{n+1};F_j) \neg \text{IDENT-OO}(W_i, W_{n+1};F_j)$	
$\text{IDENT-OO}(o(W_i), o(W_{n+1}))$		*
$\neg \text{IDENT-OO}(o(W_i), o(W_{n+1}))$	*	*

Since our candidate outputs consist solely of possible outcomes for  $W_{n+1}$ , without varying the representation of  $W_{n+1}$  itself, the component constraint  $\text{IDENT-OO}(W_i, W_{n+1};F_j)$  must be either always obeyed or always disobeyed (for any given  $i$  and  $j$ ) across the whole set of candidate outputs. If it’s disobeyed (i.e.  $W_i$  and  $W_{n+1}$  are not identical in feature  $F_j$ ), then the conjoined constraint in (12) will evaluate all candidate outputs as a violation. In this case, Noncommittal Constraint Irrelevance means we can ignore this particular conjoined constraint. However, if this component constraint is obeyed (i.e.  $W_i$  and  $W_{n+1}$  are identical in feature  $F_j$ ), then the final decision is left to the other half of the conjoined constraint, namely  $\text{IDENT-OO}(o(W_i), o(W_{n+1}))$ .

Under what circumstances is  $\text{IDENT-OO}(o(W_i), o(W_{n+1}))$  obeyed? Here is where the assumption of atomicity is crucial. Given this assumption, this constraint is obeyed if and only if the outputs of  $W_i$  and  $W_{n+1}$  are entirely identical,

which means there is some atomic outcome  $X_k$  such that  $o(W_i) = o(W_{n+1}) = X_k$ . This means that this constraint is violated (along with the entire conjoined constraint) whenever  $o(W_{n+1})$  is some atomic outcome other than  $X_k$ . Thus if the premise is true (i.e.  $W_i$  and  $W_{n+1}$  are identical in some feature), the conjoined constraint will be violated by all candidate outputs except one (namely the one where  $o(W_{n+1}) = X_k$ ).

For example, suppose that  $o(W_1) = X_1$ , and that  $W_1$  and  $W_{n+1}$  are identical in feature  $F_1$  (e.g. they both share value  $[+F_1]$ ) but not in feature  $F_2$ . One corner of the resulting tableau (14) will thus appear as follows, where the stars in the bottom row symbolize the consistent violation of these constraints for all candidate outputs other than  $o(W_1) = X_1$ .

(14)

	OO^OO-	OO^OO-	
	$(W_1, W_{n+1}; F_1)(o(W_1), o(W_{n+1}))$	$(W_1, W_{n+1}; F_2)(o(W_1), o(W_{n+1}))$	...
$o(W_{n+1}) = X_1$		*	...
$o(W_{n+1}) = X_2$	*	*	...
...	*	*	...

In general, then, these constraints only act in two ways: either they don't do anything, or they reject all candidate outputs but one. This limitation of winners to at most one per constraint makes a slightly modified version of Anttila's Theorem applicable. The proof is virtually the same as that given in Myers 2000b for Anttila's Theorem, and may be informally stated as follows. Given Noncommittal Constraint Irrelevance, we only have to consider constraints that pick a single winner. The probability that a given candidate will win overall, then, is simply the probability that a constraint that favors it is ranked at the top (thus making all the other constraints powerless).

(15) Anttila's Theorem for constraints that choose at most one winner:

If all constraints evaluate at most one candidate as optimal, then the probability that candidate  $X_1$  is optimal overall, given completely random constraint ranking, is

$$P(X_1) = |C_{X_1}| / [ |C_{X_1}| + |C_{X_2}| \dots + |C_{X_a}| ],$$

where  $|C_{X_i}|$  = number of constraints that evaluate  $X_i$  over the alternative candidates

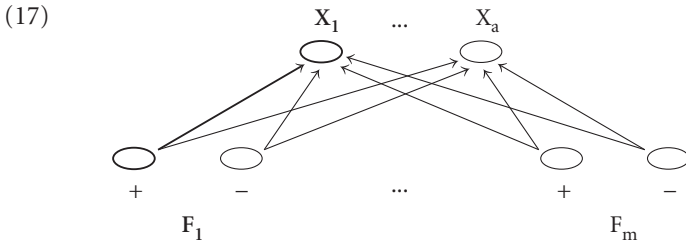
This theorem has a special interpretation in our case, however. The set  $C_{X_i}$  (i.e. the set of all constraints that evaluate  $X_i$  over the alternative candidates) contains all analogical constraints that require  $W_{n+1}$  to match some analogical trigger word  $W_k$

in some feature and that require the outcomes for  $W_{n+1}$  and  $W_k$  both to be  $X_i$ . Keep this sentence in mind; a variant of it will return shortly.

Now I proceed to show that this model works precisely the same way as a *linear associator*. In this simplest of all connectionist networks (see Anderson 1995 for a lucid introduction and references), there are two layers of nodes, and each node is connected to all other nodes in the other layer. In training such a model, two vectors of node activations are presented to the model, and learning occurs through a Hebbian rule, i.e. a connection is strengthened if the two nodes that it connects are simultaneously activated. In our case, node activations during training must be either 0 or 1, all connection weights are initialized to 0, and the rule increases a connection weight by adding 1 if and only if both connected nodes have activation 1. Using standard connectionist notation, the rule can be stated as follows:

(16)  $\Delta w_{ij} = a_i a_j$ ,  
 where  $w_{ij}$  is the weight of the connection between nodes  $i$  and  $j$ ;  $a_i$  and  $a_j$  are the activations of nodes  $i$  and  $j$ , respectively; and  $\Delta w_{ij}$  represents the amount added to  $w_{ij}$  each time  $a_i$  and  $a_j$  are changed.

As for the architecture of the network, one layer will of course consist of a set of nodes for the atomic outputs  $X_1, \dots, X_a$ . The other must consist of sets of nodes representing values of the features  $F_1, \dots, F_m$ , used for the word forms  $W_1, \dots, W_n$ . For example, if these features were binary, the architecture would be as shown in Figure (17).



To see how the model works, consider the situation illustrated above in (14), where  $o(W_1) = X_1$ , and  $W_1$  and  $W_{n+1}$  both share value  $[+F_1]$ . During training,  $W_1$  (represented in features) would be presented to the bottom layer, thus activating the node  $[+F_1]$ , and  $o(W_1)$  would be presented to the top layer, thus activating the node  $X_1$ . Since these two nodes would be simultaneously activated, the Hebbian learning rule would add 1 to the weight of the connection between them (highlighted in the above diagram).

When training is complete, we freeze learning and present the word  $W_{n+1}$  (represented in features) to the bottom layer. What we want to know is the relative activation of the outcome nodes  $X_1, \dots, X_a$ , since this indicates how likely it is that

$o(W_{n+1})$  is to be realized as one of these outcomes. As in all connectionist networks, the activation for each outcome node is derived from the sum of the weights of the connections leading into it from active nodes. The most commonly used connectionist networks today (e.g. feedforward networks trained with backpropagation, Hopfield networks, and so on) take this input sum and then run it through a non-linear function. What makes a linear associator linear is that it does not: the activation of an output node is directly proportional to the sum of the incoming weights. Our activation function is thus as follows:

$$(18) \quad a_i = \sum_j a_j w_{ij},$$

where  $a_i$  and  $a_j$  are the activations of nodes  $i$  and  $j$ , respectively, and  $w_{ij}$  is the weight of the connection from input node  $j$  to output node  $i$ .

Surprisingly, perhaps, this architecture and these equations mean that the activation of outcome node  $X_1$ , relative to all the other node activations, is calculated in precisely the same way as the probability  $P(X_1)$  in the OT model (i.e. in (15)). Since in our example we know that  $W_{n+1}$  has feature value  $[+F_1]$ , the weight of the highlighted connection in (17) between  $[+F_1]$  and  $X_1$  will be added into the activation of node  $X_1$ . This weight itself represents the number of instances in which two things are simultaneously true: a word  $W_i$  has the value  $[+F_1]$  and the outcome  $o(W_i)$  of this word is  $X_1$ . In general, the activation of  $X_i$  will represent the total number of instances such that  $W_{n+1}$  matches some analogical trigger  $W_k$  word in some feature and such that the outcomes for  $W_{n+1}$  and  $W_k$  are both  $X_i$ . (Here is the reappearance of the sentence, in modified form, that I asked the reader to keep in mind earlier.) In other words, the activation of node  $X_i$  is precisely identical to  $|C_{X_i}|$  (i.e. total number of  $X_i$ -favoring constraints in the OT model). This in turn means that the proportion of activations represented by  $X_i$  is given by the formula in (15), which divides  $|C_{X_i}|$  by the sum of all “non-noncommittal” constraints (which is equivalent to the sum of all output activations in the network model). Under the atomicity assumption, then, the OT model of analogy is equivalent to a linear associator.

Linear associators have very well-understood strengths and weaknesses (see Anderson 1995 for discussion). Among their strengths is the fact that they are actually found in the nervous systems of some simple animals, and more to the point here, that they capture the essential properties needed for analogy (namely, the properties described in the previous section, such as gradient similarity effects, gang effects, and frequency effects). Among their weaknesses are technical mathematical limitations that may not be relevant here (e.g. they cannot distinguish nonorthogonal vectors in the training set, and like all two-layer networks they cannot learn exclusive-or or parity), but they also suffer from a problem that makes them less than ideal for the quantitative analysis of analogy: they are overly in-

decisive. Due to the lack of a nonlinear activation function, they tend to waver between states rather than showing crisp categorical behavior. (Categoricity is something of a problem for connectionism in general, but linear associators are absolutely abysmal.)

Before demonstrating these strengths and weaknesses in the next section by pitting the OT model directly against AM, we should first briefly consider what happens if we discard the assumption that outcomes must be atomic. Unlike (standard) AM, the OT model has no problem using featural representations for outcomes. Consider again the general situation, identical to the one we have been examining, but where the set of candidate outcomes consists of all possible representations generated by the features  $F_1, \dots, F_m$ . The analogical constraints will then have the following form, where the second component constraint now refers to just one feature:

$$(19) \text{ IDENT-OO}(W_i, W_{n+1};F_j) \wedge \text{ IDENT-OO}(o(W_i), o(W_{n+1});F_k)$$

These constraints no longer choose at most one candidate as optimal. Instead, they either evaluate all candidates the same (namely, as in the previous discussion, if the words  $W_i$  and  $W_{n+1}$  don't match in feature  $F_j$ ), or they accept some of the candidates (i.e. if  $o(W_i)$  matches  $o(W_{n+1})$  in feature  $F_k$ ) and reject the rest. If such a constraint favors any candidates at all, the number of favored candidates will almost always be greater than one. For example, if there are  $m$  features all with the same valency  $v$ , then each non-noncommittal constraint will favor  $v^{m-1}$  candidates. Tableau (20) illustrates this with three binary features, where  $v^{m-1} = 4$ .

(20)

$W_1 = [+F,+G,+H],$ $W_{n+1} = [+F,+G,-H],$ $o(W_1) = [-F,-G,-H]$	OO $\wedge$ OO- $(W_1, W_{n+1};F)(o(W_1), o(W_{n+1});G)$	...
$o(W_{n+1}) = [+F,+G,+H]$	*	...
$o(W_{n+1}) = [+F,+G,-H]$	*	...
$o(W_{n+1}) = [+F,-G,+H]$		...
$o(W_{n+1}) = [+F,-G,-H]$		...
$o(W_{n+1}) = [-F,+G,+H]$	*	...
$o(W_{n+1}) = [-F,+G,-H]$	*	...
$o(W_{n+1}) = [-F,-G,+H]$		...
$o(W_{n+1}) = [-F,-G,-H]$		...

These considerations show that the versions of Anttila's Theorem we used earlier cannot apply here, since no single constraint can alone be responsible for choosing an output candidate as optimal. However, we still don't have to rank all the constraints every possible way and tally up the results, because a given outcome



candidate  $o(W_{n+1}) = [\alpha_1 F_1, \dots, \alpha_m F_m]$  (where  $\alpha_i$  represent feature values) can still only be chosen as optimal under very well-defined circumstances. Namely, in order for this candidate to be chosen, it must be that for every feature  $F_i$ , at least one constraint favoring  $[\alpha_i F_i]$  must outrank all constraints that favor  $[-\alpha_i F_i]$  (i.e. any other value for this feature); this follows directly from the definition of OT constraints and constraint ranking (see Samek-Lodovici & Prince 1999 for more on the foundational mathematics of OT). If  $|C[\alpha_i F_i]|$  and  $|C[-\alpha_i F_i]|$  represent, respectively, the number of constraints that favor  $[\alpha_i F_i]$  and the number that favor  $[-\alpha_i F_i]$ , we can use the reasoning behind Anttila's Theorem to deduce that the probability that the optimal candidate contains  $[\alpha_i F_i]$  must be as follows:

$$(21) \quad P(o(W_{n+1}) = [\dots \alpha_i F_i \dots]) = |C[\alpha_i F_i]| / [|C[\alpha_i F_i]| + |C[-\alpha_i F_i]|]$$

Now, two constraints that refer to different features (i.e. a constraint that favors  $[\alpha_i F_i]$  and a constraint that favors  $[\alpha_j F_j]$ ) do not interact at all. That is, no matter how they are ranked with respect to each other, the outcome will be the same. In lieu of a formal proof, I offer Tableau (22) to ponder, where the relative ranking of the constraints  $*[+F]$  and  $*[-F]$  (and of  $*[+G]$  and  $*[-G]$ ) does indeed affect which candidate will win, but not the relative ranking of  $*[+F]$  and  $*[+G]$  (nor of  $*[-F]$  and  $*[+G]$ , and so forth). For example, if  $*[+F]$  is ranked above  $*[-F]$  as shown, the first two candidates can never win no matter how  $*[+G]$  and  $*[-G]$  are ranked, even if one or both outranks  $*[+F]$  (try it and see).

(22)

	$*[+F]$	$*[-F]$	$*[+G]$	$*[-G]$
$[+F, +G]$	*		*	
$[+F, -G]$	*			*
$[-F, +G]$		*	*	
$[-F, -G]$		*		*

What this means is that the probability that a given ranking chooses a candidate containing  $[\alpha_i F_i]$  is independent of the probability that this optimal candidate contains  $[\alpha_j F_j]$  as well. In the above tableau, for example, the probability that the optimal candidate contains  $[+F]$  is 1/2 (by the formula in (21), which can also be confirmed by hand), and the probability that it contains  $[+G]$  is also 1/2. Neither fact is dependent on the other in any way. This allows us to apply the multiplication rule from probability theory, deriving the probability  $P([+F, +G]) = 1/2 \cdot 1/2 = 1/4$  (which may be confirmed by examining all 24 possible rankings of the constraints in (22)). In general, the probability that  $o(W_{n+1}) = [\alpha_1 F_1, \dots, \alpha_m F_m]$  is given by the following formula (completing the proof is left as an exercise for the reader):

$$(23) \quad P(o(W_{n+1}) = [\alpha_1 F_1, \dots, \alpha_m F_m]) = \prod_i |C[\alpha_i F_i]| / [|C[\alpha_i F_i]| + |C[-\alpha_i F_i]|]$$

Interpreting this in connectionist terms is more difficult than when we made the atomicity assumption, but it does seem to have some interesting properties. As before, the number  $|C[\alpha_i F_i]|$  can be thought of as the sum of all the connection weights leading into an output node, this time representing the feature value  $[\alpha_i F_i]$ . Now, however, we have something like a nonlinear activation function, or more properly, a function that takes as arguments the activations, for each feature, of the feature value node of the target outcome relative to the activations of the nodes for the other values for that feature. This function may tend to make the model more decisive, since it involves multiplication rather than merely addition, but since the multiplication involves fractions less than or equal to 1, it can only work to decrease activation. Further thought is needed to explore the quantitative implications of this aspect of the model, and I won't discuss this further in this paper. The primary point to note here is that while representing outcomes with features has not been implemented in AM, it poses no special difficulty in the OT model of analogy (at least from the theoretical side).

## 5. Analogy in Optimality Theory and AM

Given the lack of sophistication of the OT model beneath all of its complex notation, one might expect it to perform rather poorly when confronted with actual analogical tasks to carry out. In this section I show that it does indeed perform much worse than AM. Nevertheless, at a higher level of description, the OT model actually performs remarkably well given its generative origins: in virtually every case, it correctly chooses which of the alternative outcomes should be the preferred one. Its weakness lies solely in the degree of probability it assigns to this outcome (always much lower than it should). I end the section by suggesting how the quantitative predictions of the OT model might be improved by borrowing ideas from connectionism, and alternatively, how the model could be made into a notational variant of the AM algorithm, with interesting consequences for AM itself.

Consider first an AM analysis of Finnish past tense *ti* ~ *si* allomorphy (namely the one in Skousen 1992:310–322; more recent AM analyses of this and related problems in Finnish are found in Skousen 1989 [written after Skousen 1992] and in this volume). In this analysis, attention was restricted to a small set of two-syllable verbs ending in tAA (where A represents a low vowel), some of which form the past tense with the *ti* allomorph, some with the *si* allomorph, and some with either (at particular token frequencies of occurrence that vary word by word). Unpacking the description in Skousen 1992, the analysis uses seven contextual variables (i.e. features), listed in the following (with my own labels):

- (24) [ $\pm$ C1]: a binary feature representing the presence or absence of an initial consonant  
 [C1 value]: a multivalued feature representing the onset consonant or the lack thereof  
 [V1 value]: represents the first vowel  
 [ $\pm$ V2]: represents the presence or absence of a second vowel  
 [V2 value]: represents the second vowel or the absence thereof  
 [ $\pm$ C2]: represents the presence or absence of a stem-final consonant  
 [C2 value]: represents the stem-final consonant or lack thereof

Skousen (1992) first gives the model a data set of 42 verbs, and then tests it on verbs not in the data set, including *viertää* ‘to slope’. AM predicts that the probability of choosing the allomorph *ti* for this verb is very low:  $P(vierti) = 0.00153$ . The model therefore both picks up on a real pattern in the data, and is very decisive about its response. The result is so sharp that it appears as if it’s due to a rule. Based on the data given, it is tempting for a linguist (e.g. myself) to state such a rule, namely if the stem is a closed syllable, choose *si*. Nevertheless, the mechanism used here is actually analogy, not a general rule; AM even allows one to list forms by their degree of responsibility for the analogy. Skousen (1992:321) ends his discussion by pointing out that the three factors affecting the strength of the analogy here are (using my terminology) gradient similarity effects between analogical trigger and target, the frequency of the analogical trigger, and (using the original wording) the “extensiveness of the homogeneity”.

As I’ve shown in previous sections, the OT model captures the first two of these three factors, but like connectionism and other non-AM models of analogy, it ignores homogeneity. How far can the OT model get with the same data set, the same features, and the same test word *viertää*? To find out, I calculated the predicted probabilities for the two allomorphs given all possible rankings of all possible analogical conjoined constraints conforming to the Proportion Principle (or equivalently, their relative degree of activation in a linear associator). In practical terms, what I did was as follows. For each data verb input, I counted the number of feature values that matched those in the input *vier*, multiplied this sum by the number of tokens of *ti* and *si* reported for this data verb, and finally added up the totals for *ti* and for *si*. Some of my calculations are shown in Table (25), which also includes the grand totals of the activations for *ti* and *si* given the input *vier*.

(25)

	a	b	c	d	e	f	g	h	i	j	k	l
<i>vier</i>	+C1	/v/	/i/	+V2	/e/	+C2	/r/	a+...+g	No. ti	No. si	h·i ti	h·j si
<i>hoi</i>	1	0	0	1	0	0	0	2	26	0	52	0
<i>i</i>	0	0	1	0	0	0	0	1	2	0	2	0
<i>kiel</i>	1	0	1	1	1	1	0	5	0	22	0	110
<i>kier</i>	1	0	1	1	1	1	1	6	0	16	0	96
...	...	...	...	...	...	...	...	...	...	...	...	...
Total:											954	1475

The predicted probability is thus  $P(vierti) = 954/(954+1475) = 0.39275$ . The fact that this number is less than 0.5 means that the OT model agrees that the preferred form should actually be *vierti*, not *vierti*. That is, under a winner-take-all interpretation, the OT model and AM both choose the correct outcome. However, the probability  $P(vierti)$  predicted by the OT model is of course far higher than the near-zero probability predicted by AM. This may be a consequence of the OT model's linear (i.e. indecisive) nature, or it may be related to its ignoring homogeneity.

Other differences in the behavior of the OT model and AM can be seen if we list the data verb inputs in order by their relative contribution to the analogical effect. In the OT model, ranking is by the number of feature matches weighted by token frequency (i.e. the values listed in the last two columns in (25)). The following Table (26) lists the ten most influential items according to each model (data for AM is from Skousen 1992:320).

(26)

AM			OT		
Verb input	Outcome	Effect	Verb input	Outcome	Effect
kier	si	0.270	pi	ti	748
piir	si	0.258	tie	si	432
kiel	si	0.169	pyy	si	180
siir	si	0.086	piir	si	120
rien	si	0.061	löy	si	112
pyör	si	0.018	kiel	si	110
viil	si	0.009	ve	ti	100
mur	si	0.009	kier	si	96
kiil	si	0.005	myön	si	90
vään	si	0.003	kään	si	78

Examination of this table shows that whatever the ultimate cause, the proximal cause of the OT model's quantitative problems is that it is too easily fooled by

false analogies. The input *pi*, for example, has a large influence merely because it happens to share two features with *vier* (i.e. [+C1] and [V1=/i/]), and because it is high in frequency. That’s too bad for the model, since the outcome for *pi* is *ti*, not the desired *si*. In spite of such problems, the OT model does manage to include in the top ten some of the items that AM also considers important, namely the inputs *piir*, *kiel*, and *kier*. Moreover, two other items ranked highly by the OT model (but not by AM) have the “correct” syllable structure according to the linguistic analysis, namely the closed-syllable inputs *myön* and *kään* (AM instead lists *rien*, *pyör*, *viil*, *mur* and *vään*). This means that fully half of the ten most influential data points for the OT model are precisely the ones that should have the most influence. Given the extreme simplicity of the OT model, this is a rather remarkable achievement.

The unimpressive level of quantitative accuracy leaves room for improvement, of course. One way to improve it is to hand-pick the features in accordance with the linguistic analysis (mentioned above) that states that the crucial factor is syllable structure alone. This means that we ignore all features except [ $\pm$ C1], [ $\pm$ V2] and [ $\pm$ C2]. Carrying out the procedure, we end up with the predicted probability  $P(\text{vierti}) = 0.07368$ , which is far closer to zero, as desired. We might be able to justify this move if it were a cross-linguistic universal that suffix allomorphy is always sensitive only to syllable structure, thus implying an innate cognitive bias for some features over others. No such universal exists, however, and this move should rightly be dismissed as outright cheating.

To help learn how the accuracy of the OT model may be improved in more appropriate ways, I considered another simple example comparing the OT model with AM (and also with standard connectionism). This is the toy problem described in Baayen 1995:395 (based on an example in Skousen 1992:266–272) in which there are 22 data points composed of three features (which I will label  $F_1$ ,  $F_2$ , and  $F_3$ ). Each feature can take one of four values (0, 1, 2, 3), so the data points can be represented as strings like 002 or 332. There are two possible outcomes (A or B) which are purposely chosen to conform to a simple rule: if  $F_2 \in \{0, 1\}$ , then the outcome is A, otherwise it is B. The existence of this rule can be seen by the regular distribution of A’s and B’s in the following Table (27).

(27)

$F_1 \rightarrow$	0				1				2				3			
$F_2 \rightarrow$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$F_3 \downarrow$																
0			B	B		A	B			A		B	A			
1		A		B					A			B				B
2	A		B						A	A					B	B
3					A					A		B		A		

Given these three four-valued features, there are 42 other possible data points (the empty cells in (27)). AM does not give a strictly categorical response in most of these cases. For data point 000, for instance, it predicts the probability  $P(A) = 0.871$ , rather than  $P(A) = 1$  as required by the rule. (Of course, this begs the question of how easily human beings could also see this particular pattern as rule-governed.) Nevertheless, AM is never wrong about which outcome should be more probable. More importantly, its error rate is very low. This can be calculated with a number of methods; I used two. In the first method, I took all the data points for which the rule predicts  $P(A) = 1$  and subtracted from 1 the actual probability provided by AM (e.g. 0.871 in the above case). The mean error, calculated this way, was a quite respectably low 0.057 (chance performance of course would be 0.5). As another measure of error rate, for every test point I calculated the Euclidean distance (commonly used in studying connectionist models) between AM's predicted values for  $P(A)$  and  $P(B)$  and the correct values. For example, for data point 000, the correct probabilities for A and B respectively are (1, 0). AM predicted (0.871, 0.129). The Euclidean distance between these two points is 0.182. Here chance performance would be half the length of the diagonal of a unit square (i.e. 0.707); AM's mean error value was the still very low 0.081.

How does the OT model fare on the same data? Again, it depends on how you look at it. Table (28) shows the OT model's predictions for preferred outcome for the 42 test points (the original data points are shaded). The model only made one mistake, incorrectly claiming that  $P(223A) = P(223B)$ . Given not just the simplicity of the OT model but also the sparse and scattershot evidence for the AB rule, I suggest that this should count this as another success.

(28)

$F_1 \rightarrow$	0				1				2				3			
$F_2 \rightarrow$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$F_3 \downarrow$																
0	A	A	B	B	A	A	B	B	A	A	B	B	A	A	B	B
1	A	A	B	B	A	A	B	B	A	A	B	B	A	A	B	B
2	A	A	B	B	A	A	B	B	A	A	B	B	A	A	B	B
3	A	A	B	B	A	A	B	B	A	A	A/B	B	A	A	B	B

A closer look at the results reveals the usual quantitative problems, however. Using the same measures of accuracy applied to AM, the mean error rate shown by the OT model's predictions of  $P(A)$  was 0.343 (chance = 0.500), and the mean error rate by Euclidean distance was 0.486 (chance = 0.707). Both results are statistically better than chance, but the error rate is still far higher than that for AM.

Earlier I mentioned two possible causes for the quantitative shortcomings of the OT model: its lack of a nonlinear activation function and its ignoring of ho-

mogeneity. To better understand which is responsible in this case, I compared the behavior of the OT model with that of a two-layer connectionist network which does have a nonlinear activation function, but as with connectionism in general is not particularly sensitive to homogeneity. Like the linear associator associated with the OT model, the output layer of this connectionist network consisted of just two nodes (one for each outcome A and B) and three sets of four input nodes (for the three four-valued features). The difference was that the network used a sigmoid (i.e. S-shaped) activation function and was trained using the backpropagation learning algorithm; since it only had two layers, this made it essentially equivalent to a perceptron (again I recommend Anderson 1995 for lucid discussion of these concepts). This model (simulated using the Tlearn software; see Plunkett & Elman 1997) had absolutely no trouble learning the AB rule. I assumed that given any particular input, the activation of an output node (always between 0 and 1) represented the degree of probability of the model assigning that outcome given that input (a commonly made interpretation in the connectionist literature). Its error rate for P(A) was 0.038 and by Euclidean distance 0.085, roughly as low as for AM. It appears, then, that at least for this particular simple problem, the accuracy of the OT model might be improved simply by giving it a nonlinear activation function.

How could this be accomplished? The simplest nonlinear activation function used in connectionist models is a step function. This is a function that has some constant value (say 0) for all inputs below some threshold, and some other constant value (say 1) for all inputs above the threshold. Unfortunately, this idea is doomed from the start, since the output node activations will now always be just 0 or 1, which makes it impossible to interpret them as continuously varying probabilities.

The particular sigmoid function used with backpropagation and other connectionist models is unlikely to be coaxed from the simple mathematics underlying the OT model. Continuous nonlinear effects may arise if outcomes are represented with features, as discussed at the end of the previous section, but this can't help us with the Finnish and *e-r* problems examined here, which use atomic outcomes. Moreover, any other attempt to create a continuous sigmoid activation function must face the problem of where to locate the flexion point (i.e. the threshold). In most connectionist models, this point is located where the input is 0, but this can't work for the OT model as it currently stands. In the linear associator associated with the model, the input activation nodes and the connection weights are always nonnegative, making it impossible to have a negative sum feeding into the output node activation function. Thus if we maintain the general structure of the OT model, the location of the threshold must somehow be made to depend on the size of the training set (since connection weights increase arithmetically as more items are trained).

An alternative way to derive a continuous nonlinear activation function might be to posit "evil twins" for the IDENT-OO constraints, that is constraints like DIFF-

$OO(W_i, W_j; F_k)$  that require words  $W_i$  and  $W_j$  to have *different* values for feature  $F_k$ . Although there is no precedent for such constraints in the OT literature, by including measures of difference we would make the model more consistent with theories of comparison in the cognitive science literature (e.g. Tversky 1977). It is also possible that DIFF-OO constraints could be made to interact with the IDENT-OO constraints in such a way as to create the equivalent of a sigmoid activation function. This is because there is only one way that two words  $W_i$  and  $W_j$  can be completely identical, and only one way they can be completely different (assuming binary features), but there are many ways that they can be partially similar and partially different. The resulting binomial distribution is an approximation of the normal distribution, which in turn approximates the first derivative of the sigmoid function commonly used in connectionist modeling. Unfortunately, exploring this intriguing possibility would take us far beyond the scope of this paper.

I showed above that connectionism and AM performed equally well in the *e-r* problem. Why not set aside nonlinearity and instead try to incorporate AM's analysis of homogeneity into the OT model? Given the connectionist-like nature of the OT model, this is, unsurprisingly, rather difficult to conceptualize. AM measures homogeneity by means of overlapping sets and subsets of forms, a device that has no obvious parallel in connectionism or the OT model. This makes it difficult to work out a detailed strategy for making OT work like AM without doing undo violence to its inner OT-nature (and thus possibly alienating the generative linguists whom I hope to count among my audience).

However, a compromise can easily be reached between OT and AM, albeit at a rather superficial level. The final step of the standard AM algorithm, after determining the homogeneous supracontexts and sets of data points (outcome pairs) with their associated pointers, is the random selection rule of usage. Since some pointers point to one outcome, others to another, and so on, the rule of usage predicts relative probabilities that are directly proportional to the number of pointers. If one were to write a formula for this, it would appear as follows:

$$(29) \quad P(X_1) = |p_{X_1}| / [ |p_{X_1}| + |p_{X_2}| \dots + |p_{X_a}| ],$$

where  $|p_{X_i}|$  represents the number of pointers pointing to outcome  $X_i$ .

This is of course identical to the formula given earlier in (15) for the probabilities predicted by the OT model (assuming atomic outcomes). This means that if the AM algorithm is used to generate the proper number of analogical conjoined constraints, OT can be used to generate the probabilities. For example, consider the example given in Skousen 1989:22–37, which predicts the probabilities of the outcomes *e* versus *r* for the context 312, given a set of five data points built of three four-valued features. For each pointer in each homogeneous supracontext



generated by the AM algorithm (Skousen 1989:36), we posit an OT-like conjoined constraint as given below ( $W_t$  represents the target word, here 312):

$$(30) \text{ POINT-OO}(W_i, W_j) \wedge \text{IDENT-OO}(o(W_j), o(W_t))$$

The first component constraint requires that  $W_i$  point to  $W_j$  (e.g. 310 points to itself, or 032 points to 212) and the second component requires that the outcomes for the analogical trigger (i.e. the item that the pointer points to) and the target 312 must be completely identical. The first component thus serves as a counting mechanism; the total number of the constraints forcing identity between  $o(W_j)$  and  $o(312)$  will simply be the number of pointers pointing to  $W_j$ , just as in the AM algorithm. That is, if the first component constraint is violated (i.e. there is no such pointer), the conjoined constraint will be violated by every candidate output and so can be ignored (in accordance with Noncommittal Constraint Irrelevance).

Tableau (31) shows these constraints in action. To save space, I've left out all noncommittal constraints (i.e. those where there is no pointer). A simple application of Anttila's Theorem results in the predicted probabilities  $P(312e) = 4/13$  and  $P(312r) = 9/13$ , precisely as in standard AM (unsurprisingly).

(31)

	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$	$o(312)$
	=	=	=	=	=	=	=	=	=	=	=	=	=	=
	$o(310)$	$o(310)$	$o(310)$	$o(310)$	$o(311)$	$o(311)$	$o(311)$	$o(311)$	$o(212)$	$o(212)$	$o(212)$	$o(032)$	$o(032)$	
312					*	*	*	*	*	*	*	*	*	*
→ e														
312	*	*	*	*										
→ r														

To make this analysis more palatable to a generativist comfortable with OT, we would need to unpack the constraint POINT. There's no avoiding a full-fledged AM analysis eventually (nor should we necessarily want to, of course), but we might be able to put it off somewhat if we let AM provide us just with the homogeneous supracontexts, and within each, the number of outcomes of each type. Once we know the size  $s$  of each supracontext and the number  $n$  of outcomes in it of some type, the number of pointers for this outcome in this supracontext is just  $s \cdot n$ . The total number of pointers pointing to some outcome (i.e. the activation of the outcome node in the OT linear associator) is thus  $\sum_i s_i n_i$ . This is the sort of simple mathematics that the OT model could perhaps accommodate, but again exploring this in detail would take us too far afield.

For readers more familiar with AM than with OT, the last part of the above discussion may seem like a trivial parlor trick, but I think there are serious reasons for considering it. First, it represents a perhaps surprising point of contact between two historically different approaches to language, namely those provided by gener-

ative linguistics and by AM. The first step towards cooperation is communication, and I hope that by recognizing this point of contact, scholars of different stripes can learn to use a shared formal language to exchange insights and data. Second, in the previous section I pointed out that it seems difficult to imagine how AM could be modified so that it allows outcomes to be represented with features. Translating the last steps of AM into OT notation serves as a useful aid to the imagination. In fact, all one has to do is modify the constraint in (30) to that in (32), where the second component constraint now refers to a specific feature.

$$(32) \text{ POINT-OO}(W_i, W_j) \wedge \text{IDENT-OO}(o(W_j), o(W_t); F_k)$$

Making this change would require replacing the standard random rule of usage with the following probability rule (based on the formula in (23)):

$$(33) P(o(W_t) = [\alpha_1 F_1, \dots, \alpha_m F_m]) = \prod_i |p[\alpha_i F_i]| / [ |p[\alpha_i F_i]| + |p[\neg \alpha_i F_i]| ],$$

where  $|p[\alpha_i F_i]|$  represents the number of pointers pointing to an outcome containing feature value  $[\alpha_i F_i]$ , and  $|p[\neg \alpha_i F_i]|$  represents the number of pointers pointing to an outcome containing some other feature value for  $[F_i]$ .

Again, the consequences of this suggestion are not entirely clear at this point and would require much more thinking than I have space here to work through. I hope, however, that this suggestion sparks some productive thoughts in the reader's mind as well.

## 6. Beyond analogy

At the end of the preceding section I pointed out one possible way in which the OT model may inspire researchers working on AM. In this section I describe another, namely the ability of the OT model to accommodate certain kinds of non-analogical factors that conceivably do play a role in human language. After all, the OT model of analogy described so far only uses one of the two basic types of OT constraints, namely Faithfulness constraints (which I suppose could also include POINT). What about Structure constraints, which require forms to meet universal standards? To a generative linguist, it seems rather foolhardy to claim that all of language can be handled by analogy alone. There are a number of reasons for this, the simplest being that analogy can only work to breathe psychological life into a pattern if there is already something of a pattern there to start with. But where do linguistic patterns come from in the first place? The traditional answer in generative linguistics has been that they come from what OT now calls Structure constraints.

There is another answer, of course: history and physics (or more generally, any set of systematic forces working beyond the confines of a single human brain).

Over the past few decades, there has been growing acknowledgement of this alternative answer in generative circles, and some work in OT has used Structure constraints that are explicitly physical in nature (e.g. Flemming 1995; Hayes 1999a; Jun 1995; Silverman 1996; Kirchner 1997; Myers 1997). There's something vaguely disturbing about this, though. Structure constraints are supposed to be (innate) psychological things, so why should they mirror physical forces so exactly? Moreover, the generativists have never really managed to come up with a convincing reply to critics who suggest that aspects of language (e.g. word-level phonology) are systematic simply because people memorize things that have been molded by extramentalist forces over generations of speakers. For example, while it may be true that the [k]~[s] alternation in *electric-electricity* is phonetically natural in some sense, surely this naturalness plays absolutely no role in the minds of modern-day speakers, whose minds are instead occupied with maintaining this pattern through analogy (to the extent that this pattern is psychologically active at all, of course). This modular approach to phonology, where separate subtheories handle the ontogenesis (e.g. physics) and spread (e.g. analogy) of phonological patterns, is completely compatible with the AM program, I think.

Nevertheless, there do seem to be cases where linguistic patterns arise within the minds of speakers, and possibly within the same environment as the mental operations that process analogy. As a case in point, consider patterns that appear to be motivated by innate restrictions on the access, retrieval, and storage of phonological forms. An example of such a pattern is dissimilation. While phonological assimilation can be understood as the semi-fossilization of coarticulation, and hence not fundamentally a psychological phenomenon, dissimilation does not arise through the operation of purely physical forces. Instead, as Ohala (1986) has argued, it requires that listeners in some sense mentally undo perceived coarticulations; when they overshoot, the result is a dissimilation. The generative linguist Kiparsky (1986) endorses this analysis of dissimilation, since it helps explain why dissimilation rules are always lexicalized to some extent and never completely automatic. The natural phonologists Donegan and Stampe (1979) also treat dissimilation as less than purely physical, counting it among fortitions (as opposed to lenitions), which have a perceptual (i.e. psychological) rather than articulatory (i.e. physical) teleology. Taken together, these disparate observers all seem to agree that dissimilations arise not in the outside physical world, but in the mental lexicon.

But this is precisely where analogy occurs as well. No theory of analogy can work without a set of memorized exemplars to analogize from, and Skousen (1989, 1992) even makes memory (and its imperfections) an explicit part of the overall AM approach. In OT terms, this means that there is no theoretical problem with mixing analogical Faithfulness constraints together with Structure constraints, as long as these Structure constraints are motivated by lexical processing rather than physics.

To see how the OT model would do this, I would like to examine a dissimilation pattern first analyzed by Phillips (1981, 1984, 1994) (see also Myers 2000a for a briefer discussion of the same pattern). What makes this pattern particularly interesting is that Phillips (1984) uses it to call attention to an empirical corollary of the above discussion: phonological patterns that are lexically motivated tend to target lower-frequency forms first. The more frequently a form is accessed from memory, the more efficiently it is accessed, and an efficient memory is an accurate one. Hence we do not expect lexical factors to target higher frequency forms. Lower frequency forms, being harder to access, are more subject to whatever plausible patterns the memory mechanisms may use to fill in the forgotten holes. This is why analogies tend to affect lower-frequency forms more readily than higher-frequency forms, as I discussed earlier.

With this as background, now consider the pattern. Phillips (1981, 1984, 1994) describes a variable rule in Georgian English whereby the historically older /y/ is optionally deleted after alveolars (including /n/, /d/, and /t/). Crucial for the discussion here is that the probability of this occurring is inversely correlated with the frequency of the word. Table (34) lists an example from each of the five frequency classes that Phillips considers, along with the mean token frequency of each class and the mean probability of y-deletion.

(34)

Example	Mean frequency (tokens)	Probability of y-deletion
new	997.290	0.430
knew	358.380	0.545
numeral	30.290	0.601
neutral	3.594	0.718
nude	0.438	0.744

As Phillips (1984) points out, this is precisely the opposite of the pattern found with phonetically-motivated phonology, which shows positive frequency effects in rate of application or rate of lexical diffusion. For example, the optional dropping of /t/ in words like *mist* during fluent speech, which apparently has an articulatory origin (see e.g. Browman & Goldstein 1990), occurs more often in higher-frequency words than in lower-frequency words (Myers & Guy 1997; Bybee 2000). Other examples of phonetically-motivated phenomena that occur more readily in higher-frequency forms are described in Fidelholtz 1975, Phillips 1984, Kaisse 1985, Hammond 1997, and Bybee 2000, among many other places. As Phillips (1984) and Bybee (2000) have argued, such positive frequency effects are best understood as resulting mainly from physics, not lexical processing. (Metaphorically speaking, passing words back and forth through the air tends to wear them out.) What an analogical model should be able to do, then, is collaborate with lexical factors to create the negative frequency effect seen in Georgian English, but be in-

capable of interacting directly with the physical forces giving rise to positive frequency effects (such patterns should instead be ascribed to a separate module in the more general theory of phonology).

The OT model meets these criteria. As described in an earlier section, it is capable of describing the fact that lower frequency words make better analogical targets. This ability can also be used to describe the fact that lower frequency words in Georgian English are more likely to show y-deletion. Since y-deletion involves dissimilation, which I argued above is lexically motivated, we may in good conscience posit a Structure constraint to handle it (below I address the question of whether this is in fact necessary rather than merely permissible). For the sake of simplicity, we can use the following constraint (which assumes that alveolars and /y/ are both coronal, i.e. their articulation crucially involves the tongue blade):

(35) \*COR-COR: Two adjacent coronals are disallowed (e.g. \*[ny]).

Being lexically motivated, this sort of Structure constraint may freely interact with analogical Faithfulness constraints, since both originate in the processes of lexical storage and retrieval. This allows for tableaux like (36a, b), which include both this Structure constraint and a set of Faithfulness constraints that are parochial by tokens. Since the phenomenon is variable within a single dialect, I follow the OT literature on variable phonology (alluded to earlier) and assume that the constraints are freely ranked.

(36a) (1000 of these)

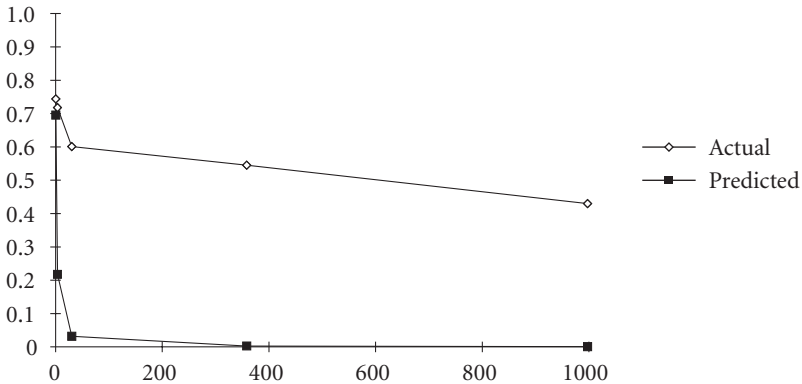
[nyu]	*COR-COR	IO- <i>new</i>	IO- <i>new</i>	...
[nyu]	*			...
[nu]		*	*	...

(36b) (1 of these)

[nyud]	*COR-COR	IO- <i>nude</i>
[nyud]	*	
[nud]		*

Applying Anttila’s Theorem, we derive the probabilities  $P([nu]) = 0.001$  and  $P([nud]) = 0.500$ . While as usual the OT model doesn’t provide us with particularly accurate numbers, it does capture the essential observation: lower frequency means a higher rate of y-deletion. Both of these points are driven home in the following graph (37), which shows how far off the OT model is in the specifics while nevertheless resulting in a curve that curves in the correct direction (here I’ve adjusted the token frequencies to get the best possible values for the OT model at the low end of the frequency distribution).

(37)



In spite of its quantitative inaccuracies, this general approach to structural factors in phonology is satisfying in an important respect: while it can in principle handle negative frequency effects like that just described, it is entirely incapable of handling positive frequency effects (e.g. with t-deletion). If a phonetically motivated Structure constraint were put into an analysis like that sketched in (36), we would falsely predict the frequency effect to be negative as well. This principled weakness is just what we want in an analogical model, which by its very nature is not phonetically motivated. Instead, as noted above, such cases require a separate module of the theory, one independent of analogy.

Nevertheless, I have not made the case that a pattern like y-deletion is necessarily due to a Structure constraint. Surely it would be more parsimonious if an analogical model were always incapable of referring to phonetic entities (like the tongue blade referred to by \*COR-COR), regardless of the motivation of the constraints involved. In particular, is it really so inconceivable that y-deletion could itself be due to some lexical process, perhaps even analogy, rather than to a specific constraint of the grammar? Clearly it is conceivable, since Dilworth Parkinson (personal communication) has suggested just such a thing, showing me how AM could derive y-deletion by analogy with higher-frequency words that lack a historical /y/ (e.g. *noon*). This analysis then predicts the negative frequency effects we want, just as with analogy generally. A version of this approach is even possible in the OT model, merely by positing a set of constraints that require *new* and *nude* to share the y-lessness feature with words like *noon*. In any case, frequency effects other than the usual positive one are likely have more than one simple cause. For example, after looking at the same Georgian dialect data, Bybee (2000) came up with a rather different explanation, namely that y-deletion is actually due to borrowing or accommodation to the standard dialect; speakers treat less familiar

words less conservatively, allowing them to be replaced with the invading pronunciations. Moreover, in a study of an on-going lexical diffusion in Montreal French, Yaeger-Dror and Kemp (1992) have even discovered a case where frequency appears to play no role at all. Instead the diffusion is affected by semantics (of a curious sort): words keep the older pronunciation if they refer to the “good old days.”

Regardless of the final verdict on such cases, I want to leave the reader with a more general lesson: the OT model may represent a case study in how to build a formal model in which analogy can directly interact with (certain) non-analogical factors (i.e. lexically motivated Structure constraints). Whether or not this is ultimately desirable is an empirical issue, but in the meantime it does seem useful for two reasons. First, generative phonologists have always preferred theories that put the extragrammatical motivations explicitly into the grammar. I think one strategy to help generativists move beyond such theories (which I feel are misguided) may be to get them to examine a model in which extragrammatical motivations (i.e. Structure constraints) are not forbidden a priori, but which predicts that they will behave in very narrowly prescribed ways. Second, research in AM has tended to dismiss too quickly one of the generativist’s primary criticisms of analogy, which is that it cannot explain how systematic linguistic patterns arise in the first place. Cases like *y*-deletion should be collected and carefully examined to determine whether they can all be reanalyzed from a purely analogical perspective, and if not, whether at least some non-analogical principles of grammar do exist. If such principles are found, something like the OT model described in this paper may help in accommodating them within a mostly analogical formalism.

## 7. Conclusions

Things are occurring in the Optimality Theory research community that should be of great interest to all those studying analogy. No longer is analogy forbidden in generative linguistics, since there are now widely accepted formal devices that are capable of capturing its essential nature (i.e. exemplar-driven constraints enforcing similarity). My own model may or may not represent a step in the development of a quantitatively successful hybrid between generative and nongenerative approaches to analogy, but it still seems to me to be a rather remarkable fact that something equivalent to a connectionist network can actually be built out of nothing but notions already current in the OT literature. At the very least, I hope that my model inspires generative linguists to learn more about other explicit models of analogy (especially AM, which deserves far more attention among linguists than it has received). At the same time, I think that researchers in AM and other nongenerative models have something to learn from OT formalism as well, in particular its use of

features and its ability to integrate analogy with non-analogical factors. Analogy is one of the central facts of human language, but it's unlikely to be fully understood without the collaboration of many scholars with different backgrounds and areas of expertise. Perhaps we're now witnessing the beginnings of this collaboration.

## References

- (References marked [ROA] are available from the Rutgers Optimality Archive <<http://roa.rutgers.edu>>.)
- Albright, Adam (2002). The lexical bases of morphological well-formedness. In S. Benjamins, W. Dressler, O. E. Pfeiffer, & M. D. Voekova (Eds.), *Morphology 2000* (pp. 5–16). Amsterdam: John Benjamins.
- Alderete, John (1999). Morphologically governed accent in optimality theory. Ph.D. dissertation, University of Massachusetts. [ROA]
- Anderson, James A. (1995). *An introduction to neural networks*. Cambridge, MA: MIT Press.
- Anttila, Arto (1997). Deriving variation from grammar. In F. Hinskens, R. Van Hout, & W. L. Wetzels (Eds.), *Variation, change and phonological theory* (pp. 35–68). Amsterdam: John Benjamins.
- Anttila, Arto, & Young-mee Cho (1998). Variation and change in optimality theory. *Lingua*, 104, 31–56.
- Baayen, R. Harald (1995). Review of *Analogy and structure*. *Language*, 71, 390–396.
- Balari, Sergio, Rafael Marín, & Teresa Vallverdú (2000). Implicational constraints, defaults and markedness. Manuscript, Universitat Autònoma de Barcelona. [ROA]
- Benua, Laura (1995). Identity effects in morphological truncation. In J. Beckman, L. Walsh Dickey, & S. Urbanczyk (Eds.), *University of Massachusetts Occasional Papers in Linguistics*, 18, 77–136.
- Benua, Laura (1997a). Affix classes are defined by Faithfulness. *University of Maryland Working Papers in Linguistics*, 5, 1–26.
- Benua, Laura (1997b). Transderivational identity: phonological relations between words. Ph.D. dissertation, University of Massachusetts.
- Boersma, Paul (1998). Functional phonology: formalizing the interactions between articulatory and perceptual drives. Doctoral dissertation, University of Amsterdam. The Hague: Holland Academic Graphics.
- Boersma, Paul, & Bruce Hayes (2001). Empirical tests of the gradual learning algorithm. *Linguistic Inquiry*, 32, 45–86.
- Booij, Geert (1997). Non-derivational phonology meets lexical phonology. In I. Roca (Ed.), *Derivations and constraints in phonology* (pp. 261–288). Oxford: Clarendon Press.
- Browman, Catherine P., & Louis Goldstein (1990). Tiers in articulatory phonology, with some implications for casual speech. In J. Kingston & M. E. Beckman (Eds.), *Papers in laboratory phonology 1: Between the grammar and physics of speech* (pp. 341–376). Cambridge: Cambridge University Press.
- Burzio, Luigi (1997a). Strength in numbers. In V. Miglio & B. Morén (Eds.), *University of Maryland Working Papers in Linguistics*, 5, 27–52.



- Burzio, Luigi (1997b). Surface constraints versus underlying representation. In J. Durand & B. Laks (Eds.), *Current trends in phonology: Models and methods* (pp. 97–122). Salford: University of Salford.
- Burzio, Luigi (1999). Surface-to-surface morphology: when your representations turn into constraints. Manuscript, Johns Hopkins University Department of Cognitive Science. Presented at the 1999 Maryland Mayfest, University of Maryland, College Park. [ROA]
- Burzio, Luigi (2000). Cycles, non-derived environment blocking, and correspondence. In J. Dekkers, F. van der Leeuw, & J. van de Weijer (Eds.), *Optimality theory: Phonology, syntax and acquisition* (pp. 47–87). Oxford: Oxford University Press.
- Burzio, Luigi (2002). Missing players: phonology and the past-tense debate. *Lingua*, 112, 157–199.
- Bybee, Joan L. (2000). The phonology of the lexicon: evidence from lexical diffusion. In M. Barlow and S. Kemmer (Eds.), *Usage-based models of language* (pp. 65–85). Stanford: CSLI.
- Crowhurst, Megan, & Mark Hewitt (1997). Boolean operations and constraint interactions in optimality theory. Manuscript, University of North Carolina at Chapel Hill and Brandeis University. [ROA]
- Derwing, Bruce L., & Royal Skousen (1994). Productivity and the English past tense: testing Skousen's analogy model. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 193–218). Amsterdam: John Benjamins.
- Donegan, Patricia J., & David Stampe (1979). The study of natural phonology. In D. A. Dinnsen (Ed.), *Current approaches to phonological theory* (pp. 126–173). Bloomington: Indiana University Press.
- Durand, Jacque, & Bernard Laks (Eds.) (1997). *Current trends in phonology: models and methods*. Salford: University of Salford.
- Fidelholtz, James L. (1975). Word frequency and vowel reduction in English. *Chicago Linguistics Society*, 11, 200–213.
- Flemming, Edward (1995). Auditory representations in phonology. Ph.D. dissertation, UCLA.
- Golston, Chris (1996). Direct optimality theory: representation as pure markedness. *Language*, 72, 713–748.
- Green, Antony D. (2001). The tense-lax distinction in English vowels and the role of parochial and analogical constraints. Manuscript, University of Rotterdam. [ROA]
- Hale, Mark, Madelyn Kissonock, & Charles Reiss (1998). What is output in OT phonology? In *Proceedings of WCCFL XVI* (pp. 223–236). Stanford: CSLI Publications.
- Hammond, Michael (1995). There is no lexicon! Manuscript, University of Arizona. [ROA]
- Hammond, Michael (1997). Lexical frequency and rhythm. Manuscript, University of Arizona. [ROA]
- Hayes, Bruce (1999a). Phonetically-driven phonology: the role of optimality theory and inductive grounding. In M. Darnell, E. Mosorvscik, M. Noonan, F. Newmeyer, & K. Wheatly (Eds.), *Functionalism and formalism in linguistics, Volume 1: General papers* (pp. 243–285). Amsterdam: John Benjamins.
- Hayes, Bruce (1999b). On the richness of paradigms, and the insufficiency of underlying representations in accounting for them. Manuscript, UCLA. <[www.humnet.ucla.edu/humnet/linguistics/people/hayes](http://www.humnet.ucla.edu/humnet/linguistics/people/hayes)>

- Hayes, Bruce, & Margaret MacEachern (1998). Folk verse form in English. *Language*, 74, 473–507.
- Jun, Jongho (1995). Perceptual and articulatory factors in place assimilation: an optimality theoretic approach. Ph.D. dissertation, UCLA.
- Kaisse, Ellen M. (1985). *Connected speech: the interaction of syntax and phonology*. Orlando: Academic Press.
- Kenstowicz, Michael (1995). Cyclic vs. non-cyclic constraint evaluation. *Phonology*, 12, 397–436.
- Kenstowicz, Michael (1997). Base-identity and uniform exponence: alternatives to cyclicity. In J. Durand & B. Laks (Eds.), *Current trends in phonology: models and methods* (pp. 363–394). Salford: University of Salford.
- Kiparsky, Paul (1978). Analogical change as a problem for linguistic theory. Reprinted in P. Kiparsky (Ed.), *Explanation in phonology* (pp. 217–236). Dordrecht: Foris Publications, 1982.
- Kiparsky, Paul (1986). Commentary on Ohala 1986. In J. S. Perkell & D. H. Klatt (Eds.), *Invariance and variability in speech processes* (pp. 400–401). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kiparsky, Paul (1988). Phonological change. In F. Newmeyer (Ed.), *Cambridge survey of linguistics, Vol. I* (pp. 363–410). Cambridge: Cambridge University Press.
- Kiparsky, Paul (1993). Variable rules. Handout for Rutgers Optimality Workshop 1.
- Kirchner, Robert (1997). Contrastiveness and faithfulness. *Phonology*, 14, 83–111.
- Kirchner, Robert (1999). Preliminary thoughts on ‘phonologization’ within an exemplar-based speech processing system. Manuscript, University of Alberta. [ROA]
- McCarthy, John, & Alan Prince (1993a). *Prosodic morphology I: constraint interaction and satisfaction*. MIT Press.
- McCarthy, John, & Alan Prince (1993b). Generalized alignment. In G. Booij & J. van Marle (Eds.), *Yearbook of morphology* (pp. 79–153). Dordrecht: Kluwer Academic Publishers.
- McCarthy, John, & Alan Prince (1995). Faithfulness and reduplicative identity. In J. Beckman, L. Walsh Dickey, & S. Urbanczyk (Eds.), *University of Massachusetts Occasional Papers in Linguistics*, 18, 249–384.
- Myers, James (1997). Canadian raising and the representation of gradient timing relations. *Studies in the Linguistic Sciences*, 27, 169–184.
- Myers, James, & Gregory R. Guy (1997). Frequency effects in variable lexical phonology. *University of Pennsylvania Working Papers in Linguistics*, 4, 215–228.
- Myers, James (2000a). Analogy and optimality. Manuscript, National Chung Cheng University.
- Myers, James (2000b). Variable constraint ranking in optimality theory. Manuscript, National Chung Cheng University.
- Nagy, Naomi, & Bill Reynolds (1997). Optimality theory and word-final deletion in Faetar. *Language Variation and Change*, 9, 37–55.
- Ohala, John J. (1986). Phonological evidence for top-down processing in speech production. In J. S. Perkell & D. H. Klatt (Eds.), *Invariance and variability in speech processes* (pp. 386–397, 401). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Perkell, Joseph S., & Dennis H. Klatt (Eds.) (1986). *Invariance and variability in speech processes*. Hillsdale, NJ: Lawrence Erlbaum Associates.

- Phillips, Betty (1981). Lexical diffusion and Southern *tune, duke, news*. *American Speech*, 56, 72–78.
- Phillips, Betty (1984). Word frequency and the actuation of sound change. *Language*, 45, 9–25.
- Phillips, Betty (1994). Southern English glide deletion revisited. *American Speech*, 69, 115–127.
- Plunkett, Kim, & Jeffrey L. Elman (1997). *Exercises in rethinking innateness: a handbook for connectionist simulations*. Cambridge, MA: MIT Press.
- Prince, Alan, & Paul Smolensky (1993). Optimality theory: constraint interaction in generative grammar. Rutgers University Cognitive Science Center.
- Prince, Alan, & Paul Smolensky (1997). Optimality: from neural networks to universal grammar. *Science*, 275, 1604–1610.
- Russell, Kevin (1995). Morphemes and candidates. Manuscript, University of Manitoba. [ROA]
- Russell, Kevin (1999). MOT: Sketch of an OT approach to morphology. Manuscript, University of Manitoba. [ROA]
- Samek-Lodovici, Vieri, & Alan Prince (1999). Optima. Manuscript, University College, London, and Rutgers University. [ROA]
- Silverman, Daniel (1996). Voiceless nasals in auditory phonology. *Proceedings of the Berkeley Linguistic Society*, 22, 364–374.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Smolensky, Paul (1995). On the internal structure of the constraint component *Con* of UG. Paper presented at UCLA. [ROA]
- Steriade, Donca (2000). Paradigm uniformity and the phonetics-phonology boundary. In M. B. Broe & J. B. Pierrehumbert (Eds.), *Papers in laboratory phonology V: Acquisition and the lexicon* (pp. 313–334). Cambridge: Cambridge University Press.
- Steriade, Donca (1999a). Lexical conservatism in French adjectival liaison. In B. Bullock, M. Authier, & L. Reed (Eds.), *Formal perspectives in Romance linguistics* (pp. 243–270). Amsterdam: John Benjamins.
- Steriade, Donca (1999b). Lexical conservatism and the notion *base of affixation*. Manuscript, UCLA. <[www.linguistics.ucla.edu/people/steriade/steriade.htm](http://www.linguistics.ucla.edu/people/steriade/steriade.htm)>
- Tversky, Amos (1977). Features of similarity. *Psychological Review*, 8, 327–352.
- Yaeger-Dror, Malcah, & William Kemp (1992). Lexical classes in Montreal French: the case of (ɛ:). *Language and Speech*, 35, 251–293.

## CHAPTER 12

# The hope for analogous categories

Christer Johansson

### 1. Introduction

Syntactic and semantic categories may play a major role in language acquisition and the internal structuring of language. If we have no prior knowledge of what those categories are, we would hardly recognize them when we see them (a problem known as Meno's dilemma). Linguists therefore often assume linguistic categories to be innate. The radical alternative would be to say that there are no fixed categories, but a process that gives results as if it worked using underlying categories.

In analogical modeling, categorial behavior is not seen as primarily caused by underlying categories, but as an effect of how linguistic units are exemplified in speech and writing and stored in memory. Thus, individual words are not stored under an explicit category label, but rather in a context. For example, the category of 'can' is only noticed in a meaningful context such as '*I can see a can.*' However, the analogical support for a given context gives cues as to which 'categories' could fit, and naturally the same word would have different defaults depending on the context.

### 2. Categories

What could we mean by a linguistic category? A possible answer to this question is that a category is a label used as a stand-in for mutually exchangeable objects during processing. A slightly different answer is that a category is just the observation that some objects are interchangeable without affecting the grammatical status of the utterance. For example, words that could replace *can* in a sentence such as *the can exists* could be called nouns. Category is thus a dynamic concept that only exists in a situation. The first kind of category is useful for describing the behavior of

a collection of individual words, and both types make it possible to generalize from separate instances.

### 2.1 Lexical categories and words

Some objects inherently belong to a specific category. For example, buttons with four holes belong to the category of ‘four-holed buttons.’<sup>1</sup> Other categories have no such defining features. A string of letters gives few clues as to the category of the string, which is the core of Chomsky’s (1975) critique of analogy in language processing (Itkonen & Haukioja 1996). However, analogy involves more than physical resemblance: lexical categories are essentially about relations between words of *grammatical* sentences. It would be desirable to find a useful bidirectional one-to-many mapping between categories and words, but there is little hope for this. Using an appropriate context is more promising. For example, automatic part-of-speech tagging is very accurate nowadays (Brill 1994, inter al.). Using information about the most common tags of a word, and the context in which the word is presented, automatically delivered tag sequences are typically more than 96% accurate.

## 3. Similarity

An analogical approach depends on noting the similarity between different objects to predict properties of new objects from knowledge about old objects. For words, the physical similarity might be misleading. Chomsky (1975: 140–142) gives these two sentences:

- (1) John’s friends appeared to their wives to hate one another.
- (2) John’s friends appealed to their wives to hate one another.

The change of an /r/ to an /l/ changes the functional roles of who hates who. Now, we would not like to argue that (1) and (2) have any great analogical support. *Appear* and *appeal* seldom share the same lexical context, let alone functional context. Itkonen and Haukioja (1996) give the following two sentences in reply:

- (3) John appears to be sleeping.
- (4) \*John appeals to be sleeping.

Sentence four is obviously ungrammatical, but how would we know that if we have experienced sentences (1) and (2)? (1) and (2) support *appear* and *appeal* in the same category, since two words (and a sentence break) of left context and 7 words (and a sentence break) of right context are shared, but (3) and (4) show that the two verbs will not commonly share one syntactic context.

It is vital that the context support the same *functional roles*; the lexical material is of less importance. Thus, similarity is in the meaning of the sentence, which is not tangible in the same sense that we can point to the physical similarity of /r/ and /l/ in *appear* and *appeal*. In sentence (1), the wives are experiencers, while in sentence (2) they receive an appeal, but to know this the learners must know what *appear* and *appeal* mean before constructing the parses, at least to the extent that they know which functional roles to expect from either verb.

The argument is obviously a little circular: to form a representation of the functional roles of a sentence (i.e., to parse the sentence) it is necessary to know the functional roles of the sentence. We can get such information from previously experienced sentences, where these 'functional roles' were obvious enough to be perceived in a concrete situation. Categories that are not perceivable, such as the internal states of living beings, would have to be handled through analogy.

Many examples of analogical modeling are based on support from contexts of physical units, such as representations of sounds and word forms. In the style of standard corpus linguistics, frequencies are calculated from data collections that are taken to be representative of objective units (and not the subjective meaning of the sentence). Useful analogical support would depend on having correct units of contexts, and it seems that something like the meaning of the sentence would be the correct measure (given the previous examples (3) and (4)). Some starting point must exist for analogy to expand knowledge.

Assuming that we have determined the appropriate units for the task at hand, there is yet the question about where we find similarity. For example, word similarity could be in the global distribution of words (i.e., the data collection is the categorization), and/or categorial similarity could be locally stored at the lexical level.

An often used static formalization of similarity is cluster analysis, but such attempts often fail to establish a valid distance metric and appropriate weights of relevant features. Such problems of finding objective similarity lead us to accept that similarity is essentially a dynamic concept, so that the similarity of any two objects depends on the context in which they are observed.

#### 4. Reducing the problem

To get around some of these difficulties, an artificial language was created with a very limited set of vocabulary items.

Lexical categories are difficult to analyze in typical natural language corpora. The first reason is that almost all natural words can be used as representatives of many plausible categories. The second related reason is that even if words were

assigned multiple categories, it would still be hard to estimate the correctness of how those categories are distributed in various contexts. The third reason is the difficulty of finding valuable categories in any collection of grammatical word strings if the use and reference are excluded. Such collections contain too much irrelevant information, and at the same time it might not contain the relevant information for the task (which we suspect to be the functional relations between the words of the sentence). The following will specify how time and lexical complexity was dealt with in the computational experiments.

#### 4.1 Reducing time

For an empirical experiment, a five-word window was used, where each context consisted of four words with the 'category' of that context marked by the remaining fifth word. Time can thus not exceed five consecutive words. The five positions of the category word were evaluated separately. In the artificial language, subject and object roles were given *indirectly* by having different forms for subject and object pronouns.

#### 4.2 Reducing the vocabulary

There is a need to focus on representatives, at least in the open word classes. Words of the open word classes (nouns, verbs and adjectives) have low individual frequencies, but high type frequencies. Some type-classification might be possible without knowing the syntactic class of the word. For example, concrete nouns are observable and might share semantic properties at various levels. It is therefore reasonable to assume that types, represented by familiar words, syntactically represent (i.e., stand in for) open class words. Ignorance of the lexical abundance may be bliss for the learner, as it greatly reduces learning complexity.

#### 4.3 Constructing a language

Example sentences were constructed using an 'English' sentence structure, exemplifying subject and object defective relative clauses, as well as center embedded relative clauses. There were only two nouns that occurred in singular and plural forms, the 'blips' and the 'blops,' and there were three verbs with singular and plural forms and with different transitivity: *give* (to give something to somebody), *bloop* (to bloop something), and *bleep*, as in '*The blip that bleeps the blops bleeps.*' Subject and object forms of personal pronouns in singular and plural were added (i.e. *he, they, him* and *them*). One preposition (*to*) was added to make *dative shift* possible (*He gave blips to the blops.* vs. *He gave the blops blips.*). From the previ-

ous discussion, we have a chance to detect some regularity within contexts, but we would have problems finding regularities that go beyond the contexts. One hope was that the reliance on the subject noun for the correct number of the verb would make it possible to differentiate nouns from verbs.

The following shows some typical sentences:

‘The nice blop bleeps the blops blank’  
(e.g. *The nice man/bird feeds the birds/men.*)

‘The blips that bloop the blop bleep blank’  
(e.g. *The men that feed the bird sing.*)

‘The blip that the blops bloop bleeps blank’  
(e.g. *The bird that the men feed sings.*)

‘The blops that the blip bleeps bloop blank’  
(e.g. *The birds that the man feeds fly.*)

‘He gives the blop to them blank’  
(e.g. *He gives the food to them.*)

‘They give him the blop blank’  
(e.g. *They give him the food.*)

The functional roles are not given explicitly, except for subject and object forms of pronouns. Number agreement is exemplified, but not explicitly forced. Testing the material showed that any two of the 20 words could satisfy the criteria of being exchangeable in some context, with a recursive notion of exchangeable, even with as much as three words of context. The training set was composed of 440 ‘sentences’, presented as five word contexts with 5295 nouns (2900 plural, 2395 singular), 4390 verbs (2765 plural, 1625 singular) and 16370 other words.

The test set was composed of 498 novel ‘sentences’ presented as five word contexts with 5420 nouns (2915 plural, 2505 singular), 1580 verbs (1085 plural, 495 singular) and 16370 other words. These two sets will be used in the following empirical comparison between two learning mechanisms.

#### 4.4 Justification

The simplifications made in handling time may seem artificial. The general problem is that it is difficult to find representative data for something as complex as language, and at the same time provide both a means of evaluation and a task that is relevant but manageable.

Natural language corpora typically do not reflect spread between individuals, nor the age of acquisition, and only represent a small part of the distribution of



the end-result, and are therefore similarly artificial reflections of input to language learners.

## 5. A simple Memory Based Learner

A problem for memory-based models is to justify the choice of categories. In the following a mechanism that may use previous experience to find underlying similarities, without using explicit category labels, is sketched. The heart of the mechanism is to separate the identity of the item (e.g. its phonetic form) from its similarity to other items (in terms of syntactic or phonological properties). Similarity is considered to be an acquired phenomenon, and thus not inherent in the item per se. Finding similarities is accomplished by using a context, which is restricted in this example to neighboring words. Similarity between items is accumulated in a similarity key, which is separate from the identity of the item. The key becomes meaningful when it is compared to other keys. We will later compare performance with another memory-based mechanism, which was provided with category labels.

The memory-based learner presented here is inspired by work on an instance based learning algorithm that competes with an algorithmic process and eventually takes over the job of that process (Logan 1988). This model was expanded with the ability to create its own internal representation of similarity between units of the process. In contrast to most neural network simulations the identity of the symbols are represented separately (i.e., as a symbol or index number). Figure 1 shows an outline of a simple Memory Based Learner<sup>2</sup> (sMBL) that creates an internal representation for lexeme symbols by utilizing the best match from Long Term Memory (LTM) to the current five words in Short Term Memory (STM). The representation of similarity starts as an ordered sequence of random binary values. It is possible to think of such sequences as a proposal for feature values, therefore a specific position in the sequence will be referred to as a feature, which may have a value of 0 or 1. Note that the *identity* of each item never changes.

These features can be used to find *best matches*, although they lack specific reference. Initially, best matches will tend to match exact words since that guarantees that at least those words will have a perfect match. With time, words that are used in similar contexts may get more similar representations of similarities through the use of simple learning rules.

A very simple pair of rules acts on the status of the individual feature match between the input (short term memory) and the retrieved best match from long term memory. The rule pair statistically reduces the number of missed features in the future by creating a match with a higher probability than creating a mismatch.

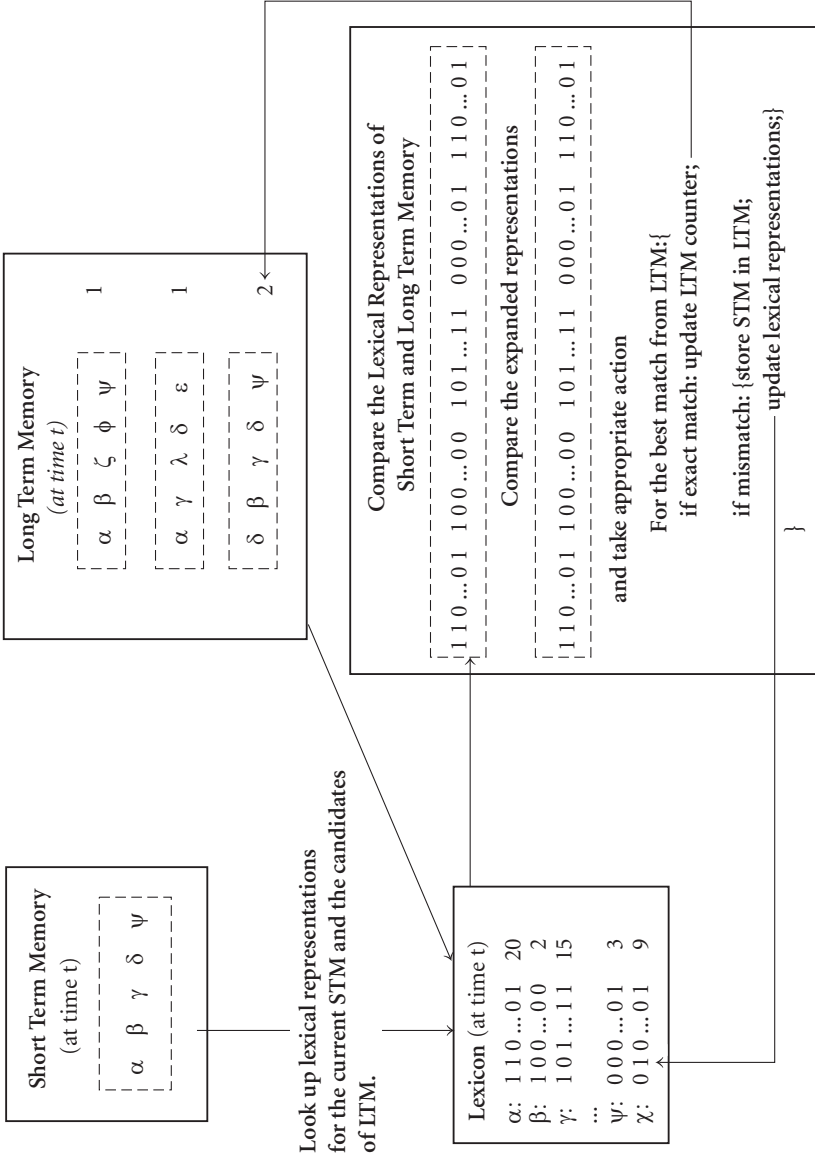


Figure 1. Outline of a simple Memory Based Learner

*Rule 1a.* If there is a mismatch between a feature in STM and its corresponding feature in LTM, then with probability 0.5 switch the value of the feature in STM.

*Rule 1b.* If there is a *match* between a feature of STM and LTM, then with probability  $p$  switch the value of the feature in STM ( $p$  was set to 0.05).

If there were any mismatches, then the content of the STM was novel, and it is consequently stored after the rules 1a and 1b have been applied to all features of the ‘similarity key.’ Perfect matches of the whole content of STM are never stored, but do update a counter of how many times that content has occurred.

These rules cause the similarity key stored in the feature vectors of lexemes to become more similar for words occurring in similar contexts, while keeping each word fairly distinct from other words. The rules are guided by random changes that happen with varying frequencies depending on the match status of individual features. These features are initially meaningless, and the procedure to change them does not have any precise plan or goal for making features behave better (unlike error feedback in connectionist models).

## 5.1 Results

The memory-based learner was tested with a training and a test set from the constructed language described previously. In the test phase, one model word (e.g., all ‘blips’) was replaced by an untrained word with a new random signature. The mechanism was given novel sentences with these new words. The task was to retrieve the best matching five word sequence, and thus a previously experienced word for the new word. This was repeated for six different random signatures, for each of 8 different model words (4 nouns, 4 verbs).

The model (Johansson & Stowe 2000) was successful at retrieving words of the same category as the model words for these unknown words. It got 80–100% correct for unknown words of the noun category, but performance for verbs was poorer (20–80%, see Figure 2). Verbs were typically confused with words that would otherwise be in proximity to nouns (e.g., determiners *{a, the}*, the generic adjective *nice*, or the generic preposition *to*).

## 6. Analogical Modeling

Skousen’s (1989, 1992) Analogical Model (AM) was applied to the same five word contexts as in the previous experiment. One word was left out and replaced by its category (*Noun*, *Verb*, or *Other*) making it a *slightly easier* task than the previous experiment. Each category has four words of context, and categorization was

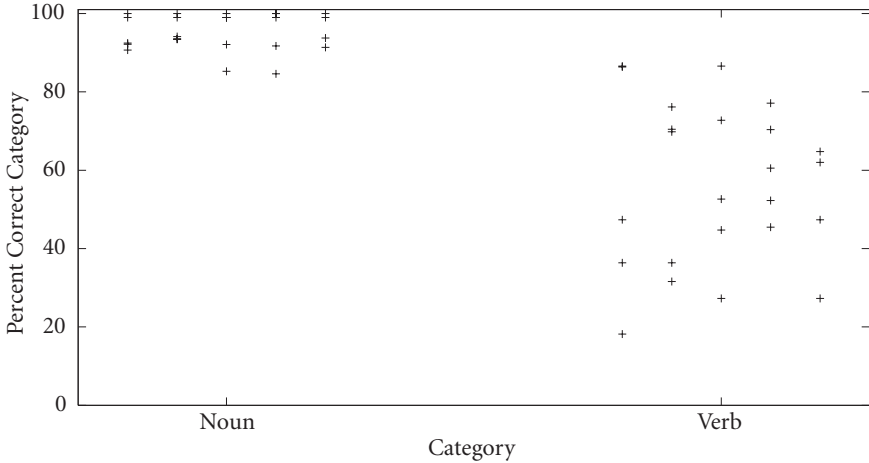


Figure 2. Success of a Memory Based Learner

modeled by how words co-occur within the analogical sets, and how that supports each of the three categories (N, V, O). The same test set was used as in the previous example. The category with the highest support was taken to be the category suggested by this method.

### 6.1 Lexical contexts and analogy

A lexical context consists of words that occur together with the words of interest. It is assumed that a stream of words is observed through a (short term) memory ‘window’ which has a limited capacity to hold local contexts. The window gives a temporal dimension to the possible contexts. In the case that the window contains two words, the context can occur either to the left, e.g., ‘*in* ← *the*’, or to the right of the word, e.g., ‘*the* → *cat*.’ A second word may have a contextual relation with ‘*the*’ if both left and right contexts are shared, as in ‘*in the / in a : the cat / a cat*.’

This is more generally expressed as ‘*ax/ay : xb/yb*’ where *a* and *b* are context words shared by words *x* and *y*. Using only two words in the ‘window’ will find contextual relations between almost any *x* and *y*, with different strengths. For simplicity, let us consider three word sequences arranged in positions A, B, and C. Let such sequences be categorized by the word following. Figure 3 shows the possible overlaps between word sequences. For example, sub-context A–C is the 3 word sequences that have words in the A and C position in common. We may calculate the analogical support by noting how the categories fall within each of the subsets.

The example of predicting the behavior of pattern 026, given in Skousen (1989:40), is illustrated in Figure 3. The pattern 026 selects data presented in the

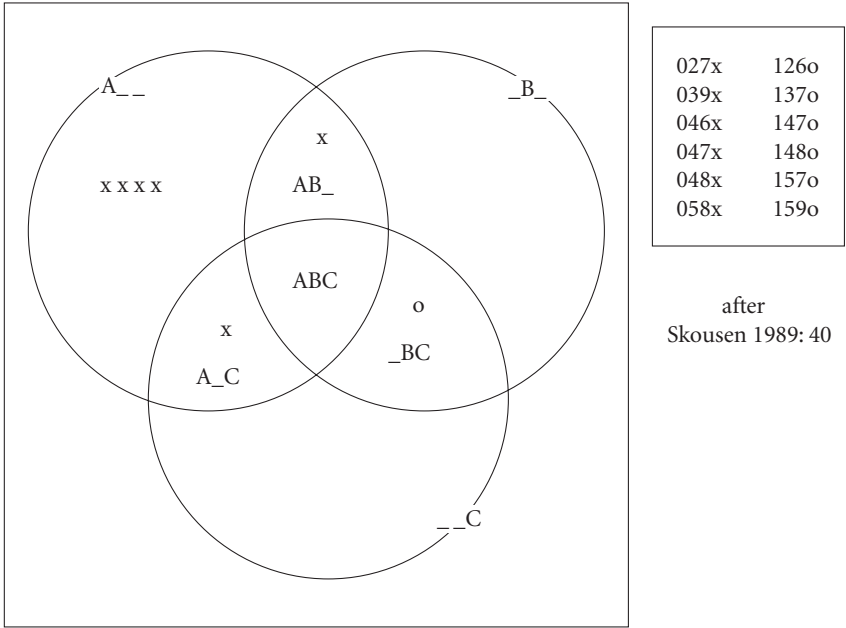


Figure 3. Sub-contexts

figure by partial match. The selected data are plotted by their category (x or o) at the appropriate place in the diagram.

In order to calculate the probability of category x for pattern 026, (see Table 1) all possible pairs *within each sub-context* are investigated. Pairs could either support the same category, in which case they are counted as one piece of evidence for that category; or they could support different categories, which cancels the support of that pair. If the full context is available (i.e., the full context is a member of ABC), it will be part of all sub-contexts, and consequently form pairs with all contexts, and therefore have a powerful impact. More information can be found in the works of Skousen (1989, 1992). In the simulations to follow all calculations have been performed by a program supplied by Royal Skousen on the internet.<sup>3</sup>

6.1.1 Time and position of context

In a three word window, AB can cooccur with the word of interest in three ways: ABX, AXB, and XAB, showing that X can be experienced at three temporal positions. We mark the position of the words used as categories with an X. A three word memory span would produce the three context arrangements ‘abx/aby : cxd/cyd : xef/yef’. For example, ‘in the house / in the garden : the house is / the garden is : house is nice / garden is nice’ supports a similarity of ‘house’ and ‘garden’. We can repeat

Table 1. Calculating the analogical support of pattern 026

sub-context	support x	support o
A --	6 · 6	
A B -	1	
A - C	1	
- B C		1
TOTAL:	38 (x) (38 out of 39 predict x)	1 (o)

this procedure for a growing context window, and find candidate pairs sharing increasingly specific categories.

Assuming that such pairs have the same category, we can form larger categories by assuming the category equivalent to the connected graphs of the reciprocally connected word pairs. Say that the pairs ‘*apple – banana*’ and ‘*banana – peach*’ have been supported, then the triplet ‘*apple – banana – peach*’ is also supported since ‘*apple*’ can reach ‘*peach*’ from ‘*banana*’.

In a corpus test (using the Susanne corpus, Sampson 1994), it was found that mostly function words were grouped together using this strategy, since only high frequency words survive a demand of overlap for longer contexts. This means that classification by analogy is better based on other information than only word form, as argued previously.

## 6.2 Results

The following presents the success of AM on the same task given to sMBL. The results for each context position of the missing word (X) can be seen in Figure 4. Nouns perform the best when both right context and left context are available, whereas performance on verbs improves in the complimentary case of exclusive right or left context.

Overall performance is lower than for sMBL on basically the same task. This is a little disappointing since the task was made easier for AM. The results in Figure 4 are based on training on three categories only (Noun, Verb, and Other). Using all 20 words was possible (Table 2) but involved some decisions made post-hoc about which category had been selected, especially in the case of ties. This problem was not present for sMBL since it always presented the top match, and ties were decided by giving the most recent match priority.

The quality of pairs of words judged to be in the same category by AM is clearly better for the ABCXD context (Table 2). These pairs exemplify correct category and number. Correct number is found in matches between nouns and pronouns

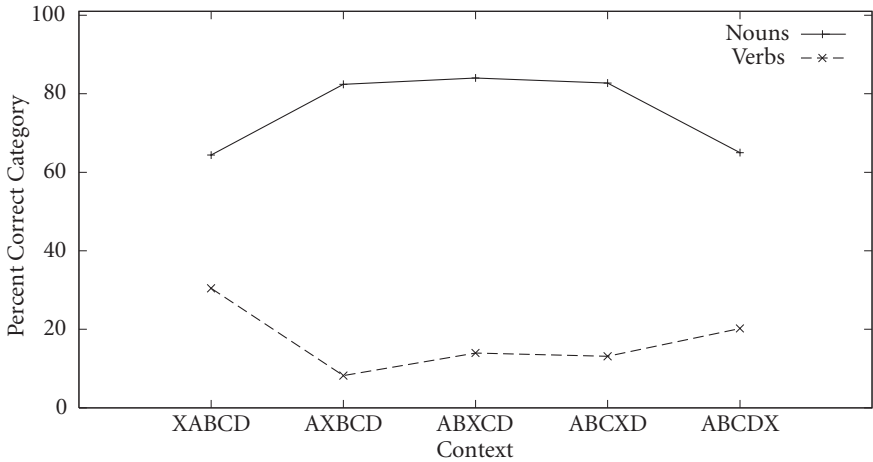


Figure 4. Success of Analogical Modeling

in subject form. That ‘him’ and ‘them’ match means that number does not matter grammatically for object pronouns.

It can be noted that verbs are rarely placed in the same category; the example shows ‘give – bloop’, which are both transitive verbs in the plural number, even though *give* can take two objects (the direct and the indirect object). That verbs do not generally have much support from other verbs is in a sense good since each verb represents intransitive, transitive and double transitive verbs in the singular or plural form.

Four general observations can be made. First, all arrangements of temporal contexts are not equal. Second, contexts may select relevant categories of words by using co-occurrence within the analogical sets. Third, no feedback is necessary

Table 2. Examples of exchangeable words

XABCD	AXBCD	ABXCD	ABCXD	ABCDX					
blip	nice	the	a	blips	blops	blips	blops	blip	nice
the	a	them	him	to	eos	blip	blip	the	a
that	to	that	to	that	to	them	him	that	eos
that	eos	nice	the	blip	nice	the	a	that	to
them	him	blip	blops	blip	blops	blops	they	them	him
blip	blops	blips	blops	the	a	give	bloop	nice	the
nice	the	blips	blip	them	him	give	eos	to	eos
blips	blops	to	eos	blips	blip	blips	they	blips	blops
blips	blip	blip	nice	give	eos	eos	bloop	give	bloop
give	bloop	nice	a	that	eos	blips	blip	blip	blops

about the actual word of the contexts, in either the memory-based learner or the model of analogical support. Lastly, nouns seem more easily inferred from *lexical* context than verbs.

## 7. Discussion

The memory based learner and AM are in fact similar in many respects. First of all, both models operate without feedback about performance. Secondly, they rely on a database of saved experience. Thirdly, both models are, in principle, able to detect when an event has *not* been experienced before. Lastly, no reason has been presented so far that would exclude one of the models given the other.

The difference in performance is most likely due to the fact that the memory-based learner was allowed to create a similarity between appropriate lexical items. Simple memory based learning operates by selecting a stand-in for the current input, based on previous experience. Another difference is that sMBL was forced to deliver one alternative only, whereas AM could rate the amount of analogical support for many alternatives. There is no principled reason why AM cannot be extended to handle similarity between the symbols of the contexts.

With a realistic vocabulary ( $10^5$  items), it would be unrealistic to expect exact matches for any longer contexts. The number of possible 4-tuples would be in the order of  $(10^5)^4 = 10^{20}$ . It would be risky to base an analogy on a few exemplars in such a vast space, if there were only a few close to identical matches. Given that there is a lower number of appropriate categories (e.g. 10 functional roles), we would be more likely to find a more supportive analogical set with a high rate of highly matching contexts.

### 7.1 Why is AM performance low?

The performance is visibly lower for AM than for sMBL, but AM is more constrained. The first constraint is that AM calculates the objective support for the category of a new pattern based on the current database as it is. There are no assumptions about the distributions of the alternatives, and there is no attempt to replace words for each other. Thus, words do not stand-in for each other in the exemplars (which is one of the ideas about categories presented earlier).

A second constraint is that AM does not have an optimizing strategy. Let me give an example. Say that two categories, x and o, occur by chance but with different rates, for example 80% x and 20% o. If we try to model this distribution we would likely be less than 80% correct. In fact, it is likely that we have 64% (i.e.,  $80\% \cdot 80\%$ ) matches on the x category, and similarly 4% matches on the o category. I am not



saying that the ‘language’ we used was random, but there were contexts where the correct category was unpredictable. An *optimal* strategy to category selection under uncertainty is to choose the most frequent category. Under the assumption that the test cases repeat the distribution of the training set, this would guarantee the success-rate of the most common alternative (in this case 80%). Furthermore, we would know that 20% of the cases will be missed, and that the category of those cases will be o.

AM uses a variant of random selection, and it can therefore approximate the frequency distributions of the given data, which is not possible in some other models (Skousen 1989: 81–86). It could therefore be argued that AM is a suboptimal but formally correct model of language performance. The emphasis is actually not on predicting the correct classes, but rather how strongly to expect them in a context. Skousen cites some supporting research that show that older children can acquire “selection by plurality” as a strategy for increasing success when a reward is given. The ability to select between two different strategies show that we are indeed likely to use separate instances to predict behavior since we can approximate not only the relative frequencies of items, but also the most common alternative.

The sMBL uses only the best matching instance. This closely approximates implementing a strategy of selecting the most frequent alternative given a context, since when an ambiguous context occurs it will be more likely to choose the more frequent (in fact, ties are settled by frequency of the context). Therefore, it is not so surprising that this learner performs a little better than AM. The second reason that this version of MBL performs better is that it constructs similarity keys in order to detect words with a similar usage. This ability is currently not present in AM. Moreover, the sMBL approach makes it possible to select a (best) match by going through the database once, whereas the AM approach needs to calculate the support from all combinations selected from the database by the current input, which is a much tougher computation. The most significant difference, however, is that sMBL implicitly detects classes from un-classified examples, whereas other memory-based approaches rely on pre-classified examples.

AM is certainly useful as a benchmark for other learning approaches, as it quantifies how much you can trust the support in the database given no other assumptions (e.g. about the distributions in the database). Algorithms that gain such information from exposure to the database are likely to produce higher performance, but that performance will depend on qualities of that database and the methods to generalize between items of that database.

## 7.2 Why is verb performance poor?

There are several reasons why nouns perform better than verbs. The simplest reason is that there are 20% more nouns than verbs in the training set. A second reason is that verbs have three degrees of transitivity.

Nouns are more determined by context, since determiners (*a*, *the*) always start a noun phrase, and the end of a noun or verb phrase is often followed by either *the*, *a*, or *that*. In addition to these distributional factors, a verb is also a relation between words, whereas nouns can be used as an independent label.

Verbs prove tougher than nouns to learn for human learners as well. Studies suggest that this may be due to different conceptual requirements between the categories, or different informational constraints. A summary and some newer results can be found in Gillette, Gleitman, Gleitman, & Lederer 1999.

## 8. Conclusion

Selecting more appropriate representations and focus on a subset of lower complexity would help ‘categories’ to emerge in the observed behavior. However, there are still some miles to go before our algorithms show clearly categorical behavior.

In the process of comparing AM with our version of simple MBL, some characteristics of the models were detected and investigated. It could prove helpful to view AM as the correct measure of analogical support. AM gives the objective support from the database with very little added assumptions. The success-rate of AM is therefore a rough measure of the complexity of the database. Models that perform better on the same task would have to motivate why this is so.

In the case of sMBL, we detected useful similarities between items, which made it possible to support lexical decisions by approximate categories rather than individual words. Furthermore, this was accomplished without actually providing explicit category labels.

## Acknowledgments

This article was prepared while the author enjoyed a fellowship from the Science and Technology Agency of Japan, at the Electro Technical Laboratory, Tsukuba. Support from host researcher Koiti Hasida is kindly acknowledged.

## Notes

1. This is obviously an allusion to Strindberg's term 'buttonology' for 'scientific' categorization of uninteresting facts.
2. C-source code is available on request to the author on e-mail: <christer.johansson@lili.uib.no>
3. <<http://humanities.byu.edu/am/>>

## References

- Brill, Eric (1994). Some advances in rule-based part of speech tagging. In *Proceedings of the twelfth national conference on artificial intelligence (AAAI94)* (pp. 722–727). Seattle, Washington.
- Chomsky, Noam (1975). *Reflections on language*. New York: Pantheon.
- Gillette, Jane, Henry Gleitman, Lila Gleitman, & Anne Lederer (1999). Human simulations of vocabulary learning. *Cognition*, 73, 135–176.
- Itkonen, Esa, & Jussi Haukioja (1996). A rehabilitation of analogy in syntax (and elsewhere). In A. Kertész (Ed.), *Metalinguistic im wandel: Die 'kognitieve wende' in wissenschaftstheorie und linguistik* (pp. 131–177). Peter Lang.
- Johansson, Christer, & Laurie A. Stowe (2000). Lexical categorization from exemplars. Manuscript.
- Logan, Gordon (1988). Towards an instance theory of automatization. *Psychological Review*, 95, 492–527.
- Sampson, Geoffrey (1994). *The Susanne corpus*. University of Sussex.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.

PART VI

## Quantum computing and the exponential explosion



## CHAPTER 13

# Analogical Modeling and quantum computing\*

Royal Skousen

### The exponential explosion and quantum computing

Analogical modeling, from the very beginning, has proposed that in predicting behavior all possible combinations of variables should be tested (either directly or indirectly). If there are  $n$  variables for a given context, there will be  $2^n$  supracontexts (or combinations of variables) to consider. Basically, increasing the specification by one variable doubles the memory requirements as well as the running time (Section 6.1 of Skousen 1989; also see Daelemans, Gillis, & Durieux 1997). There have been numerous attempts to deal with this intractability: fine-tuning the computer program, revising the algorithm so that it would not have to keep track of every possible supracontext, and using parallel processing.

A new approach to dealing with the problem of the exponential explosion in analogical modeling has been to re-interpret analogical modeling in terms of quantum computing. (For a general introduction to quantum computing, see Williams & Clearwater 1998; Lo, Popescu, & Spiller 1998; Berman, Doolen, Mainieri, & Tsifrionovich 1998; or Hey 1999.) One distinct theoretical advantage of quantum computing is that it can simultaneously keep track of an exponential number of states (such as  $2^n$  supracontexts defined by an  $n$ -variable given context), thus potentially reducing intractable exponential problems to tractable polynomial analyses (or even linear ones). In certain well-defined cases it has been shown (in pseudo-code only, since there is no complete hardware implementation of quantum computing thus far) that the exponential aspects of programming can be reduced to one of polynomial degree (which entails tractability, unlike exponential cases). Quantum computing allows for certain kinds of simultaneity or parallelism that exceeds the ability of normal computing (sequential or parallel). The examples discussed thus far in quantum computing involve numbers, especially cryptography, as in Peter Shor's algorithm for determining the prime factors of a long integer (see, for example, Williams & Clearwater 1998: 133–137).

One reason for considering quantum analogical modeling is that the exponential factor seems to be inherent in all approaches to language processing. Thus far, linguistic evidence argues that virtually all possible combinations of variables can be used by native speakers in predicting language. The exponential problem is explicitly required in analogical modeling, and normal kinds of parallel processing will probably fail to solve this problem. Nor is the exponential explosion in predicting language restricted to analogical modeling. Other exemplar-based approaches and neural networks (connectionist approaches) also encounter exponential problems since researchers using these non-declarative approaches must decide how to limit their predictions to those based on the “most significant” variables. The difficulty for these other approaches is in the training stage, where the system has to figure out which combinations of variables are significant, a global task that is inherently exponential.

In the early 1980s, as Skousen was writing *Analogy and Structure* (published later as Skousen 1992) and setting down the basic principles of analogical modeling, he had no idea of its possible connection with quantum mechanics or the possibility that quantum computing might be used to do analogical modeling. Of course, at that time there was only the initial formulation of what quantum computation might involve (for instance, in Feynman’s early ideas and Deutsch’s universal quantum computer, plus Landauer’s and Bennett’s earlier work on reversible computation). Skousen’s motivation for analogical modeling was linguistic, although in its mathematical formulation in *Analogy and Structure* considerable attention was paid to measures of uncertainty and accounting for the general nature of rule systems.

The original characterization of analogical modeling has surprisingly remained unchanged over the last two decades. Its application to a number of linguistic problems (both general and specific) has shown that analogical modeling continues to make the right kinds of predictions, perhaps because of its similarity with quantum mechanics, a theory which has been successfully applied to virtually all aspects of physical behavior since its first formulation in the 1920s. More recently, there has been an important realization that quantum reality and information theory are closely related, emphasized, for instance, by John Archibald Wheeler (see his article “Information, Physics, Quantum: The Search for Links” in Hey 1999:309–336). The close relationship between analogical modeling and information theory implies that the striking similarities between analogical modeling and quantum mechanics may not be accidental at all – that in actuality the mechanisms used by speakers of languages to learn and use language may involve quantum computing.

One advantage of analogical modeling is that no mathematical (or statistical) calculation is actually used in determining the analogical prediction; instead, there is just the simple comparison of deterministic and non-deterministic supracon-

texts. This kind of decision-making process is based on what is referred to as a natural statistic. Natural statistics are psychologically plausible and avoid any direct consideration of probability distributions, yet have the ability to predict stochastic behavior as if some underlying probability distribution were known. The simplicity of analogical modeling suggests that some very basic operators could be used to determine a quantum analogical set that would then be reduced to a single supracontext (combination of variables) whenever decoherence (or observation) occurs.

### Similarities between analogical modeling and quantum computing

One initial reason for pursuing the possibility of quantum computing of analogical modeling is that a number of striking similarities have been discovered between analogical modeling and quantum mechanics:

1. Traditional statistics assume some complicated underlying mathematical functions, but from natural statistics (which involve no direct numerical calculations) we can derive the results of standard statistics if we assume that the probability of remembering any given data occurrence equals precisely one-half. This relationship implies that traditional statistics can be derived from natural statistics if data occurrences are accessed through, say, a spin-up state (given two equally probable quantum states, spin-up and spin-down).
2. In both quantum mechanics and analogical modeling, there is an underlying linearity as well as an observed squaring. In quantum mechanics, prior to observation, an exponential number of quantum states can be simultaneously accounted for, yet when observed, this superposition of many states is collapsed into a single one, a process referred to as decoherence. Prior to observation, each quantum state is assigned an amplitude, but this amplitude is squared to give a probability when observation occurs. A single observation leads to this decoherence and squaring of the amplitude. In analogical modeling, there is an exponential number of supracontexts (combinations of variables) for a given context. We keep track of the number of occurrences (a linear function) for each supracontext. When we come to predicting an outcome, one of the supracontexts is selected and the probability of selecting that supracontext is proportional to the square of the number of occurrences in that supracontext. The squaring naturally results from selecting a pointer to an occurrence rather than directly selecting an occurrence.
3. In analogical modeling, a quadratic measure of agreement is used to measure certainty. Agreement is based on the idea that one gets a single chance to determine the outcome. This single observation corresponds to the decoherence that occurs when a quantum system is observed. Moreover, this measure of



agreement corresponds to Schrödinger's wave equation, where squaring is used to determine the probability of occurrence.

In the next three sections, these points are discussed in some detail.

### Traditional statistics from natural statistics

While investigating natural statistics, Skousen (1998) discovered that when the probability of remembering is one-half, we get standard statistical results (including the ability to account for the traditional "level of significance" used in statistical decision making). However, there seemed to be no inherent motivation for why this one-half probability of remembering should lead to traditional statistics. But the one-half probability can be justified if we interpret it as corresponding to storing the individual occurrences of a database by means of a vector composed of quantum bits, each with an equal chance of being accessed or not (much like an electron's spin, with its two states of up and down).

There are two specific results from natural statistics that argue for the significance of the one-half probability of remembering (Skousen 1998:247–250). First, consider the task of estimating the probability of occurrence  $p$  for an outcome. Suppose we have two possible outcomes, either  $s$  or  $t$ . Suppose further that we have been given the following string of outcome data:

*s s s t s t t t t t s t s t t t t s t*

If we have perfect memory (where the probability  $r$  of remembering is one), then in natural statistics, the probability  $p$  of predicting the  $s$  or  $t$  outcome is directly proportional to the relative frequency of each outcome in the data. So in this string of occurrences, where there are 8 examples of  $s$  and 12 of  $t$ , we get the following predictions under perfect memory ( $r = 1$ ):

$$p(s) = 8/20 = 0.4 \quad p(t) = 12/20 = 0.6$$

When memory is perfect, we always get this same estimated probability  $p$  for the outcome  $s$  (namely, 0.4); in other words, there is no variance in our estimate for  $p$ :

$$\text{Var}(p) = 0 \text{ if } r = 1 \text{ (perfect memory)}$$

Suppose there are  $n$  occurrences in the data and that  $m$  occurrences are remembered. We can first show that the expected value  $E$  of the probability  $p$  of an outcome is simply the probability of that outcome – that is, we have an unbiased estimator for  $p$ :

$$E(p) = p \text{ (outcome)}$$

When we consider the variance for this estimator, we get the following relationship (Skousen 1998:248):

$$\text{Var}(p) = 1/(n-1) \cdot E(p)(1-E(p)) \cdot (E(n/m) - 1)$$

The two expectations,  $E(p)$  and  $\text{Var}(p)$ , hold no matter what  $r$ , the probability of remembering, is.

When  $r = 1/2$ , a given data occurrence is remembered – or is accessible – half the time (on the average). Under these conditions and for large  $n$ , the number of remembered occurrences ( $m$ ) is approximately equal to  $n/2$ . Thus the expected value for the ratio  $n/m$  will be approximately equal to 2. This means that for large  $n$  we get the following asymptotic relationship for the variance of  $p$  when  $r = 1/2$ :

$$\text{Var}(p) \approx 1/(n-1) \cdot E(p)(1-E(p))$$

Now this asymptotic measure of variance derived from natural statistics (but only when the memory is  $1/2$ ) is precisely the same as the traditional unbiased estimate of variance (which assumes that the relative frequency is first used to estimate  $p$ ).

Now consider a second statistical task. Suppose we have some data with the two outcomes  $s$  and  $t$ , and we want to predict the most frequent of these two outcomes. For simplicity of calculation, suppose our outcome data for this example consists of only the following four occurrences:

$s s s t$

Now the chances of the  $s$  outcome being more frequent than the  $t$  outcome is assured if we have perfect memory (when  $r = 1$ ). Under such conditions, there will always be three occurrences of  $s$  and one of  $t$ , so there will be no uncertainty in our prediction:

$$p(s > t) = 1 \text{ if } r = 1$$

On the other hand, when  $r = 1/2$ , each occurrence of the four will be remembered – or accessed – half the time (on the average), which will thus give 16 equally possible cases (Table 1).

In 11 cases of these 16 cases, the more frequent outcome will be  $s$ , while in one case,  $t$  will be the more frequent. In three cases, we get a tie between  $s$  and  $t$ , so we split the probability in those cases. And in one case, we forget all four occurrences. In that case, we are unable to make a prediction. We represent this as the null outcome ( $\emptyset$ ) in the list of possibilities:  $s$ ,  $t$ , and  $\emptyset$ . Given an imperfect memory of  $r = 1/2$ , the overall probability that natural statistics predicts  $s$  as the more frequent outcome therefore equals  $25/32$ .

Natural statistics ends up making predictions that are equivalent to standard statistical decision theory, which sets up various levels of significance to represent the probability that a null hypothesis should not be rejected. In this partic-

Table 1. 16 sets of remembered occurrences

		$p(s)$	$p(t)$	$p(\emptyset)$
1/16	s s s t	1/16	–	–
1/16	s s s -	1/16	–	–
1/16	s s - t	1/16	–	–
1/16	s - s t	1/16	–	–
1/16	- s s t	1/16	–	–
1/16	s s - -	1/16	–	–
1/16	s - s -	1/16	–	–
1/16	s - - t	1/32	1/32	–
1/16	- s s -	1/16	–	–
1/16	- s - t	1/32	1/32	–
1/16	- - s t	1/32	1/32	–
1/16	s - - -	1/16	–	–
1/16	- s - -	1/16	–	–
1/16	- - s -	1/16	–	–
1/16	- - - t	–	1/16	–
1/16	- - - -	–	–	1/16
<b>Totals</b>		<b>25/32</b>	<b>5/32</b>	<b>1/16</b>

ular problem, the null hypothesis (from the natural statistics point of view) states that the more frequent outcome  $s$  is not more probable than the less frequent outcome  $t$ . There is more impreciseness in the natural statistics approach since there is a probability of predicting no outcome (in the above example,  $p(\emptyset) = 1/16$ ). Asymptotically, the same predictions are made as in standard statistics, but only when the probability of remembering is one-half.

Once more the obvious question is: Why should natural statistics be equivalent to traditional statistics only when  $r = 1/2$ ? This result naturally follows if each exemplar (or occurrence in the data) is accessed via a quantum bit (qubit) which is in either a spin-up ( $\uparrow$ ) or a spin-down ( $\downarrow$ ) state, and for which only one of these two states will permit accessibility. We suppose that each qubit has an equal chance of being in one of these two states. The direct asymptotic consequences will be that (1) the variance for estimating the probability of an outcome will be the standard unbiased estimate of variance, and (2) predicting the most frequent outcome will be the same as in standard statistical decision theory.

Accessibility to data also solves another difficult problem, that of randomness itself. In simulations of probabilistic behavior, computers can use complicated pseudo-random functions to produce a sequence of integers. Such a sequence may appear random for long strings, but ultimately it is not random, but instead is fully predictable (by the pseudo-random function). It is psychologically implausible that

these complicated pseudo-random functions might be directly used by humans to predict non-deterministic language behavior.

On the other hand, true randomness is inherent at the quantum level. By providing random access to an occurrence (or to a pointer to an occurrence) in terms of qubits, we get actual randomness. In his descriptions of random selection as a rule of usage, Skousen never stated how the speaker would in fact be able to randomly select an occurrence (or a pointer to an occurrence). The problem of randomness was ignored in his initial work (Skousen 1989:37 and 1992:222). But by making an occurrence (or its pointer) accessible only when the assigned qubit is, say, in a spin-up state, actual randomness could be achieved. Furthermore, the statistical results would be asymptotically the same as standard statistics when we assume that the chances of the two qubit states (spin-up and spin-down) are equal.

### Probabilities in quantum mechanics, pointers in analogical modeling

In analogical modeling, there is a lattice of supracontexts, partially ordered by the relationship of set inclusion. This lattice is defined by the given context, which is the set of variables for which we are trying to predict the outcome. Given  $n$  variables in the given context, there are  $2^n$  possible (unordered) combinations of those variables. In analogical modeling, each one of these possible combinations is called a supracontext. For instance, in attempting to predict the pronunciation of the initial  $c$  of the word *ceiling* in terms of the 3 letters following the  $c$  (namely, *eil*), we set up  $2^3 = 8$  supracontexts for this given context (*eil*). For each supracontext we identify which exemplars belong and note their pronunciation of the initial  $c$  letter, such as the /k/ sound for *coin*, the /s/ sound for *cell*, and the *ch* sound (represented as /č/) for *chin* (Table 2).

Table 2.

	linear			squared			exemplars
	$k-c$	$s-c$	$č-c$	$k-c$	$s-c$	$č-c$	
eil	–	–	–				
ei-	–	–	–				
e-l	–	1	–	0	1	0	<i>cell</i>
-il	–	–	–				
e--	–	3	–	0	9	0	<i>cell, cent, certain</i>
-i-	1	–	1	2	0	2	<i>chin, coin</i>
× --l	1	3	–				
× ---	21	9	3				

Some of these supracontexts have no occurrences ( $eil$ ,  $ei-$ , and  $-il$ ). Some have only one type of outcome ( $e-l$  and  $e--$ ) and are therefore deterministic in behavior. One ( $-i-$ ) is non-deterministic, yet has no subcontext that behaves differently. This kind of non-deterministic supracontext and the deterministic ones are homogeneous in behavior. Finally, there are some supracontexts ( $--l$  and  $---$ ) for which there is at least one subcontext that behaves differently. Such non-deterministic supracontexts are heterogeneous. The  $\times$ 's placed in front of the last two supracontexts mark these two supracontexts as heterogeneous.

In quantum computing, we will have  $n$  qubits for a given context of  $n$  variables, but these  $n$  qubits, unlike  $n$  classical bits, will allow us to simultaneously represent  $2^n$  states – namely, the superposition of all possible supracontexts. The advantage of quantum computing is that it allows massive simultaneity.

Each qubit has two states for each variable  $i$ :

spin up	$\uparrow$	1	variable $i$ in supracontext
spin down	$\downarrow$	0	variable $i$ zeroed out

These qubit variables defined by the given context are not assigned their spin-up and spin-down states independently of each other. Instead, there are important correlations between the qubits (referred to in quantum mechanics as entanglement). Moreover, each qubit is normally in a probabilistic state, a mixture of spin up and spin down.

For each of the  $2^n$  supracontexts, we assign an amplitude. Ultimately, when we come to observe our lattice of supracontexts, we can require that the squares of these supracontextual amplitudes are normed; that is, the sum of the squared amplitudes equals one. This norming basically requires that for each supracontext the squared amplitude represents the probability of selecting that supracontext. The norming is really only necessary because probabilities themselves are mathematically defined as normed – that is, as a measure on the line  $[0,1]$ .

One important requirement for applying quantum computing to analogical modeling is that all empty and heterogeneous supracontexts must end up with zero amplitude (equivalent to zero probability of being selected). We need, of course, reversible operators to zero out heterogeneous supracontexts and make sure the empty supracontexts remained zeroed out. The remaining homogeneous supracontexts will, of course, show entanglement between the qubits representing the variables.

The first important connection between analogical modeling and quantum computing is that the number of occurrences assigned to a given supracontext is equivalent to the amplitude. In other words, linearity in analogical modeling corresponds to the amplitude in quantum computing. In our example for *ceiling*, we have the following amplitudes prior to norming, but after determining heterogeneity (Table 3).

Table 3.

	<i>k-c</i>	<i>s-c</i>	<i>ĉ-c</i>	<i>amplitude = occurrences</i>
111 ↑↑↑	–	–	–	0 empty
110 ↑↑↓	–	–	–	0 empty
101 ↑↓↑	–	1	–	1 deterministic
011 ↓↑↑	–	–	–	0 empty
100 ↑↓↓	–	3	–	3 deterministic
010 ↓↑↓	1	–	1	2 non-deterministic
× 001 ↓↓↑	1	3	–	0 heterogeneous
× 000 ↓↓↓	21	9	3	0 heterogeneous

The requirement of normality means that the actual amplitude is the frequency of occurrence divided by the square root of the sum of the squared frequencies ( $\sqrt{\sum x^2}$ ) – in this case, the norming fraction is  $1/\sqrt{14}$  since  $1^2 + 3^2 + 2^2 = 14$ .

Thus the number of occurrences in each homogeneous supracontext (the linear measure) is proportionally related to the amplitude. We can therefore give an alternative representation using Schrödinger's wave equation  $\Psi$  (in Dirac's notation). In the following example, each occurring homogeneous supracontext (101, 100, 010) is represented as a possible state:

$$|\Psi\rangle = 1/\sqrt{14} |101\rangle + 3/\sqrt{14} |100\rangle + 2/\sqrt{14} |010\rangle$$

Now in quantum computing, the probability of occurrence for each homogeneous supracontext will be the square of the amplitude. In order to predict the behavior of our system, we need to select a single supracontext from our superposition of  $2^n$  supracontexts. In other words, observational decoherence of the superposition is equivalent to selecting an occurring homogeneous supracontext, but instead of using occurrences to make the selection, we use pointers to do that. In other words, the squaring of the amplitude in quantum computing is equivalent to selecting a pointer to an occurrence rather than selecting an occurrence directly. This means that if we use quantum computing to do analogical modeling, we will always be selecting the squaring function of analogical modeling. Earlier work in analogical modeling allowed either linearity or squaring (Skousen 1992: 8–9), but now the choice of squaring over linearity is motivated.

In our example, decoherence of the superposition therefore leads to a probability. The probability of each supracontext is proportional to the square of the number of occurrences in that supracontext – in other words, proportional to the number of pointers to occurrences in that supracontext (Table 4).

By norming the number of pointers, we get the following probabilistic predictions using quantum analogical modeling (Table 5).

Table 4.

	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	<i>probability = pointers</i>
111 ↑↑↑	–	–	–	0 empty
110 ↑↑↓	–	–	–	0 empty
101 ↑↓↑	–	1	–	1 deterministic
011 ↓↑↑	–	–	–	0 empty
100 ↑↓↓	–	9	–	9 deterministic
010 ↓↑↓	2	–	2	4 non-deterministic
× 001 ↓↓↑	3	12	–	0 heterogeneous
× 000 ↓↓↓	693	297	99	0 heterogeneous

Table 5.

	<i>probability</i>	<i>k-c</i>	<i>s-c</i>	<i>č-c</i>	<i>exemplars</i>
101	$(1/\sqrt{14})^2 = 1/14$	0	1	0	<i>cell</i>
100	$(3/\sqrt{14})^2 = 9/14$	0	9	0	<i>cell, cent, certain</i>
010	$(2/\sqrt{14})^2 = 4/14$	2	0	2	<i>chin, coin</i>

The probabilities, of course, represent the squares of the amplitudes given by Schrödinger’s wave equation  $\Psi$ :

$$|\Psi\rangle = 1/\sqrt{14} |101\rangle + 3/\sqrt{14} |100\rangle + 2/\sqrt{14} |010\rangle$$

Prediction in analogical modeling will also require each supracontext to be linked to actual exemplars.

### Measuring uncertainty in terms of disagreement

The normal approach to measuring uncertainty has been to use Shannon’s “information”, which is equivalent to the entropy of classical statistical mechanics. This measure is a logarithmic measure (of the form  $\sum p \log p$ , where  $p$  is the probability of a particular outcome). Shannon’s uncertainty is equivalent to the number of yes-no questions needed (on the average) to determine the outcome. The natural interpretation of this measure is that one gets an unlimited number of chances to discover the correct outcome, an unreasonable possibility for a psychologically based theory of behavior. Furthermore, the entropy for continuous probabilistic distributions is always infinite. This last property forced Shannon to come up with an artificial definition for the entropy of a continuous distribution. (See the discussion in Sections 1.11 and 3.8 of *Analogy and Structure*, Skousen 1992:30–37, 89–91.)

In Chapters 1–3 of *Analogy and Structure* (written in 1983, published in 1992), Skousen developed a quadratic measure of uncertainty called disagreement. This measure was applied to language behavior in *Analogical Modeling of Language* (written in 1987 and published in 1989). The measure of disagreement is the probability of two randomly chosen occurrences disagreeing in outcome (namely,  $1 - \Sigma p^2$ , where once more  $p$  is the probability of an outcome). There is a corresponding measure of agreement, namely the probability of agreement in outcome for two randomly chosen occurrences (that is,  $\Sigma p^2$ ). The natural interpretation of these quadratic measures is that one gets a single chance to guess the correct outcome. Further, the agreement density for a continuous probabilistic distribution  $f(x)$  is easily and naturally defined as  $\int f^2(x) dx$ . This measure of agreement almost always exists. In fact, it is a much better measure of variation for a continuous distribution than the traditional variance (Skousen 1992: 83–84).

This same quadratic measure of agreement is found in Schrödinger's wave equation as  $\int |\psi(x)|^2 dx$ . In order to get an overall probability of one, the integral over the entire space is normed, but still it is the squaring function that is used to determine the probability of a subspace. Analogical modeling uses this squaring function to measure the agreement density for a continuous probability distribution (see Chapter 3 of *Analogy and Structure*, Skousen 1992: 71–91). If Schrödinger's wave equation is a real function (rather than the more general case allowing complex functions), we get the same precise formulation for the density agreement found in *Analogy and Structure*, but without the norming (namely,  $\int \psi^2(x) dx$ ).

## Reversible operators

Having determined that there seems to be some extraordinarily close connections between analogical modeling and quantum computing, we turn to how we might define appropriate quantum operators for determining the analogical set of homogeneous supracontexts.

In designing a quantum computational system for analogical modeling, every operator meets the following two requirements:

1. *simultaneity*: each operator must be defined so that it can apply simultaneously to each of the  $2^n$  supracontexts;
2. *reversibility*: each operator must be reversible.

The first requirement allows us to take advantage of the simultaneity of quantum computing. The second requirement basically means that no erasure of data is permitted prior to observation of the system (that is, prior to decoherence of the superpositioned supracontexts). Each data occurrence, after being read, must be kept.



Any computational result must be recoverable, and by keeping all the input data, we insure recoverability.

Let us consider what we mean by a reversible operator. (The discussion in this section is for readers unfamiliar with quantum computing. The examples follow the explication in Berman, Doolen, Mainieri, & Tsifrinovich 1998:51–58.) The basic idea is that after an operator has applied, we are able to determine from the final (or output) state what initial (or input) state it came from. This requirement of recoverability basically means that there is a unique one-to-one connection between inputs and outputs, that no mergers or splits occur, only a shifting (or renaming, so to speak) of representations.

One clear example of a reversible operator is negation. An  $n$ -gate (where the  $n$  stands for negation) is reversible because we simply switch or flip the polarity of a state  $a$  (true to false and false to true). In the following listing (Table 6),  $a_i$  represents the initial state of  $a$ , while  $a_f$  represents the final state of  $a$ .

Table 6.  $n$ -gate

$a_i$	$a_f$
0	1
1	0

So given a final state  $a_f$  of 0 (false), we know that  $a_i$  was 1 (true); similarly,  $a_f = 1$  implies that  $a_i = 0$ .

On the other hand, the *and*-operator is not reversible. With an *and*-gate, the final state  $c_f$  is true (or 1) only if  $a_i$  and  $b_i$  were both true (or 1). If the final state is false (or 0), then there are three possible sets of initial states (00, 01, or 10), and we do not know which set of initial states produced the false output (Table 7).

Table 7. *and*-gate

$a_i$	$b_i$	$c_f$
0	0	0
0	1	0
1	0	0
1	1	1

In quantum computing, however, we can construct a reversible gate that can be used as an *and*-gate. We do this by constructing what is called a *control-control-not* gate (or *ccn*-gate, for short). In this system, we switch the polarity of an initial state  $c_i$  only if two other initial states  $a_i$  and  $b_i$  are each true. The initial states  $a_i$  and  $b_i$  act as control states and  $c_i$  acts as a *not* state (thus, *control-control-not*). We get the following input-output relationships for the *ccn*-gate (Table 8).

Table 8. *ccn*-gate

$a_i$	$b_i$	$c_i$	$a_f$	$b_f$	$c_f$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

 Table 9. *ccn*-gate ( $c_i = 0$  checked)

	$a_i$	$b_i$	$c_i$	$a_f$	$b_f$	$c_f$
✓	0	0	0	0	0	0
	0	0	1	0	0	1
✓	0	1	0	0	1	0
	0	1	1	0	1	1
✓	1	0	0	1	0	0
	1	0	1	1	0	1
✓	1	1	0	1	1	1
	1	1	1	1	1	0

For this reversible gate, there are eight possible sets of initial states and eight possible sets of final states. For the first six cases, the set of final states is identical to the set of input states (thus  $000 \rightarrow 000$ ,  $001 \rightarrow 001$ ,  $010 \rightarrow 010$ ,  $011 \rightarrow 011$ ,  $100 \rightarrow 100$ ,  $101 \rightarrow 101$ ). For the last two cases, we simply switch the polarity of the  $c$  state (thus  $110 \rightarrow 111$  and  $111 \rightarrow 110$ ). This results in a unique one-to-one function between all the sets of states. No information is lost, and from every set of output states we can determine the unique set of input states from which it was derived. We also emphasize here that with a *ccn*-gate the two control states  $a$  and  $b$  make no change whatsoever. In a sense, these two states represent labels.

Now from this *ccn*-gate, we can define a reversible *and*-operator by considering only those cases where the initial state  $c_i$  equals zero. Given the entire *ccn*-gate, we mark these four cases with a check mark (Table 9).

If we isolate these four cases where  $c_i = 0$ , we can see that we have the equivalent of an *and*-gate (Table 10).

The basic difference between a non-reversible *and*-gate and a reversible *ccn*-gate acting as an *and*-gate is that in the reversible gate the input states  $a$  and  $b$  are carried over identically as output states. In other words, the initial information about the states  $a$  and  $b$  is kept intact in the reversible gate.

Table 10. *and*-gate (a *ccn*-gate with  $c_i = 0$ )

	$a_i$	$b_i$	$c_i$	$a_f$	$b_f$	$c_f$
✓	0	0	0	0	0	0
✓	0	1	0	0	1	0
✓	1	0	0	1	0	0
✓	1	1	0	1	1	1

Reversibility essentially requires that we have to keep track of the input. Richard Feynman, one of the first who proposed applying quantum mechanics to computing, realized that reversibility meant that the input would be reproduced along with the output at the end of the computation (as noted by Richard Hughes in Hey 1999: 196):

But note that input data must typically be carried forward to the output to allow for reversibility. Feynman showed that in general the amount of extra information that must be carried forward is just the input itself.

This result is of great significance for analogical modeling and, in fact, for all exemplar-based systems – namely, reversibility leads to exemplar-based systems. If some form of quantum computing is used for language prediction, then all the exemplars used in a computation must be recoverable (at least up until decoherence). Quantum computation of any language-based system will therefore be an exemplar-based one, even if the system ends up acting as a neural net or as a set of rules!

### Quantum analogical modeling

Within analogical modeling, a supracontext is heterogeneous whenever any sub-contextual analysis of that supracontext leads to an increase in disagreement (Skousen 1989: 23–37). It turns out that this decision procedure is equivalent to the most powerful test possible. However, by introducing imperfect memory (equal to one-half), the power of the test can be reduced to standard statistical testing (Skousen 1998: 247–250).

This single decision procedure can be re-interpreted so that no mathematical calculation is ever involved; not even a measurement of disagreement between occurrences is necessary. This reworking of the procedure for determining homogeneity was discussed in both *Analogical Modeling of Language* and *Analogy and Structure* (Skousen 1989: 33–35 and 1992: 295–300). There it was shown that there are two types of homogeneous supracontexts: (1) the supracontext is deterministic in behavior (only one outcome occurs); (2) if the supracontext is non-deterministic, all its non-deterministic behavior is restricted to a single subcontext (or subspace). In the original algorithm for analogical modeling, testing for the

second type of supracontext required the program to do a layered comparison between adjacent levels of supracontexts (that is, between supracontexts representing a difference of one variable). Such an algorithm guaranteed an exponential explosion in running time.

More seriously, from a quantum computing perspective, such an algorithm could never be re-interpreted in terms of reversible operators applying simultaneously to all the supracontexts at once. By shifting the perspective to quantum computing, Skousen was able to discover that by keeping track of only two factors for a supracontext (the first outcome and the first intersect, to be explained in the next section), homogeneity could be determined by reversible operators applying simultaneously. Moreover, the original algorithm initially assigned each occurrence to the supracontext closest to the given context. Within the quantum algorithm, each occurrence is simultaneously assigned to every supracontext that can possibly include the occurrence. This simultaneity avoids the “trickle-down” effect of the original layered algorithm, which also contributed to the exponential running time of the original approach.

### Quantum computing of analogical modeling

We now see how the principles of quantum computing can be used to solve the exponential problems (in both memory and running time) for analogical modeling. This demonstration will be done in terms of the simple example from section 2.2 of *Analogical Modeling of Language* (Skousen 1989:23–37). The dataset there has five occurrences, each specified by three variables (composed of the numbers 0, 1, 2, 3) and an outcome, either *e* or *r*:

<b>dataset</b>	310e	$m(= 5)$
	032r	
	210r	
	212r	
	311r	

We let  $m$  represent the number of accessed occurrences in the dataset. We will assume that these five occurrences were randomly selected from a larger database at a level of imperfect memory of one-half. This corresponds to the idea that data occurrences are accessed through, say, a spin-up state (given two equally probable quantum states, spin-up and spin-down, for database access).

We make predictions in terms of a given context. We let  $n$  stand for the number of variables found in the given context. In the example from *Analogical Modeling of Language*, the given context is 312:

<b>given context</b>	312	$n(= 3)$
----------------------	-----	----------

No outcome ( $e$  or  $r$ ) is specified for the given context since that is what we are trying to predict. Our task then is to predict the outcome, either  $e$  or  $r$ , in terms of the three variables  $312$ . In this example,  $n$  is three.

For this given context, we now define  $2^n (= 2^3 = 8)$  supracontexts by means of  $n (= 3)$  qubits. The supracontexts specify a powerset – namely, all the possible groupings of variables that can be theoretically used to predict the outcome for the given context. Initially, each of these  $n (= 3)$  qubits are equally assigned to two possible random states, one or zero. For a given supracontext, if a variable is assigned a one (1), this means that that variable is used to help define the contents of that supracontext. On the other hand, a zero (0) means that that variable will be completely ignored for that supracontext.

For the given context  $312$ , we therefore have the following  $2^n (= 2^3 = 8)$  supracontexts:

<b>supracontexts</b>	111	$\exp(n) = 2^n (= 2^3 = 8)$
	110	
	101	
	011	
	100	
	010	
	001	
	000	

The supracontext  $110$ , for example, means that the first two variables will be considered, but the third one will be ignored.

As we read each data occurrence, we determine its intersect with the given context. For instance, the first data occurrence is  $310e$ . When compared with the given context  $312$ , we see that the first two variables agree, but the last one does not. The corresponding intersect for  $310e$  and  $312$  is therefore  $110$ . For our five data occurrences ( $310e$ ,  $032r$ ,  $210r$ ,  $212r$ , and  $311r$ ), we have the following five intersects:

<b>intersects</b>	110	$310e \ \& \ 312$	$m(= 5)$
	001	$032r \ \& \ 312$	
	010	$210r \ \& \ 312$	
	011	$212r \ \& \ 312$	
	110	$311r \ \& \ 312$	

For each data occurrence, we also record its outcome, whether it is  $e$  or  $r$ .

For each supracontext, we need to determine certain kinds of information, but only using reversible operators that can simultaneously apply to all of the  $2^n$  supracontexts. In order to derive the analogical set (a superposition of all the possible supracontexts), we determine the following information as each data occurrence is read:

- |                         |                           |
|-------------------------|---------------------------|
| 1. include              | 5. first intersect        |
| 2. sum                  | 6. plurality of intersect |
| 3. first outcome        | 7. heterogeneity          |
| 4. plurality of outcome | 8. amplitude              |

In each case, we assign a qubit or a register of qubits to store this information for the supracontexts.

We first discuss how we determine the *include* qubit. As we read each data occurrence, we need to determine which of the  $2^n (= 2^3 = 8)$  supracontexts the occurrence can be assigned to. We do this by defining a register of  $n (= 3)$  qubits, which we refer to as the *contain* register. Initially, each qubit in this register is assigned a one, but when a data occurrence is read, some of these ones will be changed to zeros, depending on the intersect for that data occurrence. From this evolved *contain* register of qubits, we can then determine whether we include the occurrence in each of the supracontexts.

We start out then by determining which supracontexts the first data occurrence (*310e*) will be included in. We assign an individual *include* qubit to *310e*, initially set to ones (Table 11).

As already noted, the intersect for *310e* is *110*. In our representation (Table 11), the intersect *110* is placed under the data occurrence *310e*, but for convenience' sake we also take the intersect of the data occurrence being currently considered and place it right above the eight supracontexts. The intersect *110* is used to determine which supracontexts will include the data occurrence *310e*. This occurrence *310e* should be contained in four supracontexts: *110*, *100*, *010*, and *000*. We determine which supracontexts contain this data occurrence by applying the following reversible operator for each of the  $n (= 3)$  variables:

**contain**

for  $i = 1$  to  $n$

if *intersect* [ $i$ ] = 0 and *supracontext* [ $i$ ] = 1

then *contain* [ $i$ ] = 0

This operator means that when we compare *110* (the intersect for *310e*) with the eight supracontexts, we need only deal with the intersect variables that are zero (0). Now if the corresponding supracontextual variable is a one (1), we change the corresponding *contain* variable from its initial one (1) to zero (0). Reversibility is obtained because we do not change the actual supracontextual specifications, but instead use the *contain* register as “work space”.

Applying this operator simultaneously to all eight supracontexts gives us the following evolution in the *contain* register (Table 12).

Now if for a given supracontext the *contain* register has only ones, then the data occurrence *310e* will be contained within that supracontext. After applying the *contain* operator three times (once for each variable), we correctly get *111* for

Table 11.

	supracontext	contain	include
data occurrence			310e
intersect	110		110
	111	111	1
	110	111	1
	101	111	1
	011	111	1
	100	111	1
	010	111	1
	001	111	1
	000	111	1

Table 12.

	supracontext	contain	include
data occurrence			310e
intersect	110		110
	111	111→110	1
	110	111→111	1
	101	111→110	1
	011	111→110	1
	100	111→111	1
	010	111→111	1
	001	111→110	1
	000	111→111	1

the supracontexts *110*, *100*, *010*, and *111*. The occurrence *310e* will therefore be included in these four supracontexts, but not the other four.

To determine the actual *include* qubit for a data occurrence, we need to use the *include* operator  $n$  ( $= 3$ ) times, once for each qubit in the *contain* variable (that is, once for each variable):

```

include
  for  $i = 1$  to  $n$ 
    if contain [ $i$ ] = 0
      then include = 0

```

In the example of *310e*, of course, only the third qubit of the *contain* register has any zeros. So by using the *include* operator, the *include* qubit for *310e* becomes correctly set (Table 13).

Table 13.

	supracontext	contain	include
data occurrence			310e
intersect	110		110
	111	110	0
	110	111	1
	101	110	0
	011	110	0
	100	111	1
	010	111	1
	001	110	0
	000	111	1

Table 14.

	supracontext	contain	include
data occurrence			310e
intersect	110		110
	111	110→111	0
	110	111→111	1
	101	110→111	0
	011	110→111	0
	100	111→111	1
	010	111→111	1
	001	110→111	0
	000	111→111	1

In order to continue using the *contain* register to determine the *include* qubit for the next data occurrence, we need to reset the *contain* register to all ones. We do this by reversely applying the *contain* operator:

**reverse contain**

for  $i = 1$  to  $n$

if *intersect* [ $i$ ] = 0 and *supracontext* [ $i$ ] = 1

then *contain* [ $i$ ] = 1

Applying this reversed operator, we get the original initial state for the *contain* register (Table 14).

Before we read the next data occurrence, we determine the amplitude for each of the possible supracontexts at this stage of the quantum evolution. To do this, we set up a number of qubit registers that designate the following information for each supracontext: *the sum, the first outcome, the plurality of the outcomes, the first intersect, the plurality of the intersects, the heterogeneity, and the amplitude.*



Table 15.

supracontext	sum	outcome		intersect		hetero	ampl
		1st	plur	1st	plur		
111	0	–	0	–	0	0	0
110	0	–	0	–	0	0	0
101	0	–	0	–	0	0	0
011	0	–	0	–	0	0	0
100	0	–	0	–	0	0	0
010	0	–	0	–	0	0	0
001	0	–	0	–	0	0	0
000	0	–	0	–	0	0	0

Initially, prior to considering any data occurrence, these qubit registers are all equally assigned zeros (Table 15).

After we have determined which supracontexts include a particular data occurrence, we then apply the following operators simultaneously to each supracontext. In each case, we give the same name to the operator as the name of the qubit register that stores the result:

**sum**

if *include* = 1 for the current data occurrence  
 then increment *sum* by one (that is,  $sum = sum + 1$ )

**first outcome and plurality of outcome**

if *first outcome* is empty  
 then store the outcome of the data occurrence in *first outcome*  
 otherwise (*first outcome* is filled)  
 set *plurality of outcome* equal to one

**first intersect and plurality of intersect**

if *first intersect* is empty  
 then store the intersect of the data occurrence in *first intersect*  
 otherwise (*first intersect* is filled)  
 set *plurality of intersect* equal to one

**heterogeneity**

if both *plurality of outcome* and *plurality of intersect* equal one  
 then set *heterogeneity* equal to one

**amplitude**

if *heterogeneity* = 1  
 then *amplitude* = 0  
 otherwise (*heterogeneity* = 0)  
 set *amplitude* equal to *sum*

Three of these registers can be represented by a single qubit (namely, *plurality of outcome*, *plurality of intersect*, and *heterogeneity*). The others need to contain specific qubit representations of various information:

<i>sum</i>	zero or a positive integer
<i>first outcome</i>	an outcome
<i>first intersect</i>	an $n$ -bit representation
<i>amplitude</i>	zero or a positive integer

It should be noted that each of these could be accessed by a single qubit plus some associated informational register whenever the qubit is set to one:

<i>sum</i>	0 if there are no occurrences 1 if there is at least one occurrence register gives <i>sum</i>
<i>first outcome</i>	0 if no (first) outcome has yet been found 1 if a first outcome has been found register gives <i>first outcome</i>
<i>first intersect</i>	0 if no (first) intersect has yet been found 1 if a first intersect has been found register gives <i>first intersect</i>
<i>amplitude</i>	0 if there is no amplitude 1 if there is an amplitude register gives <i>amplitude</i> (the same integer as in <i>sum</i> )

It should also be noted here that certain states, once reached for a given supracontext, are not changed throughout the evolution of the superpositioned system (up through decoherence or observation). Suppose we use the single-qubit system, as just described. Then whenever any of the following qubits is set to one, that qubit and any associated register will never be changed as long as the superposition is maintained: *first outcome*, *plurality of outcome*, *first intersect*, *plurality of intersect*, and *heterogeneity*. The value for *sum* for a given supracontext, on the other hand, will never decrease. The value for *amplitude* will also never decrease except when heterogeneity is achieved, in which case the amplitude will be immediately reduced to zero. And from then on, the amplitude for this supracontext will always remain at zero.

As long as we keep track of all the data occurrences in the dataset, all these operators are reversible. This reversibility basically means that our system must be an exemplar-based system of prediction if we are going to use quantum computing to determine the analogical set for a given context.

We now read, one at a time, the five data occurrences (*310e*, *032r*, *210r*, *212r*, and *311r*). For each data occurrence, we first compare it with the given context *312*

and determine the intersect for that occurrence, then apply the sequence of operators (*contain*, *include*, *sum*, *first outcome*, *plurality of outcome*, *first intersect*, *plurality of intersect*, *heterogeneity*, and *amplitude*) and finally at the end of the sequence reverse the *contain* register (that is, apply the operator *reverse contain*) before reading the next data occurrence. The results of reading these data occurrences and applying the operators appear in Figure 1.

In quantum mechanics the values of the amplitudes are systematically adjusted so that their squares sum to one. But as already pointed out, such norming procedures are the result of specifying that probabilities are real numbers from 0 to 1. In analogical modeling, there are no underlying probabilities, only occurrences and pointers to occurrences. Under conditions of imperfect memory, analogical modeling does produce probabilistic behavior, but without directly learning probabilities or using them. The amplitude for a homogeneous supracontext is directly proportional to the number of occurrences for that supracontext. Its probability of being selected is directly proportional to the number of pointers to occurrences in that supracontext – which is the square of the number of occurrences.

This squaring occurs in quantum computing whenever decoherence occurs. But in quantum analogical modeling, the squaring does not involve mathematical calculation. Instead, it is the result of selecting from all the homogeneous supracontexts one of the pointers to occurrences. We do not select an occurrence itself; that kind of selection would lead to setting the probability of predicted outcome as proportional to the amplitude. Instead, we select a pointer to an occurrence, which gives the probability of predicted outcome as proportional to the amplitude squared.

Since the supracontexts are the quantum states, decoherence is equivalent to observing one of the homogeneous supracontexts, then selecting one of the pointers to occurrences in that supracontext. Nonetheless, it is worth noting that one could directly select one of the pointers to occurrences in any of any of the homogeneous supracontexts and get the same results – namely, the proportional probability of random selection defined by the frequency squared. Furthermore, since we are keeping track of all the data occurrences, we do not really need to keep track of the sum and amplitude per se, only the information that determines the heterogeneity of each supracontext.

This view of decoherence rejects Shannon's unbounded measure of uncertainty, which allows an unlimited number of yes-no questions to guess the correct outcome. Analogical modeling allows only one guess and is equivalent to a measure of simple disagreement between pairs of occurrences. Analogical modeling thus looks at behavior in terms of events and connections between events (that is, as data occurrences and pointers between those occurrences). In analogical modeling, this measure of disagreement thus shows up directly whenever observation or decoherence occurs.

given context: 312		initial state (no data occurrences read yet)				supracontext				include (non-read   read)				sum		outcome		intersect		hetero		ampl	
		311r	212r	210r	032r	310e																	
		contain	include (non-read   read)																				
111		111	1	1	1	1																	
110		111	1	1	1	1																	
101		111	1	1	1	1																	
011		111	1	1	1	1																	
100		111	1	1	1	1																	
010		111	1	1	1	1																	
001		111	1	1	1	1																	
000		111	1	1	1	1																	
intersect (after 1st data occurrence read)																							
310 & 312 = 110																							
supracontext		contain				include (non-read   read)				sum		outcome		intersect		hetero		ampl					
110		111	1	1	1	1																	
111		111 → 110	1	1	1	1																	
110		111 → 111	1	1	1	1																	
101		111 → 110	1	1	1	1																	
011		111 → 110	1	1	1	1																	
100		111 → 111	1	1	1	1																	
010		111 → 111	1	1	1	1																	
001		111 → 110	1	1	1	1																	
000		111 → 111	1	1	1	1																	
reverse contain																							

Figure 1.

intersect (after 2nd data occurrence read) 032 & 312 = 001		include (non-read   read)		sum	outcome 1st pl	intersect 1st pl	hetero	ampl
supracontext	contain	311r	212r 210r	032r 001	310e 110			
001								
111	111→001	1	1	0	0	-	0	0
110	111→001	1	1	0	1	e 0 110	0	1
101	111→011	1	1	0	0	-	0	0
011	111→101	1	1	0	0	-	0	0
100	111→011	1	1	0	1	e 0 110	0	1
010	111→101	1	1	0	1	e 0 110	0	1
001	111→111	1	1	1	0	r 0 001	0	1
000	111→111	1	1	1	1	e 1 110	1	0
reverse contain								
intersect (after 3rd data occurrence read) 210 & 312 = 010		include (non-read   read)		sum	outcome 1st pl	intersect 1st pl	hetero	ampl
supracontext	contain	311r	212r 210r	032r 001	310e 110			
010								
111	111→010	1	1	0	0	-	0	0
110	111→011	1	1	0	1	e 0 110	0	1
101	111→010	1	1	0	0	-	0	0
011	111→110	1	1	0	0	-	0	0
100	111→011	1	1	0	1	e 0 110	0	1
010	111→111	1	1	1	1	e 1 110	1	0
001	111→110	1	1	1	0	r 0 001	0	1
000	111→111	1	1	1	1	e 1 110	1	0
reverse contain								

Figure 1. (continued)

intersect (after 4th data occurrence read)		include (non-read   read)		sum	outcome	intersect	hetero	ampl
supracontext	contain	311r	212r 011	210r 010	032r 001	310e 110	1st pl	1st pl
011								
111	111→011	1	0	0	0	0	-	0
110	111→011	1	0	0	0	1	e	0
101	111→011	1	0	0	0	0	-	0
011	111→111	1	1	0	0	0	r	0
100	111→011	1	0	0	0	1	e	0
010	111→111	1	1	1	0	1	e	1
001	111→111	1	1	0	1	0	r	0
000	111→111	1	1	1	1	1	e	1
<b>reverse contain</b>								
intersect (after 5th data occurrence read)		include (non-read   read)		sum	outcome	intersect	hetero	ampl
supracontext	contain	311r	212r 011	210r 010	032r 001	310e 110	1st pl	1st pl
110								
111	111→110	0	0	0	0	0	-	0
110	111→111	1	0	0	0	1	e	1
101	111→110	0	0	0	0	0	-	0
011	111→110	0	1	0	0	1	r	0
100	111→111	1	0	0	0	2	e	1
010	111→111	1	1	1	0	1	e	1
001	111→110	0	1	0	1	0	r	0
000	111→111	1	1	1	1	1	e	1
<b>reverse contain</b>								

Figure 1. (continued)

Table 16.

	include					sum	hetero	ampl	prob
	<i>311r</i>	<i>212r</i>	<i>210r</i>	<i>032r</i>	<i>310e</i>				
	110	011	010	001	110				
111	0	0	0	0	0	0	0	0	0
110	1	0	0	0	1	2	0	2	4
101	0	0	0	0	0	0	0	0	0
011	0	1	0	0	0	1	0	1	1
100	1	0	0	0	1	2	0	2	4
010	1	1	1	0	1	4	1	0	0
001	0	1	0	1	0	2	0	2	4
000	1	1	1	1	1	5	1	0	0

Table 17.

	include					ampl	prob	pointers	
	<i>311r</i>	<i>212r</i>	<i>210r</i>	<i>032r</i>	<i>310e</i>			<i>e</i>	<i>r</i>
	110	011	010	001	110				
110	1	0	0	0	1	2	4	2	2
011	0	1	0	0	0	1	1	0	1
100	1	0	0	0	1	2	4	2	2
001	0	1	0	1	0	2	4	0	4

In our example, after reading the five data occurrences, we have two (non-occurring) supracontexts with no occurrences (where the sum equals zero), two heterogeneous supracontexts, and four occurring (non-zero) homogeneous supracontexts. When observation takes place, we randomly select one of these four non-zero homogeneous supracontexts in proportion to their number of pointers to occurrences (Table 16).

For each of the occurring homogeneous supracontexts, we can readily determine how many pointers point to each of the two possible outcomes, *e* and *r* (Table 17).

Thus the chances of selecting the outcome *e* (that is, the chances of selecting a pointer to an occurrence having the *e* outcome) is 4 (= 2 + 2), while the chances of selecting the outcome *r* (that is, the chances of selecting a pointer to an occurrence having the *r* outcome) is 9 (= 2 + 1 + 2 + 4). The probability of the *e* outcome is therefore 4/13 (≈ 0.31), and the probability of the *r* outcome is 9/13 (≈ 0.69). These are the same results derived in Section 2.2 of *Analogical Modeling of Language* (Skousen 1989:23–37). The approach there, however, is based directly on the principle of minimizing the quadratic measure of disagreement. The quantum

computational approach considers the plurality of outcomes and intersects, but derives the very same analogical set.

We can see from the superposition of  $2^n$  supracontexts that the exponential explosion of analogical modeling is reduced to a polynomial function of  $n$ . For each of  $m$  data occurrences accessed from the database, we will need a single *include* qubit. In terms of memory requirements, quantum analogical modeling will require a linear qubit size of  $O(m + n)$ . On the other hand, the required running time is  $O(m \cdot n)$ , a multiplicative function. However, for a set number of data occurrences, the running time will be a linear function of  $n$ , the number of variables. These results provide the tractability we need for a viable exemplar-based approach to language prediction.

David Eddington, in a preface to his paper “Analogy and the Dual-Route Model of Morphology” given at the Conference on Analogical Modeling (the paper is published as Eddington 2000), has compared analogical modeling and its problem with exponentiality to a heavyweight boxer, very slow but powerful. However, if quantum computing can be applied to analogical modeling, we may have a heavyweight that is exponentially faster than anyone conceived of. Up to this time, we have perhaps worried too much about the exponential explosion, as if this were a problem that must be solved by any other means. Quantum computing suggests that we treat the exponentiality of analogical modeling as inherent. Instead of trying to avoid the exponential explosion, we should embrace it!

## Analogical quantum mechanics

It is also worth noting that analogical modeling may provide an interpretative model for quantum mechanics itself. As many have noted, the problem with quantum mechanics is that it is a formalism in search of an interpretation (see Cushing 1998:271–355, especially Chapter 23).

Analogical modeling does not actually posit underlying probabilities – there are no inherent probabilities. Instead, analogical modeling proposes occurrences (or events) and pointers (or connections) between occurrences. The notion of agreement and disagreement between occurrences leads to a natural measure of (un)certainly, one that directly models the linear/squared relationship of amplitudes and probabilities in quantum mechanics. The superpositioned supracontexts in analogical modeling, however, keep track of occurrences, not amplitudes per se. Decoherence leads to selecting a pointer to an occurrence. The probabilities are the result of selecting a pointer to an occurrence. Furthermore, the predictions are based on a single observation, as is analogical modeling (especially given its measure of disagreement instead of Shannon’s information, which permits any number



of observations). The norming of probabilities is not inherent to quantum mechanics. The real question is whether the results are probabilistic. Setting a norm on the probability measure is merely a mathematical convention. If one wishes, one can continually norm the amplitudes to obtain an observed probability between zero and one.

## Acknowledgment

I wish to thank members of the Analogical Modeling Research Group for their comments on the ideas of this paper: Dil Parkinson, Deryle Lonsdale, Theron Stanford, and Don Chapman.

## Notes

\* Analogical Modeling Research Group Report QAM:rjs000825. An earlier version of this paper was posted as a preprint to <<http://arXiv.org>> under the heading of quantum physics, paper number 0008112, on 28 August 2000.

## References

- Berman, Gennady P., Gary D. Doolen, Ronnie Mainieri, & Vladimir I. Tsifrinovich (Eds.). (1998). *Introduction to quantum computers*. Singapore: World Scientific.
- Cushing, James T. (1998). *Philosophical concepts in physics: The historical relation between philosophy and scientific theories*. Cambridge: Cambridge University Press.
- Daelemans, Walter, Steven Gillis, & Gert Durieux (1997). Skousen's analogical modeling algorithm: a comparison with lazy learning. In D. Jones & H. Somers (Eds.), *New methods in language processing* (pp. 3–15). London: University College Press.
- Eddington, David (2000). Analogy and the dual-route model of morphology. *Lingua*, 110, 281–298.
- Hey, Anthony J. G. (Ed.). (1999). *Feynman and computation: Exploring the limits of computers*. Reading, MA: Perseus Books.
- Lo, Hoi-Kwong, Sandu Popescu, & Tom Spiller (Eds.). (1998). *Introduction to quantum computation and information*. Singapore: World Scientific.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1998). Natural statistics in language modeling. *Journal of Quantitative Linguistics*, 5, 246–255.
- Williams, Colin P., & Scott H. Clearwater (1998). *Explorations in quantum computing*. New York: Springer-Verlag.

PART VII

## Appendix



## CHAPTER 14

# Data files for Analogical Modeling

Deryle Lonsdale

### Introduction

This paper is a discussion of how to prepare data for use by the Analogical Modeling (AM) system. Given the tutorial aspects of the paper, it assumes that the reader is already familiar with the overall AM approach, and this paper will therefore not discuss the background of the theory (see Skousen's overview article in this volume). Similarly, it will not give the specifics on how to run the program itself. (This information is found in the following paper by Parkinson, also in this volume.) Finally, this paper will not compare the AM approach itself with other types of data-oriented systems.

Instead, the following topics will be covered:

- the nature of data processed by the AM system
- various means of representing AM data
- special conventions used in encoding AM data

### What is data?

We begin with a general discussion of data. Data can be thought of as information that is helpful in making a decision. We seek data on current and predicted weather trends when deciding when to schedule a picnic or golf game. Data on recent performance of a given mutual fund is useful when deciding where to invest money. Information in the form of past evaluations of a course or an instructor can be invaluable to students interested in enrolling in a given college course.

The nature of the data used in decision-making varies according to the type of problem to be addressed. In many cases, as in the previous examples, numeric or arithmetic data (in the form of percentages, temperatures, or values chosen from a scale) are useful for decisions. On the other hand, sometimes data might not be

very amenable to characterization in terms of a number. In such cases, we sometimes categorize data or group it into different classes that can then be described with a label. For example, the color of each crayon in a box of crayons may not be described very well numerically; instead, it is more useful to use commonly-defined category labels (or terms) for colors to describe the color of a given crayon: red, green, blue, etc. In this discussion we will consider both continuous scalar descriptions of data (the first numeric kind mentioned) and discrete, categorical descriptions of data (the second kind mentioned here).

Items of data are often not isolated; rather, data is often presented as a collection, series, or list of several related properties that, taken together, describe some interesting item. For example, the description of a book might contain several items of data: its title, the name of its author(s), its publisher, its publication year, the number of pages, and so on. Numeric data can be used for the publication year and number of pages, whereas category labels (i.e. names) would be used for the title, author, and publisher.

Clearly, the type of data to be used in computation should be commensurate with the operations to be performed on (or with) the data. Category labels such as “hot”, “cool”, “frigid” may be useful for some purposes in characterizing the temperature of certain days, but clearly they would be inadequate if one wished to calculate the average temperature for these days. An important aspect of any computational approach, including AM, is to decide how best to characterize the data in a way that is amenable to treatment by algorithmic processes used by computers.

## Examples of data

In this section we will examine a few types of widely used data to appreciate the types of information typically encoded for various purposes. They come from a repository of datasets that have been compiled for use by people who develop and evaluate machine learning systems and do research in other computational aspects of data categorization (Blake, Keogh, & Merz 1998).

The Congressional voting record is a dataset that contains information on the voting record of members of the House of Representatives in the U.S. Congress. One year’s worth of voting records, namely those taken in 1984, has been made available for use by the machine learning community. Figure 1 shows an example of the information that is contained in this dataset.

This figure mentions that there are 435 data instances; specifically, the votes recorded for 435 members of the House of Representatives (267 members of the Democratic Party and 168 members of the Republican Party). For each member, the results of 16 votes have been recorded; for each issue the House voted on, each

Number of Instances: 435 (267 Democrats, 168 Republicans)  
 Number of Attributes: 16 + class name = 17 (all Boolean valued)  
 Attribute Information:

1. Class Name: 2 (democrat, republican)
2. handicapped-infants: 2 (y,n)
3. water-project-cost-sharing: 2 (y,n)
4. adoption-of-the-budget-resolution: 2 (y,n)
5. physician-fee-freeze: 2 (y,n)
6. el-salvador-aid: 2 (y,n)
7. religious-groups-in-schools: 2 (y,n)
8. anti-satellite-test-ban: 2 (y,n)
9. aid-to-nicaraguan-contras: 2 (y,n)
10. mx-missile: 2 (y,n)
11. immigration: 2 (y,n)
12. synfuels-corporation-cutback: 2 (y,n)
13. education-spending: 2 (y,n)
14. superfund-right-to-sue: 2 (y,n)
15. crime: 2 (y,n)
16. duty-free-exports: 2 (y,n)
17. export-administration-act-south-africa: 2 (y,n)

```
n,n,y,y,y,y,n,n,y,y,n,y,y,n,y, republican.
y,n,y,y,y,y,y,n,y,n,y,n,y,y, republican.
n,y,y,n,y,u,y,n,n,y,y,n,y,n,y,y, democrat.
y,n,y,n,y,y,n,n,n,n,n,n,n,n,y, democrat.
n,y,n,y,y,n,n,n,n,y,y,y,n,n, republican.
y,y,y,n,y,y,n,y,n,n,y,n,y,n,y,y, democrat.
n,y,y,n,n,n,y,y,y,y,y,n,n,n,y,y, democrat.
n,n,n,n,y,y,y,n,n,n,y,y,y,n,y, democrat.
y,y,y,n,n,n,y,y,y,y,y,n,n,n,n,y, democrat.
n,y,y,n,n,y,y,y,n,y,y,n,y,y,n,u, democrat.
n,n,y,y,n,n,y,y,y,y,n,n,n,y,y,y, republican.
```

**Figure 1.** Features and sample data instances from the Congressional Voting Record dataset (Blake, Keogh, & Merz 1998)

member's vote (*y* for "yes" or *n* for "no") is listed. This dataset therefore consists of 435 lines, one per House member, each of which records a member's votes and the party he or she belongs to.

For example, the first data instance in the dataset consists of the line:

```
n,n,y,y,y,y,n,n,y,y,n,y,y,y,n,y, republican.
```

Read from right to left, this says that one Republican House member voted "yes" for the first issue, "no" for the second one, "yes" for the third one, and so on until the last one, for which a "no" vote was cast. The other 434 data instances are similarly encoded. This dataset would be useful in many different ways. Machine learning systems, for example, could use this type of data to arrive at a prediction (called an outcome in the AM system) for how a given House member might vote for a given issue, given his or her past voting record and comparing it to other members' votes.

Number of Instances: 101  
 Number of Attributes: 18 (animal name, 15 Boolean attributes, 2 numerics)  
 Attribute Information: (name of attribute and type of value domain)

1. animal name: Unique for each instance
2. hair Boolean
3. feathers Boolean
4. eggs Boolean
5. milk Boolean
6. airborne Boolean
7. aquatic Boolean
8. predator Boolean
9. toothed Boolean
10. backbone Boolean
11. breathes Boolean
12. venomous Boolean
13. fins Boolean
14. legs Numeric (set of values: {0,2,4,5,6,8})
15. tail Boolean
16. domestic Boolean
17. catsize Boolean
18. type Numeric (integer values in range [1,7])

```
aardvark,1,0,0,1,0,0,1,1,1,1,0,0,4,0,0,1,1
antelope,1,0,0,1,0,0,0,1,1,1,0,0,4,1,0,1,1
bass,0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,4
bear,1,0,0,1,0,0,1,1,1,1,0,0,4,0,0,1,1
boar,1,0,0,1,0,0,1,1,1,1,0,0,4,1,0,1,1
buffalo,1,0,0,1,0,0,0,1,1,1,0,0,4,1,0,1,1
calf,1,0,0,1,0,0,0,1,1,1,0,0,4,1,1,1,1
carp,0,0,1,0,0,1,0,1,1,0,0,1,0,1,1,0,4
catfish,0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,4
chicken,0,1,1,0,1,0,0,0,1,1,0,0,2,1,1,0,2
chub,0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,4
clam,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,7
```

**Figure 2.** Features and sample data instances from the Zoo Animals dataset

Another dataset commonly used by the machine learning community is the zoo dataset. As indicated in Figure 2, it consists of 101 data instances, each describing a type of animal. The relevant features (of which there are 18) include descriptions of whether the animal typically has hair, fins, feathers, lays eggs, is a predator, and so on. The outcome is the animal name.

For one final example of an interesting dataset, consider the mushroom toxicity dataset. Figure 3 shows that this dataset has 8124 instances. Each one consists of 22 features, each describing some attribute of a particular mushroom. Besides these features, each instance also has a boolean (two-valued) feature that states whether that particular mushroom is poisonous or edible.

Number of Instances: 8124  
Number of Attributes: 22 (all nominally valued)  
Attribute Information: (classes: edible=e, poisonous=p)

1. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
2. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
3. cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
4. bruises?: bruises=t, no=f
5. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
6. gill-attach: attached=a, descending=d, free=f, notched=n
7. gill-spacing: close=c, crowded=w, distant=d
8. gill-size: broad=b, narrow=n
9. gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
10. stalk-shape: enlarging=e, tapering=t
11. stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
12. stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
13. stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
14. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
15. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
16. veil-type: partial=p, universal=u
17. veil-color: brown=n, orange=o, white=w, yellow=y
18. ring-number: none=n, one=o, two=2
19. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
20. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
21. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
22. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

p,x,s,n,t,p,f,c,r,n,k,e,e,s,s,w,w,w,p,w,o,p,p,k,s,u  
e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,p,n,n,g  
e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,w,p,w,o,p,p,n,n,m  
p,x,y,w,t,p,f,c,n,n,e,s,s,w,w,p,w,o,p,p,k,s,u  
e,x,s,g,f,n,f,w,b,k,t,e,s,s,w,w,w,w,o,e,n,a,g  
e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,p,k,n,g  
e,b,s,w,t,a,f,c,b,g,e,c,s,s,w,w,w,p,w,o,p,p,k,n,m

Figure 3. Features and sample data instances from the Mushroom Toxicity dataset



## 1. Encoding data

In this section we take a closer look at how to encode data. We begin by defining terms that have been used informally above. A *dataset* is some amount of information that concerns a set or collection of objects. A *data instance* is the description of a particular object for which we have some information. Each data instance can be composed of *features* or *attributes* that each describe some property of that data instance. For example, a dataset could be composed that describes the set of automobiles in the parking lot outside a given building at a particular time. Typically, one data instance could be provided for each auto in the lot. For example, if the car I drove to work was in that lot, a data instance would be provided to account for that auto. Each auto's data instance would consist of a description of the properties of that auto. Salient features that could be used in such a data instance might be its color, year, model, whether or not it has air conditioning, the license plate number, whether or not it has a valid parking permit, and so on.

What is an AM data file? First and foremost, it is a collection of data instances that we want to process with AM. Specifically, it contains one instance per line (sometimes called the feature vector). Each of these data instances contains, in its line in the file, several features. Each of the features has its own range of possible values. In this manner it is possible to give information to the AM system about data instances. A very important aspect of a data instance is its associated outcome, which tells us what that data instance represents when it is processed by AM. When a data instance has an associated outcome, we call it a "labeled instance". Typically several dozen, hundred, or thousand related labeled instances are collected in a data file, which is then given to AM for processing.

In general terms, AM uses the data in a data file to predict the behavior for similar instances. The features in each data instance are processed, the combination of features is related to the specified outcome, and then data instances are compared and contrasted with each other. It is important that the data file be encoded consistently and correctly, so that the labeled instances can be used properly by the system. Note that the datasets just mentioned above were not specifically encoded for AM processing; however, they are consistent in the way they represent each data instance. Because this is so, it is possible to convert these datasets relatively easily into the proper AM format.

Besides the data file, another file called the configuration file is used by the AM system. This file is used to set various processing options that control how AM processes the data instances. This file also tells the program how the dataset has been encoded. A brief discussion on how data instances can be encoded will be given later in this paper.

In addition to the data instances we have just mentioned, one other type of item is input into the AM system: the test instance. This is how users can ask the

AM system to guess what the outcome should be for a given set of features. It is important to note that the test instances are encoded in exactly the same manner as the data instances are, except that the test instances need not be accompanied by an outcome.

After reading and processing all the data instances from the data file, the AM system reads all of the test instances prepared by the user (which are contained in a test file). The system then determines the probability of the various outcomes for each test instance and reports the outcome results for each of these instances.

As mentioned above, the encoding of data depends on the nature of the question that is being addressed. Since most of the work done in AM concerns natural language data, we next discuss typical problems in using AM to describe natural language. Crucial to this discussion will be how to encode the data as AM-compatible data instances.

## 2. What kinds of language data?

Now that we have discussed the properties of data and datasets, it is possible to address the types of language data that are typically processed in an AM setting.

Language data of many different types has been treated by researchers in the natural language processing, speech, and machine learning communities. For example, one might want to address a phonological issue such as: when do we pronounce the word “the” with a schwa (as in [ðə]) versus when do we pronounce it with a high front diphthong (as in [ði])? Morphological issues can also be addressed, especially when selection of an allomorph is relevant; for example, when do we say “a” vs. “an”? For an example of a syntactic issue that is relevant to questions of parsing preferences, one might model how we decide where the prepositional phrase attaches in sentences like “I saw a man with a beard” versus “I saw a man with a telescope.” Word-sense issues are commonly addressed in the area of semantics; for example, which sense of “bank” is used in the sentence “I deposited my money in the bank” versus “I fished along the north bank of the river.” Recent work in discourse acts and other pragmatic issues leads to research into such questions as what kind of communication is being used when one says “Close the door” versus “Push the <PageDown> button.” We will look at each of these kinds of data in the next few paragraphs in order to sketch how datasets in each of these areas can be constructed.

Much of the work that has been done in AM has focused on areas related to phonology, which involves the study of how sounds are interrelated and influence each other. Traditionally, there have been a few different levels of phonological analysis, most of which have been investigated to some extent in AM. The basic

unit of phonological interest is the sound segment or phoneme, which represents an individual sound that is fundamental to the language's speech patterns. In most phonological work, the International Phonetic Alphabet (IPA) or other standardized alphabet is used to represent the separate sounds in a language; fortunately for those whose language uses the Roman alphabet, many of the sound symbols will be readily recognizable. Several of the sounds use diacritics (e.g. *ʃ*, representing the "sh" sound), and a few use symbols that appear in other languages but are not used in modern English (e.g. *ð*, representing the "th" sound in the word "then"). Many people prefer to represent individual sounds in their AM datasets by only using the ASCII character set, which is restricted to the characters 0 through 9 as well as uppercase and lowercase forms of the letters A through Z and the standard punctuation marks. It is common practice in AM work to represent sounds that require a diacritic or that fall outside the ASCII set in terms of ASCII characters, effectively recoding them to conform to ASCII codes. For example, Skousen (1989) uses *S* to represent the "s" sound, and *s* to represent the "s" sound. In fact, it is not necessary to restrict oneself to the ASCII character set; one AM study succeeded in determining where word-boundary separations should take place in Thai text (which often is devoid of word-delimiting spaces); in this case the data items consisted of 16-bit characters taken from the Thai character set and encoded directly from the Thai language.

Besides the sound segment, other levels of structure are also used in phonology and lend themselves well to an AM analysis. For example, each sound segment can be described in terms of binary articulatory features that characterize how the sound is produced or what the salient properties of its sound wave are. These properties often spread to neighboring sound segments; this process is called assimilation. Accounting for assimilation and similar effects is possible when relevant features are appropriately encoded in data instances.

Another level of phonological structure often discussed in phonological research is the syllable. In many languages the syllable is crucial for determining such properties as stress or accent, vowel length, tone, and so on. Typically the syllable is sub-divided into the nucleus (which usually contains vowels), the onset (which precedes the nucleus in a syllable), and the coda (which follows the nucleus in a syllable). For example, the word "computer" can be transcribed as *kəm/pju/tər*, with three syllables; in the first, the onset consists of the sound segment "k", the nucleus is the schwa "ə", and the coda is the sound "m".

Salient phonological properties that lend themselves particularly well to an AM analysis are those that reflect regular processes, such as word-syllable stress and aspiration (seen in English in the distinction between the aspirated [p] in "poke" versus the unaspirated one in "spoke"), voicing (seen in the unvoiced [l] in "plug" versus the voiced one in "bleed"), and vowel length (long as in "bad" versus short as in "bat").

Consider the problem discussed in Skousen 1989 related to the spelling of the [h] sound at the beginning of English words. There are three ways such words can be spelled: with an *h*, as in “how”, with a *wh* as in “who”, and with a *j* as in “jicama”. To process this three-way distinction in AM, each word is encoded as a separate data item using twelve variables reflecting phonological features. Each data item has one outcome associated with it: *h*, *wh*, or *j*. A set of 821 data instances was collected, and AM did very well in determining when given and novel items should be spelled with which variant.

Another area of linguistic description commonly used in AM-based modeling involves the morphological structure of language. Morphology deals with the structure of words and usually addresses such issues as the root form of a word, its part-of-speech category (e.g. verb, noun, adjective, etc.), the affixes it may take (e.g. prefixes, suffixes, etc.) and the ways it may be used in forming compounds. For example, in English the word “fish” may be either a noun or a verb, depending on its use in a sentence or phrase. From the noun form we may form an adjective “fishy” and compounds “white-water fish”, “tropical fish”, and so on. Sometimes variant forms of an affix (called allomorphs) may be observed in a language. For example, in English the same prefix meaning “not” has four allomorphs (*il-*, *im-*, *ir-*, and *in-*) depending on the phonological properties of the first sound of the root it is added to (*illogical*, *impossible*, *irreverent*, *insubordinate*). In English, the agentive suffix varies based on various properties of the verb root: *act+or* versus *read+er*. When compounds are formed in English, sometimes words are separated with hyphens, sometimes with a space, and sometimes not at all (*topsy-turvy*, *computer screen*, and *pancake* respectively). Though not a morphological property of English, many languages divide nouns (and other types of lexical items) into various classes; for example, French divides its nouns into two genders (masculine and feminine), and German has three genders (masculine, feminine, and neuter). Sometimes the gender of a word is difficult to determine or remember, even for a native speaker; for example, the French words “antilope” and “automne” are problematic masculine and feminine words, respectively. Trying to determine which part-of-speech tag, allomorphic variant, gender, or compounding connective is appropriate in a given situation is the type of situation that has been well studied within AM. In many languages these issues become quite complex, and this volume includes papers that address morphological issues in various languages.

An example of a dataset that deals with lexical or morphemic variation concerns use of the variant forms of the indefinite determiner (*a* vs. *an*) in English. As explained in Skousen 1989, deciding when to use “a” versus “an” is usually considered a straightforward problem, though there are instances where this becomes an interesting issue. Consider how different people select either of the variants in collocations such as *a(n) hypotenuse* or *a(n) hypothetical*. In fact, there are several cases that illustrate that this is not a trivial problem; consider the following

instances extracted from proposal abstracts present in U.S. Department of Energy documents (ACL/DCI 1991):

alumino silicate fibre holding an helical wire set in grooves inner  
 e target have been determined. An ytterbium target of  $4 \text{ g/cm}^2$  ha  
 ted receptor with borohydride, an  $^3\text{H}$ -labeled alcohol is released, sugg  
 ted from the deposition chamber to a UHV chamber equipped with Auger spec  
 king the donor nitrogen atoms. An x-ray diffraction structural analysi  
 s discussed. An application of an hydrodynamic study in the North Sea  
 his value is then corrected by an magnification factor called  $K_e$  that  
 es and concludes that they are an wholly inadequate response to the  
 ron sputtering. Preparation of an Y-Ba-Cu-O film directly on MgAl{sub  
 s radiographic sign appears as an horizontal line between two soft  
 onnecting the gas supply lines an gas evacuation lines to each of the  
 . The burners were fired using a UK coal (pulverised at CERCHAR) and,  
 e system, octopus rhodopsin is an 11-cis pigment, while the photoprodu  
 low influenced the mobility of an herbicide which was adsorbed by the  
 pensation, i.e.  $\langle e/h \rangle = 0.76$ , at an hadronic energy resolution of  $\{\sigma$   
 he structure of earthen seals. An saturated environment will need to b  
 ray and gamma-ray observations an substantially underestimate the spec  
 erature in the same way as for an homogeneous dirty type II supercondu  
 great reduction in their cost, an great increase in electricity rates,  
 referred to as  $\{\alpha\}$ -phase. A eutectic exists between P and C at 12  
 information presented here in an historical perspective. 55 refs., 4  
 revious years theoretical work an space-charge dominated beam dynamics  
 nd spontaneous cytotoxicity to a established tumor cell line (18 hrs a

Clearly there are some errors as well as instances where usage of “a” or “an” would vary from speaker to speaker. If one were to code such instances in a dataset for treatment in AM, the approach as outlined in Skousen 1989 could be followed. In that dataset Skousen compiles 164 instances, each of which contains 15 variables, which are all phonological in nature. The outcome is either the word “a” or “an”.

AM datasets have also been compiled to account for higher-level aspects of language use, ones above the phonological and morphological levels. For example, Skousen (1989) discusses a dataset collected by Parkinson to account for sociologically-conditioned lexical choice in Arabic. In this dataset Skousen looked at two words that mean “my brother”, *ya'axi* and *yaxuuya*, and used these as the outcomes. There were 8 variables that encoded various sociolinguistic and discourse factors (such as the social status and gender of the speaker and of the hearer) that presumably played a contextual role in determining which variant was selected. 242 data instances, collected from the protocols of actual-use scenarios, were used by AM to model use of these words. An interesting factor that was included in this research was that frequency information on the data instances was also taken into consideration.

## General considerations

There are several questions which must be addressed when compiling datasets for use by the AM program. In this section we discuss several of these issues.

First of all is the issue of exponentiality. As with many computationally complex systems, the nature of the AM is such that its processing reaches practical limits. At the present time, for example, there is a limit of about 22 features. If data instances consist of more than this number of features, processing slows to such an extent that use of the system is not practical. Work is currently being done to see whether this exponential problem can be mitigated somewhat (see Stanford's article in this volume). In general, large numbers of data instances can be processed. The main factor that impacts performance is the number of variables that is used to encode each data instance.

Much discussion in the machine learning community involves the issue of robustness. This has to do with whether a system is resilient enough to be able to handle data that might be only partial, incorrect, or otherwise questionable. The AM approach is designed to leverage the fact that language use is often prone to errorful, inconsistent, and incomplete information. To assume that input is always complete, consistent, and correct would not reflect how language is used. In spite of these apparent data problems, the human faculty is remarkably resilient in processing language. To the extent that this can be shown in AM, a more exact model of language use is possible. In general, noisy data can and should be included in language datasets for AM processing, when such data is available.

Contextual effects are important in language use; however, the vast range of language contextual features introduces a few issues of feature selection. In AM, one seeks to select and work with variables that are closest to the locus of the phenomenon that is being investigated. For example, in a phonological dataset, features involved in phonological processing should be preferred to semantic or pragmatic features that would likely have little or no bearing on the issues at hand. The site of a morphophonological change should be described in terms of features surrounding the site, and not those so far away that they have no relevance to the site and its change. Along with the issue of feature proximity, another issue is that of feature differentiation. This means that one should choose enough features so as to distinguish the different data instances. An encoding scheme where the vast majority of feature encodings cluster closely in a small region of the space of all possibilities will not be as successful as a scheme where the features are more evenly distributed across the space of possibilities.

In summary, there are a few skills that contribute to the successful development of a dataset: the choice of the number of variables, identifying those features most relevant to the issues at hand, and being able to account for data instance

differentiation. Being able to satisfy these desiderata is an art, and is best acquired through experience.

Another issue with respect to attributes is that no continuous-valued attributes are currently supported by the system. For example, AM sees 1, 6 and 9 as different values, but 6 is not any “closer” to 9 than 1 is. In other words, there is no arithmetic or absolute-valued calculation of these values, other than the fact that they represent nonequal categories.

Finally, it has been noted that character sets for feature values don’t seem to pose a problem. As noted earlier, a recent student study encoded data instances with features from the 16-bit character set for Thai characters, in order to perform sentence boundary detection in Thai text, and the system performed well.

Note that when working with instances taken from corpora, the type/token distinction is relevant. Tokens are instances of data (e.g. Mark Twain’s *The Adventures of Tom Sawyer* has 71,370 word tokens). Types, on the other hand, are data instances with redundancy removed (e.g. *The Adventures of Tom Sawyer* has 8,018 word types). AM work often involves tokens, since frequencies are helpful and informative in many cases.

### True zeros versus nulls

The equal sign has special significance in AM. It fills one of two roles, and therefore one can choose to include or exclude it from processing. If it is excluded from processing, it doesn’t count as an active feature. When it is included in processing, it counts as an active feature. When excluded from processing, the equal sign usually stands for predictably redundant, non-occurring variables. The equal sign can therefore be used to represent incomplete data as a kind of template to signify that relevant information cannot be provided for these variable slots. It can also be useful when encoding sparse data with many variables. For more discussion of this point see Sections 3.1–3.2 of Skousen 1989.

### Data instance coding options

One other fundamental choice in the encoding of data instances is available to the user of AM. This centers around how many characters should be used to encode each feature of a given data instance.

One possibility is to use one character per feature in the data instance. For example, note this example of a data instance which has an outcome (the charac-

ter  $m$ ), and which uses thirteen features encoded in a variable, one character per feature, followed by a comment:

```
m CjSeCtRSaCs0= zemetras
```

Using this type of encoding can sometimes result in data instances that are difficult to read and understand. Another format is therefore possible, in which each feature in the data instance is represented by a word, with a space character separating the individual feature encodings. For example, the next few examples show encodings using space-delimited multi-character features, with commas separating the outcome column from the feature columns, and the features column from the comment column:

```
25dec, ham eggs milk,          hefty-breakfast
22nov, turkey yams stuffing,   what-a-feast
13feb, beans muffins soda,     winter-campout
14aug, tofu carrot-stick water, vegan-delight
```

Here we see an outcome (a date), a comma followed by three feature values delimited by spaces, and a second comma that defines the last column as the comment column.

## The test file

The test file is a type of file that is mandatory when using the AM system. Basically, it is used to ask the system to predict the outcome for each test instance in the file. It uses the same type of feature vectors as the data file, except no outcome need be specified. Note that any number of test instances can be contained in a file; the system iterates over them all, processing them one at a time.

## Run-time options and data

This section briefly mentions the different types of options that users of the AM system can select from in order to control how processing is carried out. These will be explained in further detail in Parkinson's following article (in this volume).

One option, called imperfect memory or forgetting, controls whether the system can remember and have access to every data item in the dataset. Any decimal number from 0 to 1 can be specified, which in turn is converted by the system to a percentage of items to be remembered. If forgetting is specified, an appropriate number of items are randomly selected to be forgotten.



Another value that the system takes into consideration is the frequency specification. Each type of data item in the dataset can be ordered in terms of frequency, with the most frequent listed first. Then, in running the program, we can use only the first  $n$  data items by restricting the frequency to  $n$ . In this way the program makes predictions in terms of the  $n$  most frequent data items. This approach means that each token of data item does not need to be separately listed as a unique data item in the input.

### Finding and coding data items

Usually people who work with AM have their own data which they seek to submit to an AM analysis. However, there are other resources that can be tapped as datasets for use in AM. For example, there exist many machine-learning archives whose purpose is to provide researchers with machine-readable versions of data that have been previously collected. The congressional voting, mushroom toxicity, and zoo animal datasets mentioned earlier are examples of data that can be obtained from such repositories. Other datasets useful in modeling language use can be obtained from organizations that collect and disseminate linguistic data, such as the Linguistic Data Consortium (LDC) and the Evaluation & Language resources Distribution Agency (ELDA). Finally, researchers who work in other paradigms have often collected data related to their own investigations, and are often willing to share their data with other researchers. For example, much work that has been done in connectionist (parallel distributed processing or neural network) approaches to natural language processing involve datasets that have been encoded into feature vectors which can be easily manipulated into AM data format.

Often it is useful to generate one's own datasets based on pre-existing resources. Several efforts in AM have been carried out on data items extracted from corpora. Others have used dictionaries and lexical knowledge bases to extract interesting information for data items. Generally, the manipulation of corpus and lexicon data in order to create AM-type datasets requires some programming ability. Many Unix commands are available to help in the manipulation of textual and lexical data (e.g. `awk`, `grep`, `sed`, and `tr`). A more platform-independent, though more programmatically structured approach is to use the Perl scripting or programming language, which is widely used for dataset construction.

## Sample applications

There are several sample applications that have been documented in the growing AM literature. For example, in Skousen 1989, several datasets are included, such as the one used to predict Finnish past-tense formation. In his book on NLP applications of AM, Jones (1996) presents information on how to develop datasets for the process of analogical cloning, although no complete datasets are presented in the book. There are also several published papers dealing with AM that include partial or complete datasets; most papers describe the process used in constructing data instances as well as the rationale for choosing the number and type of attributes. In the future it might be possible to collect a few of the most intuitive, commonplace, and straightforward datasets in an AM data archive.

## References

- Association for Computational Linguistics Data Collection Initiative (ACL/DCI) (1991). U.S. Department of Energy Scientific Abstracts. CD-ROM Collection 1. University of Pennsylvania, Linguistic Data Consortium (LDC).
- Blake, Catherine, Eamonn Keogh, & Chris Merz (1998). UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>
- Jones, Daniel B. (1996). *Analogical natural language processing*. London: University College London Press.
- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Skousen, Royal (1992). *Analogy and structure*. Dordrecht: Kluwer Academic Publishers.



## CHAPTER 15

# Running the Perl/C version of the Analogical Modeling program

Dilworth B. Parkinson

### Introduction

The Analogical Modeling (AM) program first appeared as Pascal code at the back of Skousen 1989. This program has undergone a minor correction and a number of revisions designed to make it run faster, more easily, and with more variables allowed. Except for the minor correction, the results given by the program are identical over the various versions.

The version described here was written in July 2001 by Theron Stanford based on an algorithm suggested by Royal Skousen. It is written in a combination of Perl and C and appears to be several orders of magnitude faster than earlier versions as well as less memory-intensive. This version has crunched through datasets with thousands of datapoints and 24 and more variables in minutes rather than hours. As currently written, 31 would be the maximum number of variables possible.

One important point to realize at the beginning is that the program does very little error checking of the data, test, and outcome files. The researcher is responsible for making sure those files are appropriately constructed. When someone brings me a bizarre looking set of results, the reason can often be traced back to typos in one of those files or to a mismatch between the configuration file and the way the files were in fact configured.

### 1. Getting and compiling the program

The assumption is that the user is running either Windows or a Unix-based operating system such as Unix itself, Linux, or Mac OS X; those running other operating systems may be able to compile the program themselves by modifying the instructions given below for Unix-based operating systems. No matter what operating system is used, Perl must be installed first. The latest version can be downloaded

from <<http://www.cpan.org>>. Windows users may find it easier to download from <<http://www.activestate.com>>. Version 5.6 or greater must be used for successful compilation and execution. All AM program files can be downloaded from the AM homepage at <<http://humanities.byu.edu/am/>>.

### 1.1 Running under Windows

The easiest way to run the AM program under Windows is to download the precompiled executable *amc.exe* and the Perl script *amc.pl* from the AM homepage. This only works, however, if Perl has been installed in the default directory, *C:\Perl*; in particular, the file *Perl56.dll* must reside in directory *C:\Perl\bin*. The program can then be run either from the command line or by double-clicking on the *amc.exe* icon, assuming all other data files are in the correct places as described below.

Windows (or DOS) users who wish to compile their own version of *amc.exe* can follow the instructions for Unix-based operating systems (given in the next section). However, be warned that the Perl library in the Windows port is not named *libPerl56.a* but *perl56.lib* instead, and that this library file has a structure different from its Unix counterpart. Commercial C compilers should have no problem linking to *perl56.lib*; users of free C compilers such as *gcc* will have to use the tools included in their environments to create their own *libPerl56.a* to link to. (The downloadable version of *amc.exe* was in fact compiled this way, using a port of the C compiler *gcc* to Windows provided by <<http://www.mingw.org/>>.)

### 1.2 Running under Unix-based operating systems

In this scenario, there is no other option but to obtain the source code and compile it oneself. Because of the differences between Unix-based operating systems, it is impossible to provide an executable that will run on all of them.

The source code consists of the following five files, which must be downloaded from the AM homepage: *amc.pl* is Perl code which parses the datasets and eventually writes the results to a file; *amc.c* is C code which uses the parsed data to create the supracontextual lattice and count pointers; *progeny.c* is C code containing two helper functions for traversing the lattice; and *amc.h* and *progeny.h* are header files necessary for compilation.

To compile, the user must find out where the Perl header files *EXTERN.h* and *perl.h* as well as the Perl library *libPerl.a* (or *libPerl56.a*) reside and modify the makefile accordingly. The document *perlembed* that comes with the Perl distribution can be helpful in finding these files. The C header file *iso646.h* must reside with the other standard C header files; if the user's system is missing this file, it should

be found and installed prior to attempting compilation. Once the makefile is set up, typing “make” at the command prompt should be sufficient. The program can then be run by typing its name (typically “amc”) at the command prompt, assuming that all other data files are in the correct places as described below and that *amc* has been given executable permission.

Successful compilation has taken place in a variety of environments, including Unix, Linux, Mac OS X, and Windows.

### 1.3 Note for users not running Windows or Unix

For those who cannot use the precompiled Windows version of *amc.exe* and cannot compile C programs – for instance, users of older Macintosh operating systems – versions of the AM program which only require Perl to be installed can be downloaded from the AM homepage. These do not have the speed of the latest versions, and they require larger memory allocations, but they can still handle datasets with thousands of datapoints and up to 24 variables.

## 2. Overview of AM program

The AM program takes one or more ‘givens’ or tests, and compares them to a set of data coded for any desired variables. The program uses the data to then predict an outcome for the test item. The program is controlled with a configuration file named *AM.config*. This file must be in the same folder or directory as the AM program file. The configuration file lists the name of the dataset and the testset and allows the user to set a number of other parameters. Once this configuration file has been filled out and saved, one simply invokes the AM program itself in the normal way.

In this paper I will be using a small dataset gathered for the prediction of Arabic plurals. The outcomes for the particular forms in this dataset are limited to two: the feminine sound plural (coded *fem*) and a particular broken plural pattern (coded *CaCaaCiiC*). The dataset encodes a number of singular nouns, along with an indication of the plural they take. So, for example, the singular *taSriiH* takes the feminine sound plural (*taSriiHaat*), while the singular *taqliid* takes the broken plural pattern (*taqaaliid*). To use the program, one needs to present it with a test item – in other words, a new Arabic singular noun – and the program will then predict a plural form for that noun from among the outcomes present in the dataset.

## 2.1 Naming convention

All files related to a particular set of data are contained in a subdirectory of the directory containing the AM program. The name of this subdirectory is then listed in the file *AM.config*. This is how the program knows what data and outcomes to consider, and what tests or ‘givens’ are to be used. The name for the sample dataset used here is *ArPlurals*.

## 3. Files provided by researcher

### 3.1 The data file

The data file must be given the name *data*. The data file consists only of the data, with no headers or other information. Each line of the data file represents one data point, and each line must end with a return, including the last line. The data are listed in the following order: Outcome, Variables, Specifier.

The outcome represents the result you are trying to predict. In the case of the Arabic dataset, the outcome would be either *fem* (representing the sound feminine plural) or *CaCaaCiiC* (representing the broken plural pattern). It makes no difference what you choose to represent the outcomes as long as they are distinctive. You may choose letter or number identifiers (A and B, 1 and 2) if you want the data to have a clean look, or you may choose something that reminds you more directly of what it is you are coding, as I did with the Arabic plural dataset. There is no practical limit to the number of outcomes a dataset may include.

The variables represent aspects of the input data that you wish to code for (see Lonsdale, this volume, for suggestions on choosing variables). In general, given the limitations of the program, one would want to choose fairly high level features that could conceivably have something to do with the outcome, although it is a mistake to limit oneself only to features already known to be important. In the Arabic plurals example, I chose to code for whether or not the root was doubled or sound, the three letters of the root, and the phonological class of each letter (labial, affricate, etc.). Since the data all have the same pattern there was no need to code for the vowels. In other phonologically oriented data, it is common to code for the vowels, for vowel length, and for aspects of the syllable structure (onset, coda, etc.). Variables for each data point must, of course, be in the same order each time.

The specifier at the end of the line is simply a reminder to the user of what the data actually codes. It is often not intuitively clear what a line of variables refers to, so the specifier helps the researcher remember what the encoded variables stand for. The AM program itself simply ignores the specifier. For the Arabic data, the

specifier is the singular form that takes the plural listed in the outcome and which is coded in the variables. Some users take advantage of the fact that the program ignores everything after the variables to add other information to each datapoint, such as frequency information.

### 3.1.1 *Formatting options*

**3.1.1.1 *Formatting with spaces only.*** There are two options for formatting a line of data, although both options follow the basic pattern mentioned above (Outcome, Variables, Specifier). In the first option, the variables must be represented by a single character or digit. They are therefore simply listed in order with no spaces in between them. The outcome and specifier can be of any length, but most users who choose this format option want their files to line up nicely, and one way to make this happen is to have the outcomes also have only one letter or digit.

Spaces are used to separate the outcome from the variable list, and the variable list from the specifier. The spaces can include simply the spacebar itself, a tab, or a combination of spaces and a tab. An example of some Arabic plural data points coded in this manner is given in Figure 1. Note that I have changed the outcomes from being reminiscent of what they refer to (*fem* and *CaCaaCiiC*) to simply A and B in order to improve the appearance of the data file.

```
A SSrHFRG tSryH
A SrtbRSL trtyb
A DHqqGDD tHqyq
A ScdlGSR tcdyl
B SqldDRS tqlyd
A SshlFGR tshyl
B SdbrSLR tdbyr
A SqdrDSR tqdyr
A DhddGSS thdyd
```

**Figure 1.** Arabic plural data coded with spaces only, one character outcomes

```
fem SSrHFRG tSryH 3345
fem SrtbRSL trtyb 1714
fem DHqqGDD tHqyq 1566
fem ScdlGSR tcdyl 1447
CaCaaCiiC SqldDRS tqlyd 1190
fem SshlFGR tshyl 1007
CaCaaCiiC SdbrSLR tdbyr 967
fem SqdrDSR tqdyr 906
fem DhddGSS thdyd 874
```

**Figure 2.** Arabic plural data coded with spaces only, multi-character outcomes



fem	SSrHFRG	tSryH	3345
fem	SrtbRSL	trtyb	1714
fem	DHqqGDD	tHqyq	1566
fem	ScdlGSR	tcdyl	1447
CaCaaCiiC	SqldDRS	tqlyd	1190
fem	SshlFGR	tshyl	1007
CaCaaCiiC	SdbrSLR	tdbyr	967
fem	SqdrDSR	tqdyr	906
fem	DhddGSS	thdyd	874

**Figure 3.** Arabic plural data coded with spaces only, multi-character outcomes, using tabs for spaces

Figure 2 shows the same data in the same format but with my original outcome specifications (and this time with a frequency added to the specifier). Although the program treats it the same, it is messy to look at and difficult to read.

Figure 3 shows the same data using tabs instead of spaces to separate the outcome, variables, and specifier, which improves the look.

**3.1.1.2 Formatting with commas and spaces.** The second option for formatting the data file involves separating the outcome, variables, and specifier with commas, and separating the variables from each other with spaces. This method of formatting gives the user more flexibility in the naming of variable options, which can be any number of characters. Again, the spaces separating the variables may also be tabs.

Figure 4 shows a few lines of the Arabic plural data in this format, with the names of the variables made to be a little bit more readable for a human (using Snd and Dbl for ‘Sound’ and ‘Doubled’ respectively, instead of simply S and D, and the like). Figure 5 shows the data again, this time using tabs to align it more nicely.

```
fem, Snd S r H Fric Rnl Gut, tSryH
fem, Snd r t b Rnl Stop Lab, trtyb
fem, Dbl H q q Gut Dq Dq, tHqyq
fem, Snd c d l Gut Stop Rnl, tcdyl
CaCaaCiiC, Snd q l d Dq Rnl Stop, tqlyd
fem, Snd s h l Fric Gut Rnl, tshyl
CaCaaCiiC, Snd d b r Stop Lab Rnl, tdbyr
fem, Snd q d r Dq Stop Rnl, tqdyr
fem, Dbl h d d Gut Stop Stop, thdyd
fem, Hol g y r Gut Vow Rnl, tgyyr
```

**Figure 4.** Arabic plural data formatted with commas and spaces

```

fem,      Snd  S  r  H  Fric  Rnl  Gut,   tSryH
fem,      Snd  r  t  b  Rnl  Stop  Lab,   trtyb
fem,      Dbl  H  q  q  Gut  Dq   Dq,   tHqyq
fem,      Snd  c  d  l  Gut  Stop  Rnl,   tcdyl
CaCaaCiiC, Snd  q  l  d  Dq   Rnl  Stop,  tqlyd
fem,      Snd  s  h  l  Fric  Gut  Rnl,   tshyl
CaCaaCiiC, Snd  d  b  r  Stop  Lab  Rnl,   tdbby
fem,      Snd  q  d  r  Dq   Stop  Rnl,   tqdyr
fem,      Dbl  h  d  d  Gut  Stop  Stop,  thdyd
fem,      Hol  g  y  r  Gut  Vow  Rnl,   tgyyr

```

**Figure 5.** Arabic plural data formatted with commas and spaces, using tabs for spaces

Note that these two ways of formatting the data file are not ‘mix and match.’ You must choose one way and stick with it for any particular run.

The choice between these two methods of formatting is made in the configuration file. The operative line of the configuration file says “format with commas” and the two possible settings are ‘yes’ and ‘no.’ Choosing ‘no’ means that you are choosing to format with spaces only, no commas, and with no spacing between single-character variables. Choosing ‘yes’ means that you are choosing to format with commas between the outcome, variables, and specifier, and to use spaces between the variables themselves. If this seems hard to remember, just remember that ‘yes’ means ‘use commas’ and ‘no’ means ‘don’t use commas.’

### 3.1.2 Using ‘=’ in the datafile

The equals sign (=) has a special meaning in an AM data (or test) file. That meaning is that the variable so marked is not applicable to this particular data item. It is typically used to avoid doubling up zeros in row after row of variables when a zero in one column implies a zero in one or more other columns. For example, if a particular variable marks the third vowel of a word, and the word only has one or two vowels, you would mark a zero for that variable. If the next variable marks the consonant after the third vowel, you could also mark a zero for that variable. However, the fact that there is not third vowel already implies that there is no consonant after that third vowel, so if you don’t want the program to ‘overemphasize’ the importance of the presence or absence of the third syllable, you could mark this variable with ‘=’.

## 3.2 The Outcome file

The Outcome file must be named *outcome*. Since it lists the outcomes, it is a common error to name it *outcomes*. There is no theoretical or other reason why the

A. *When Outcome and Specifier are different*

```
A fem  
B CaCaaCiiC
```

B. *When Outcome and Specifier are the same*

```
fem fem  
CaCaaCiiC CaCaaCiiC
```

**Figure 6.** Two possible Outcome files

singular is used. It was simply done that way, and no one bothered to change it. The Outcome file lists the possible outcomes for this dataset. It also allows you to list a specifier for the outcome in case you want to use something short (like A and B) to refer to the outcomes in the dataset. There is only one possible format for the outcome file: Outcome Specifier, with a space in between, with one outcome/specifier set on each line. If you don't want to have a separate specifier you simply list the name of the outcome twice on the line.

Figure 6 shows two different possible outcome files for the Arabic Plurals dataset, the first for use when the outcomes are listed as A and B in the data, and the second for use when the whole term is used in the data.

If an outcome that actually appears in the data is not listed in the Outcome file, results attributed to it will not appear in the results file, so it is important to make sure your Outcome file is complete. Figure 7 contains two short Perl programs – one for data files with commas and the other for those without commas – that go through a data file and pull out all the outcomes listed and print them in the format of an Outcome file. I recommend using something like this if you have more than just a few outcomes. It could also be used to error check the outcome portion of your data file to see if you have any typos.

### 3.3 The Test file

The Test file must be named *test*. The Test file lists the test items or 'givens' which you want to present to the program for a prediction. The important thing to remember about the test file is that it is in the exact same format as the data file. This allows you to easily use a copy of the data file as the test file so you can test the data on itself. If you choose to list an outcome for the test item, the program will simply ignore it, read in the variables and make a prediction based on them. (See Figure 8 for examples.) In the future we are planning to write a kind of post-processor that will allow the program to check to see if its predicted answer is the same as the outcome you list. However, currently the program sim-

```
#####For data files with commas and spaces#####
#!/perl
while (<>) {
  chomp;
  ($outcome,@rest) = split /\s*,\s*/, $_;
  $o{$outcome}++;
}
foreach $ot (sort keys %o) {
  print "$ot $ot\n";
}

#####For data files with spaces only, no commas#####
#!/perl
while (<>) {
  chomp;
  ($outcome,@rest) = split /\s*/, $_;
  $o{$outcome}++;
}
foreach $ot (sort keys %o) {
  print "$ot $ot\n";
}
```

**Figure 7.** Programs for the automatic creation of the Outcome file

ply throws the outcome information on the test items away. If you choose not to list an outcome with the test item, you must still maintain the formatting. This means that if you are formatting without commas, each test item would then begin with a space before the list of variables. If you are formatting with commas, each test item should begin with a comma (or a space and a comma) before the list of variables. Note that you cannot have the data file configured one way (say with commas), and the test file configured the other way (say with just spaces). They must match.

#### 4. The AM.config file

Once you have the three files described above ready (*data*, *outcome*, and *test*), put them in a subfolder or subdirectory inside the folder or directory that contains the AM program and the file *AM.config*. Then open up *AM.config* and choose the parameters you wish to set. Figure 9 shows *AM.config* set for some typical parameters.

a. *Test File with Commas and Spaces, Outcome Specified*

```
CaCaaCiiC, S r b c R L G, trbyc
CaCaaCiiC, S d r b S R L, tdryb
fem, S l q H R D G, tlqyH
fem, S l T p R S F, tlTyp
```

b. *Test File with Commas and Tabs, Outcome Specified*

```
CaCaaCiiC, S r b c Rnl Lab Gut, trbyc
CaCaaCiiC, S d r b Stop Rnl Lab, tdryb
fem, S l q H Rnl Dq Gut, tlqyH
fem, S l T p Rnl Stop Fric, tlTyp
```

c. *Test File with Spaces Only, A/B Outcome Specified*

```
B SrbcRLG trbyc
B SdrbSRL tdryb
A SlqHRDG tlqyH
A SlTpRSF tlTyp
```

d. *Test File with Commas and Tabs, No Outcome Specified (note comma at start of each line)*

```
, S r b c Rnl Lab Gut, trbyc
, S d r b Stop Rnl Lab, tdryb
, S l q H Rnl Dq Gut, tlqyH
, S l T p Rnl Stop Fric, tlTyp
```

e. *Test File with Spaces Only, No Outcome Specified (note space at start of each line)*

```
 SrbcRLG trbyc
 SdrbSRL tdryb
 SlqHRDG tlqyH
 SlTpRSF tlTyp
```

Figure 8. Sample Test files

```
Project Name      : ArPlurals
null              : exclude
given             : exclude
probability       : 1
repeat            : 1
format with commas : yes
linear or squared : squared
output to file    : no
specify frequency : no
```

Figure 9. The AM.config file

To set parameters, leave the names to the left (of the colons) unchanged and replace the settings on the right. Each of these settings is described individually:

- a. *Project Name.* List the name of the project here. This is the name of the subdirectory containing the files *data*, *outcome*, and *test*. Further, if a results file is created, it will be created in the same subdirectory and called *amcresults*. (If you are using a Perl-only version of the AM program, it will be called *results*.)

- b. *Null.* The possible settings are 'exclude' and 'include.' This refers to how you want the equals signs treated in the given context. If you want them to be simply ignored (thus reducing the number of variables) leave the setting 'exclude.' If you want them treated as real variables, set this to 'include.' This allows you to run the program both ways without having to change the data in the data file.
- c. *Given.* Again, the two settings are 'exclude' and 'include.' The typical setting here is 'exclude.' Excluding the given context means that if the program finds the given context in the data, it ignores it and runs the prediction based on the other items. If you choose 'include,' and the context is in the data, you will always get a 100% correct prediction. This is usually not what you want, but it does allow you to look at the surrounding analogical set if there is one, which can sometimes be useful. Normally one would want to leave this parameter set to 'exclude.'
- d. *Probability.* The setting here must either be '1,' or a decimal fraction between 0 and 1. If the probability is set to 1, the program simply accepts all the data in the dataset and uses it in its calculations. If the probability is set to less than 1, the program uses a randomizer to get rid of some of the data randomly. This is normally used to model the forgetting of data randomly and thereby test the robustness of the predictions. For example, if the probability is set to '0.5' and the same given is then run through 10 different times, the program will randomly leave out about half the data each time, so each time the given is being compared to a considerably different dataset. Comparing the predictions of the 10 runs would then give us an idea of the robustness of a particular pattern.
- e. *Repeat.* The setting here must either be 1 or some whole number greater than 1. There would be no point in having a number greater than 1 if the probability is set to 1, since the results would be the same each time. However, if the probability is set to a value less than 1, then you might want to enter a number of repeats here to see how consistent the predictions are under imperfect memory.
- f. *Format with Commas.* The settings here are 'yes' and 'no.' Put 'yes' if you will be using commas between the fields and spaces between the variables, and 'no' if you will be using spaces between the fields and nothing between the variables.
- g. *Linear or Squared.* Once the program has apportioned the data to its various contexts and determined the homogeneity of the contexts, it can simply count the frequency of the items (Linear) or it can count the pointers (Squared). The results are usually similar, but using the Squared setting often gives a sharper picture since it emphasizes gang effects.
- h. *Output to File.* The settings here are 'yes' and 'no.' If you put 'yes,' the program will create a file called *amcresults* in the same subdirectory with your files *data*, *outcome*, and *test*. (The file is called *results* if you use a Perl-only version of the

AM program.) If you put 'no', the results will appear wherever Perl sends its Standard Output on your installation. In Unix, this would be to the screen. In Windows, it would be to the DOS Window (probably not a very useful option). In Macintosh, it would be into a MacPerl window created by the program.

- i. *Specify Frequency.* The settings here are either 'no' or some positive integer. If you set it to a positive integer, say 100, the program will then only read in the first 100 data points, and ignore the rest. This can be useful if you have a dataset sorted by frequency and you want to test a given with increasing amounts of data starting with the most frequent, as Derwing and Skousen (1994) did in their study of the acquisition of the English irregular verb forms.

Once you have filled out the configuration file, be sure to save it, and then you are ready to run the program.

## 5. Interpreting the results of a run

The header of the results file includes much of the information from the configuration file. This is provided so you can easily remember how the run was set up when you start to analyze the results. The order of this information is as follows: the project name; what the given context is (the current test); whether examples of the given context in the data file will be excluded or included; the number of data points; the probability that a data item will be included; the total number of items excluded; whether nulls (=) in the given context will be excluded or included; whether the counting will be linear or squared; the number of active variables (given possible exclusions); and the number of active contexts.

After the header, the program then lists the items that ended up in the analogical set for that given, along with their associated outcome and the number of times they occurred (or the number of pointers to those occurrences if you chose 'squared'). This allows you to see what items are affecting the prediction made by the program. Finally, a statistical summary is given in which each outcome is listed with the number of times it is predicted, along with the relative percentages of each outcome. Figure 10 shows a typical results file. (These runs should be considered a kind of 'toy' data set since they are based on only 36 data points. They are included for illustration purposes only.) It is based on running the test words *trbyc* and *t!Typ* against a portion of the Arabic Plurals dataset. The dictionary lists the plural of *trbyc* as *taraabiic* and the plural of *t!Typ* as *t!Typaat*, so we are therefore hoping the program will predict *CaCaaCiic* for the first and *fem* for the second.

Notice that the Analogical Set lists the outcomes as they are coded in the dataset, but that the Statistical Summary also includes the Specifiers for those outcomes from the outcome file. A look at the results shows that indeed, even with

```
Project Name: ArPlurals
Given Context: S r b c R L G, trbyc
If context is in data file then exclude
Number of data points: 36
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128

Analogical Set
Total Frequency = 65
  A trtyb  8 12.31%
  B tdbyr 15 23.08%
  A tTbyq 15 23.08%
  A tpryc  4  6.15%
  B trxyS  8 12.31%
  B tcbyr 15 23.08%

Statistical Summary
  A fem          27 41.54%
  B CaCaaCiiC   38 58.46%
```

```
Project Name: ArPlurals
Given Context: S l T p R S F, tlTyp
If context is in data file then exclude
Number of data points: 36
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128

Analogical Set
Total Frequency = 45
  A trtyb 11 24.44%
  A tcdyl  9 20.00%
  A tqdyr  9 20.00%
  A thdyd  5 11.11%
  B trxyS  2  4.44%
  A tHDyr  9 20.00%

Statistical Summary
  A fem          43 95.56%
  B CaCaaCiiC    2  4.44%
```

**Figure 10.** A typical Results file



our small dataset, the program correctly predicts the plural form for *trbyc*, but with only 58% of the pointers, making it not a very strong prediction. If **selection by plurality** is chosen, then one can state that the program predicts *CaCaaCiiC* for this noun. If **random selection** is chosen, then one would predict that 58% of the time *CaCaaCiiC* will be chosen over *fem*. For the second item, a much stronger correct prediction of the *fem* plural is obtained (95%).

Figure 11 shows a results file for the given context *trbyc* when the frequency has been set to 10 (meaning that the program threw out all data after the first 10 datapoints). Note that with the reduced number of datapoints, the program now makes an incorrect prediction for the plural of this noun.

Figure 12 shows the same test item run with the setting of 'linear' instead of 'squared.' Note that as predicted this makes the prediction less sharp, but it still barely pulls through correctly at 51%.

Figure 13 shows what happens when the test item is in the data file, and you choose to include rather than exclude that context. Notice that items besides the given can occur in the analogical set because of homogeneity, but that the result is always 100% in favor of the outcome of the given in the dataset. In this case, the given *twjyh* has 68% of the pointers and thus overwhelms all the other items.

Of course, if the outcome for the given in the dataset is non-deterministic (if the given occurs more than once with different outcomes), then the percentage of the prediction will reflect the percentages in the dataset. Figure 14 shows such

```
Project Name: ArPlurals
Given Context: S r b c R L G, trbyc
If context is in data file then exclude
Number of data points: 10
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128
Analogical Set
Total Frequency = 14
  A tSryH 2 14.29%
  A trtyb 6 42.86%
  B tdbyr 6 42.86%

Statistical Summary
  A fem      8 57.14%
  B CaCaaCiiC 6 42.86%
```

Figure 11. A Results file with frequency set to 10

```

Project Name:  ArPlurals
Given Context: S r b c R L G, trbyc
If context is in data file then exclude
Number of data points: 36
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: linear
Number of active variables: 7
Number of active contexts: 128
Analogical Set
Total Frequency = 27
  A trtyb  4  14.81%
  B tdbyr  5  18.52%
  A tTbyq  5  18.52%
  A tpryc  4  14.81%
  B trxyS  4  14.81%
  B tcbyr  5  18.52%
Statistical Summary
  A fem           13  48.15%
  B CaCaaCiiC    14  51.85%

```

**Figure 12.** A Results file with counting set to linear

```

Project Name:  ArPlurals
Given Context: S w j h V D G, twjyh
Include context even if it is in the data file
Number of data points: 36
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128
Test item is in the data
Analogical Set
Total Frequency = 183
  A tHqyq      6   3.28%
  A twjyh     125  68.31%
  A tcqyd     11   6.01%
  A tfjyr1    19  10.38%
  A tCkyd     11   6.01%
  A tpkyl     11   6.01%
Statistical Summary
  A fem    183  100.00%

```

**Figure 13.** A Results file with the given included in the dataset

```
Project Name: ArPlurals
Given Context: S c l m G R L, tclym
Include context even if it is in the data file
Number of data points: 36
Probability of including any one data point: 1
Total Excluded: 0
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128

Test item is in the data
Test item is in the data
Analogical Set
Total Frequency = 352
  A tclym 176 50.00%
  B tclym 176 50.00%

Statistical Summary
  A fem          176 50.00%
  B CaCaaCiiC 176 50.00%
```

**Figure 14.** A Results file with the given included in the dataset, where the given occurs more than once with different outcomes

a result. The point of a (somewhat trivial) run like this is to show what happens when a person remembers the item in question.

Figure 15 shows a run in which the probability has been set to 0.5, and the repeat has been set to 4. One can then examine the results comparatively to see how robust the pattern is. With this small dataset it is clear that the pattern is not very robust, since the results vary widely. On the other hand, 3 out of the 4 runs did make the correct prediction. Note that the analogical sets of each run are different because the items left out vary. This changes both what items are available to be in the analogical set, and what items are available to make those items heterogeneous. You may want to consider doing runs like this with the given context ‘included’ rather than ‘excluded,’ which would model what might happen when a speaker alternatively remembers and forgets a particular form.

## 6. Compiling the results of multiple runs

Future versions are planned to allow for the compiling of results from multiple runs. However, the program does not currently do so. If you want to add such a capability yourself, you should be aware that the program “collects” the name of

```

Project Name:  ArPlurals
Given Context: S r b c R L G, trbyc
If context is in data file then exclude
Number of data points: 36
Probability of including any one data point: 0.5
Total Excluded: 18
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128
Analogical Set
Total Frequency = 56
  A  tSryH   4   7.14%
  A  trtyb  12  21.43%
  B  tdbyr  12  21.43%
  A  tTbyq  12  21.43%
  A  twjyh   4   7.14%
  B  trxyS  12  21.43%

```

Statistical Summary

```

  A  fem           32  57.14%
  B  CaCaaCiiC   24  42.86%

```

```

Project Name:  ArPlurals
Given Context: S r b c R L G, trbyc
If context is in data file then exclude
Number of data points: 36
Probability of including any one data point: 0.5
Total Excluded: 18
Nulls: exclude
Gang: squared
Number of active variables: 7
Number of active contexts: 128
Analogical Set
Total Frequency = 66
  A  tSryH   6   9.09%
  A  trtyb  12  18.18%
  B  tdbyr  12  18.18%
  A  twjyh   6   9.09%
  B  twryx   6   9.09%
  B  trxyS  12  18.18%
  B  tcbyr  12  18.18%

```

Statistical Summary

```

  A  fem           24  36.36%
  B  CaCaaCiiC   42  63.64%

```

Figure 15. A Results file with probability = 0.5 and repeat = 4

Project Name: ArPlurals  
Given Context: S r b c R L G, trbyc  
If context is in data file then exclude  
Number of data points: 36  
Probability of including any one data point: 0.5  
Total Excluded: 21  
Nulls: exclude  
Gang: squared  
Number of active variables: 7  
Number of active contexts: 128  
Analogical Set  
Total Frequency = 37

A	tSryH	2	5.41%
B	tdbyr	13	35.14%
B	tSmyM	3	8.11%
B	trxyS	6	16.22%
B	tcbyr	13	35.14%

## Statistical Summary

A	fem	2	5.41%
B	CaCaaCiiC	35	94.59%

Project Name: ArPlurals  
Given Context: S r b c R L G, trbyc  
If context is in data file then exclude  
Number of data points: 36  
Probability of including any one data point: 0.5  
Total Excluded: 13  
Nulls: exclude  
Gang: squared  
Number of active variables: 7  
Number of active contexts: 128  
Analogical Set  
Total Frequency = 51

A	tSryH	6	11.76%
A	trtyb	12	23.53%
B	tdbyr	7	13.73%
A	twjyh	6	11.76%
B	tSmyM	2	3.92%
B	twryx	6	11.76%
B	trxyS	12	23.53%

## Statistical Summary

A	fem	24	47.06%
B	CaCaaCiiC	27	52.94%

Figure 15. (continued)

the predicted outcome of the test item when it reads in the test item, so you can use this variable in adding to the code.

It is also possible to configure the specifier of the test items in such a way that it would be fairly easy to write a program that compiled and presented results automatically. The problem with doing this is that AM results have a variety of interpretations. If all you want to know is what prediction 'won' (selection by plurality), then automatic collection of results could be a good idea, but if you are more interested in the details, in multiple predictions, or simply in random selection, then this method could hide as much as it reveals.

## 7. Conclusion

Running the AM program is not difficult. It does take a certain amount of care and practice to get good at creating appropriate data and test files and in interpreting the results. Readers who get stuck are invited to contact the author at <dil@byu.edu> for further information.

## References

- Skousen, Royal (1989). *Analogical modeling of language*. Dordrecht: Kluwer Academic Publishers.
- Derwing, Bruce L., & Royal Skousen (1994). Productivity and the English past tense: Testing Skousen's analogy model. In S. D. Lima, R. L. Corrigan, & G. K. Iverson (Eds.), *The reality of linguistic rules* (pp. 193–218). Amsterdam: John Benjamins.



# Implementing the Analogical Modeling algorithm

Theron Stanford

## 1. Introduction

The Analogical Modeling (AM) algorithm is quite simple, yet one would not want to carry it out by hand except for the smallest of data sets. Consequently, it has been implemented numerous times on computers, first as a Pascal program by Skousen himself and subsequently as a series of Perl programs by others in his AM research group at BYU. Recently, the group has produced an implementation combining the strengths of Perl (version 5.6) and C, substantially reducing the running times on data sets with as many as 24 variables. This paper describes the design of this latest implementation and documents the code, which can be obtained from <http://humanities.byu.edu/am/>.

## 2. Looking again at AM from first principles

Before delving into the code, we first revisit the basic principles of AM. As we do so, it will soon become apparent that this latest implementation of the AM algorithm is perhaps the one closest so far to the original spirit of the method.

### 2.1 Subcontexts and supracontexts

We begin by looking at a concrete example – namely, the artificial one Skousen introduces in *Analogical Modeling of Language*. We consider contexts with three variables, each of which can take an integer value from 0 to 3. The outcomes are marked by  $e$  and  $r$ . There are five occurrences in the data set,<sup>1</sup> listed here as ordered pairs:

(310,  $e$ ) (032,  $r$ ) (210,  $r$ ) (212,  $r$ ) (311,  $r$ )



The given context will be 312. To make a prediction, we must look at the subcontexts and the supracontexts of the given and the relationships between them.

The *subcontexts* of the given are subsets of the data set, labeled by character strings consisting of the values of the variables of the given or their negations. An item in the data set is in a particular subcontext if its variables match those of the subcontext's label. In our particular example, there are eight subcontexts, four of which are nonempty:

SUBCONTEXT	DATA ITEMS
312	–
31 $\bar{2}$	(310, e) (311, r)
3 $\bar{1}$ 2	–
$\bar{3}$ 12	(212, r)
3 $\bar{1}$ $\bar{2}$	–
$\bar{3}$ 1 $\bar{2}$	(210, r)
$\bar{3}$ $\bar{1}$ 2	(032, r)
$\bar{3}$ $\bar{1}$ $\bar{2}$	–

The *supracontexts* are unions of the subcontexts. They are labeled by character strings consisting of the values of the variables of the given or by the symbol –, signifying that the value of the variable in that position is not considered. There are eight supracontexts, formed by unions of the subcontexts:

SUPRACONTEXT	SUBCONTEXTS
3 1 2	312
3 1 –	312 31 $\bar{2}$
3 – 2	312 3 $\bar{1}$ 2
– 1 2	312 $\bar{3}$ 12
3 – –	312 31 $\bar{2}$ 3 $\bar{1}$ 2 3 $\bar{1}$ $\bar{2}$
– 1 –	312 31 $\bar{2}$ $\bar{3}$ 12 $\bar{3}$ 1 $\bar{2}$
– – 2	312 3 $\bar{1}$ 2 $\bar{3}$ 12 $\bar{3}$ $\bar{1}$ 2
– – –	312 31 $\bar{2}$ 3 $\bar{1}$ 2 $\bar{3}$ 12 31 $\bar{2}$ $\bar{3}$ 1 $\bar{2}$ $\bar{3}$ $\bar{1}$ 2 $\bar{3}$ $\bar{1}$ $\bar{2}$

Since half of the subcontexts in our example are empty, we can delete them from the previous table to get the following:

SUPRACONTEXT	NONEMPTY SUBCONTEXTS
3 1 2	
3 1 –	31 $\bar{2}$
3 – 2	
– 1 2	$\bar{3}$ 12
3 – –	31 $\bar{2}$
– 1 –	31 $\bar{2}$ $\bar{3}$ 12 $\bar{3}$ 1 $\bar{2}$
– – 2	$\bar{3}$ 12 $\bar{3}$ $\bar{1}$ 2
– – –	31 $\bar{2}$ $\bar{3}$ 12 $\bar{3}$ 1 $\bar{2}$ $\bar{3}$ $\bar{1}$ 2

Now, we would like to determine which of these supracontexts are *homogeneous* and which are *heterogeneous*. To do so, we replace the labels of the subcontexts in the previous table with ordered pairs of the form (*subcontext*, *outcome*). The *outcome* will be that outcome shared by all items in the *subcontext*, if there is only one; otherwise, if there is more than one, it will be \*. This is our new table:

SUPRACONTEXT	NONEMPTY SUBCONTEXTS
3 1 2	
3 1 –	(31 $\bar{2}$ , *)
3 – 2	
– 1 2	( $\bar{3}$ 12, <i>r</i> )
3 – –	(31 $\bar{2}$ , *)
– 1 –	(31 $\bar{2}$ , *) ( $\bar{3}$ 12, <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> )
– – 2	( $\bar{3}$ 12, <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> )
– – –	(31 $\bar{2}$ , *) ( $\bar{3}$ 12, <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> )

This table makes it very easy to find homogeneity: the supracontexts with all outcomes the same (and different from \*) exhibit *deterministic homogeneity*, while supracontexts consisting of exactly one subcontext of outcome \* exhibit *non-deterministic homogeneity*. In this example, supracontexts – 1 2 and – – 2 are deterministically homogeneous, supracontexts 3 1 – and 3 – – are non-deterministically homogeneous, and supracontexts – 1 – and – – – are heterogeneous. We mark each heterogeneous supracontext with an  $\times$  and ignore what subcontexts belong to them:

SUPRACONTEXT	NONEMPTY SUBCONTEXTS
3 1 2	
3 1 –	(31 $\bar{2}$ , *)
3 – 2	
– 1 2	( $\bar{3}$ 12, <i>r</i> )
3 – –	(31 $\bar{2}$ , *)
– 1 –	$\times$
– – 2	( $\bar{3}$ 12, <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> )
– – –	$\times$

Notice that supracontexts 3 1 – and 3 – – contain the same list of nonempty subcontexts. With more active variables, there are usually many supracontexts that share the same list of nonempty subcontexts. To save space, we combine the lists of nonempty subcontexts as follows:

SUPRACONTEXTS	LIST OF SUBCONTEXTS
3 1 – 3 – –	(31 $\bar{2}$ , *)
– 1 2	( $\bar{3}$ 12, <i>r</i> )
– – 2	( $\bar{3}$ 12, <i>r</i> ) ( $\bar{3}$ 1 $\bar{2}$ , <i>r</i> )

$3\ 1\ 2\ 3-2$     *empty*  
 $-1-$      $---$      $\times$  (*heterogeneous*)

We insert a column that counts how many supracontexts share a particular list of subcontexts:

SUPRACONTEXTS	COUNT	LIST OF SUBCONTEXTS
$3\ 1-$ $3--$	2	$(31\bar{2}, *)$
$-1\ 2$	1	$(\bar{3}12, r)$
$--2$	1	$(\bar{3}12, r)$ $(\bar{3}\bar{1}2, r)$
$3\ 1\ 2\ 3-2$	2	<i>empty</i>
$-1-$ $---$	2	$\times$ ( <i>heterogeneous</i> )

Then the final tabulation of the analogical set depends only on the last two columns, coupled with the table listing subcontexts and their data items. (An example of the actual tabulation procedure will be shown in Section 2.3.) This paper describes how these two columns can be found quickly.

## 2.2 Filling the supracontextual lattice

Although we could do so, we usually don't completely fill in the entire supracontextual lattice before we look for homogeneous contexts and start counting. As the subcontexts are added one at a time, many supracontexts can be identified early on as heterogeneous – all they need is at least two subcontexts with conflicting outcomes (for this purpose,  $*$  counts as an outcome). Once a supracontext is seen to be heterogeneous, it can be safely skipped as the algorithm continues to add subcontexts; there is no reason to keep track of subcontexts in a heterogeneous supracontext, since they will not affect the analogical set. Furthermore, once a supracontext is found to be heterogeneous, all more general supracontexts must be heterogeneous as well; these heterogeneous supracontexts (referred to as *inclusive heterogeneity*) can be ignored as subcontexts are added.

Here we give an example of how to fill the lattice, using the same example as before. We assume that the nonempty subcontexts have already been found and had their outcomes marked:

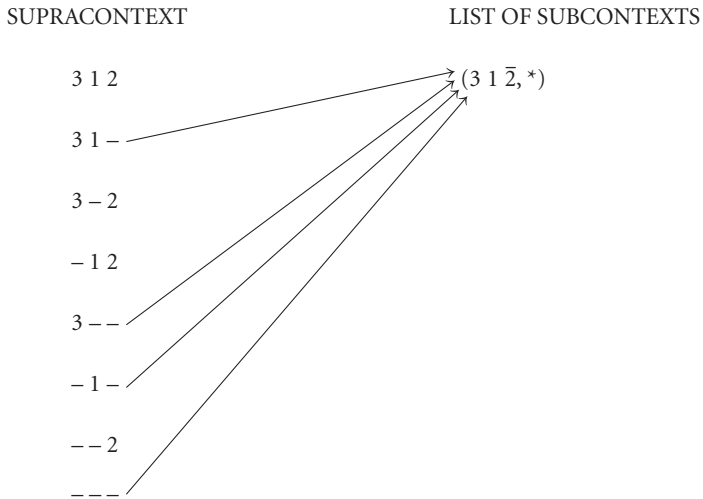
SUBCONTEXT	DATA ITEMS
$(31\bar{2}, *)$	$(310, e)$ $(311, r)$
$(\bar{3}12, r)$	$(212, r)$
$(\bar{3}\bar{1}2, r)$	$(210, r)$
$(\bar{3}\bar{1}\bar{2}, r)$	$(032, r)$

We start with an empty supracontextual lattice and accompanying table of lists of subcontexts (which is also empty), as in Figure 1.

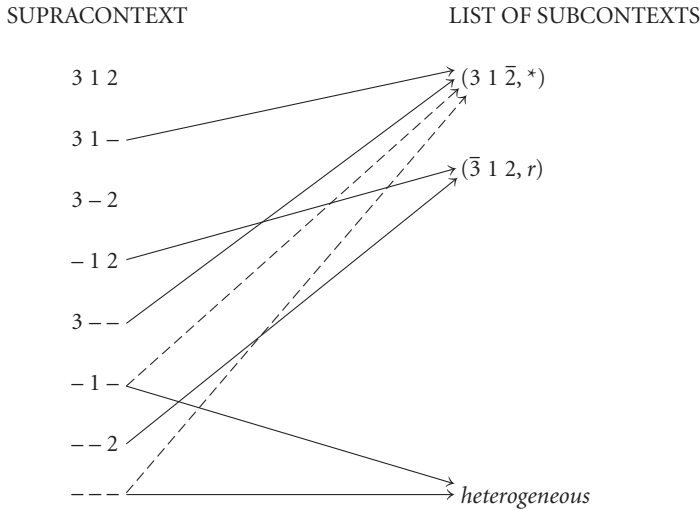
SUPRACONTEXT	LIST OF SUBCONTEXTS
3 1 2	
3 1 -	
3 - 2	
- 1 2	
3 - -	
- 1 -	
- - 2	
- - -	

**Figure 1.** The empty supracontextual lattice

It doesn't matter in which order we fill the lattice, so we'll just use the subcontexts in the order listed above. First, we add subcontext  $(31\bar{2}, *)$ ; we do this by creating a list of subcontexts which contains only the subcontext  $(31\bar{2}, *)$  and pointing to it all four supracontexts which contain it, as in Figure 2.



**Figure 2.** Subcontext  $(31\bar{2}, *)$  is added to the supracontextual lattice. Arrows are drawn to it from the supracontexts containing it.



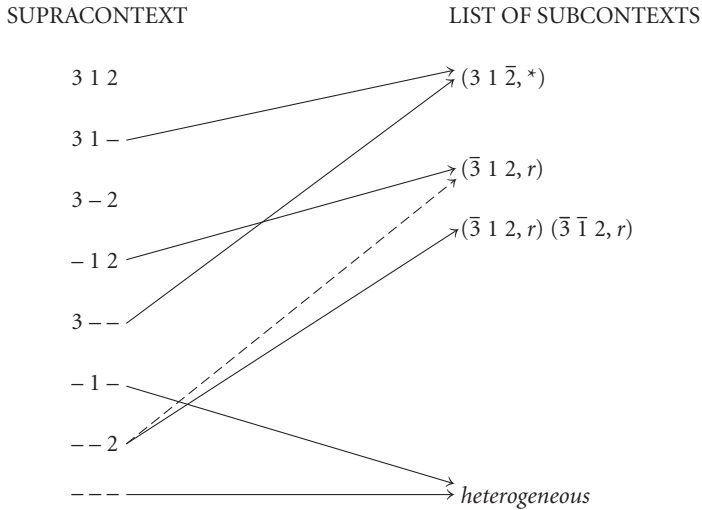
**Figure 3.** Subcontext  $(\bar{3}12, r)$  is added to the supracontextual lattice. New arrows from previously empty supracontexts  $-12$  and  $--2$  are drawn to the newly created list. Supracontexts  $-1-$  and  $---$  become heterogeneous; their previous arrows (marked with a dashed line) are moved to point to the word *heterogeneous*.

Next, we add  $(\bar{3}12, r)$ ; the result is shown in Figure 3. Two of the four supracontexts to which it is added,  $-12$  and  $--2$ , are empty; we thus create a new list containing  $(\bar{3}12, r)$  and point these two supracontexts to it. However, the other two supracontexts,  $-1-$  and  $---$ , are not empty; they point to a list with shared outcome  $*$ . Adding a subcontext with outcome  $r$  introduces heterogeneity, so the arrows from these two supracontexts are moved to point to the word *heterogeneous*. (We do not need to keep track of the actual subcontexts in heterogeneous supracontexts.)

Notice that during any stage of adding a subcontext, if a supracontext becomes heterogeneous, all more general supracontexts perforce will become heterogeneous as well and be marked so during the same stage (if they have not already been).

The next subcontext to be added is  $(\bar{3}1\bar{2}, r)$ . It should be added to supracontext  $-1-$  as well as to all more general supracontexts. However,  $-1-$  was marked as heterogeneous in the previous step; thus, there is no work to be done, and  $(\bar{3}1\bar{2}, r)$  is simply ignored.

The last subcontext to be added is  $(\bar{3}\bar{1}2, r)$ . Supracontext  $--2$  points to a nonempty list of subcontexts, but since all outcomes in this list are  $r$ , heterogeneity is not introduced; we merely duplicate the list  $--2$  points to, append  $(\bar{3}\bar{1}2, r)$ , and move the arrow. Supracontext  $---$  is already heterogeneous, so we can safely ignore it. The result is Figure 4.



**Figure 4.** Subcontext  $(\bar{3}\bar{1}2, r)$  is added to the supracontextual lattice. The original arrow from  $--2$  (marked with a dashed line) is moved to point to a new list comprised of the old list and  $(\bar{3}\bar{1}2, r)$ , since homogeneity is preserved. Supracontext  $---$  is ignored, since it is already heterogeneous.

Now that all subcontexts have been added, we can condense the information in Figure 4 to the following:

SUPRACONTEXTS	COUNT	LIST OF SUBCONTEXTS
3 1 -	3 --	2 $(3\bar{1}\bar{2}, *)$
- 1 2	1	$(\bar{3}\bar{1}2, r)$
-- 2	1	$(\bar{3}\bar{1}2, r) (\bar{3}\bar{1}\bar{2}, r)$
3 1 2	3 - 2	<i>empty</i>
- 1 -	---	2 $\times$ ( <i>heterogeneous</i> )

This is the same table as at the end of the previous section.

### 2.3 Computing the analogical set

Computing the analogical set is straightforward. We start with the last two columns of the previous table:

COUNT	LIST OF SUBCONTEXTS
2	$(3\bar{1}\bar{2}, *)$
1	$(\bar{3}\bar{1}2, r)$
1	$(\bar{3}\bar{1}2, r) (\bar{3}\bar{1}\bar{2}, r)$
2	<i>empty</i>
2	$\times$ ( <i>heterogeneous</i> )

Next, we delete any empty or heterogeneous supracontexts and replace each subcontext with its list of data items:

COUNT	DATA ITEMS
2	(310, <i>e</i> ) (311, <i>r</i> )
1	(212, <i>r</i> )
1	(212, <i>r</i> ) (032, <i>r</i> )

To count the number of pointers to any given data item, we first find which subcontext lists it occurs in and consider them one at a time. In the quadratic case, the number of pointers within a subcontext list is the number of data items in the list; in the linear case, it is 1. This number must be multiplied by the number of supracontexts which share this subcontext list; summing over all subcontext lists gives the total number of pointers.

For our example, if we assume the quadratic case, the analogical set is thus:

OCCURRENCE	NUMBER OF POINTERS
(310, <i>e</i> )	4 ( $2 \times 2$ , from row 1)
(311, <i>r</i> )	4 ( $2 \times 2$ , from row 1)
(212, <i>r</i> )	3 ( $1 \times 1 + 1 \times 2$ , from rows 2 and 3)
(032, <i>r</i> )	2 ( $1 \times 2$ , from row 3)

The linear case is handled similarly.

### 3. The program files and program flow

There are two ways for Perl and C programs to act in concert. One is to *extend* Perl by writing modules which in turn call functions in precompiled C libraries; the other is to *embed* Perl by linking C object code with the Perl interpreter – itself a precompiled C library – and making calls to it from within the C code when necessary to execute Perl code. The implementation of the AM algorithm described here uses the second method, embedding; it is the simpler of the two ways to implement and ports easily to any operating system with a C compiler and Perl. To invoke the algorithm, we simply run the C executable.

The main code for the C executable is in `amc.c` (“analogical modeling in C”), with data structures declared in `amc.h`. This code first calls subroutines in the Perl code `amc.pl` which read and parse the data, outcome, and test files and convert the data into a form which the C code can use. Next, the C code locates this data in memory and processes it as outlined in Section 2.2. Last, the C code calls subroutines in the Perl code which compute the analogical set as outlined in Section 2.3 and write out the results.

The main task of the C code is to go through the nonempty subcontexts, decide which supracontexts they belong in, and adjust the subcontext lists accordingly. The code for determining which supracontexts a subcontext belongs to is in `progeny.c`. It defines an iterator class using two functions: `ancestor()` to initialize the iterator with a subcontext, and `descendant()` to return the supracontexts one at a time. When `descendant()` returns 0, there are no more supracontexts to which the subcontext belongs.

## 4. The data structures

### 4.1 Indexing the subcontexts, supracontexts, and outcomes

A given context with  $n$  active variables has  $2^n$  subcontexts and  $2^n$  supracontexts. In the example above, the subcontexts were labeled by character strings consisting of the values of the individual variables of the given context or their logical opposites (e.g.,  $3\bar{1}2$ ); the supracontexts were labeled by character strings consisting of these same values or the symbol  $-$  (e.g.,  $3-2$ ).

Thus, once we take a context as given, a subcontext or supracontext can also be labeled by a sequence of  $n$  symbols, each one signifying either a “match” with the corresponding variable of the given or a “no match”. In other words, the subcontexts and supracontexts can be indexed by  $n$ -digit binary numbers in the range 0 to  $2^n - 1$ , using “0” to signify “match” and “1” to signify “no match”. The subcontexts and supracontexts of the example above would be indexed as follows:

INDEX	SUBCONTEXT	SUPRACONTEXT
$000_2$	$312$	$312$
$001_2$	$3\bar{1}\bar{2}$	$31-$
$010_2$	$3\bar{1}2$	$3-2$
$011_2$	$3\bar{1}\bar{2}$	$3--$
$100_2$	$\bar{3}12$	$-12$
$101_2$	$\bar{3}\bar{1}\bar{2}$	$-1-$
$110_2$	$\bar{3}\bar{1}2$	$--2$
$111_2$	$\bar{3}\bar{1}\bar{2}$	$---$

Now suppose that, say, subcontext  $3\bar{1}2$  were to be added to the supracontextual lattice. It would be added to supracontexts  $3-2$ ,  $3--$ ,  $--2$ , and  $---$ . Using binary notation, subcontext  $010_2$  would be added to supracontexts  $010_2$ ,  $011_2$ ,  $110_2$ , and  $111_2$ . It is easy to see a pattern: the indices of the supracontexts can be derived from the index of the subcontext by considering all combinations of changing the 0s to 1s.



This is precisely what the subroutines in `progeny.c` do. Subroutine `ancestor()` is passed the binary value of the subcontext to be added, while each call to `descendant()` returns the binary value of a supracontext to be added to. Subroutine `ancestor()` is also passed the number of active variables; otherwise, upon receiving the value 2, it could not determine if this represented subcontext  $010_2$  as above or perhaps  $10_2$  or  $0010_2$  or something else. Subroutine `descendant()` never returns the binary value corresponding to the subcontext; it does return 0 when there are no more supracontexts.

So, to add subcontext  $3\bar{1}2$  to the supracontextual lattice, `amc.c` would first add it to supracontext  $3-2$  by adding subcontext  $010_2$  to supracontext  $010_2$ . Then it would call `ancestor(2, 3)`, since  $3\bar{1}2$  has binary value  $010_2 = 2$  and there are three active variables. Four calls to `descendant()` would return the values 3, 7, 6, and finally 0. After each call returning a nonzero value, `amc.c` would add the subcontext to the supracontext corresponding to the returned value; in this case, the nonzero values are binary  $011_2$ ,  $111_2$ , and  $110_2$ , corresponding to supracontexts  $3--$ ,  $---$ , and  $--2$ , as expected.

The details of `ancestor()` and `descendant()` are left for the appendix to this paper; suffice it to say here that these functions work very quickly.

In comparison with the foregoing, the indexing of the outcomes is quite simple; it just starts with 1 and goes to the number of outcomes. These indices are used to mark not only individual data items, but also a subcontext if all its data items share the same outcome; otherwise, the subcontext is marked with the number 0, which corresponds to our use of  $*$  in Section 2.2.

## 4.2 Data structures in `amc.c`

Header file `amc.h` contains the type definitions necessary to implement the data structures used in `amc.c`. Type `AM_OUTCOME` must be large enough to hold the total number of outcomes, and type `AM_CONTEXT` must be large enough to hold the total number of subcontexts (or supracontexts). In the standard setup, `AM_OUTCOME` is type `unsigned char` and `AM_CONTEXT` is type `unsigned long`, but these can be modified in the interest of memory usage. For instance, if there are never more than 16 active variables, `AM_CONTEXT` could be redefined as type `unsigned short`; conversely, if the number of possible outcomes is more than 255, `AM_OUTCOME` would need to be redefined as something larger than `unsigned char`. If the sizes of these data types are changed, corresponding changes must be made in the Perl code `amc.pl`, or else data passed between the C code and the Perl code will be garbled (see Section 4.4 for details).

The remaining type, `AM_SUPRA`, is used to hold a list of subcontexts. It is defined as follows:

```
typedef struct AM_supra {
    unsigned short index;
    AM_OUTCOME outcome;
    AM_CONTEXT *data;
    AM_CONTEXT count;
    struct AM_supra *next;
} AM_SUPRA;
```

The components of AM\_SUPRA are as follows:

- `index` is a number which represents the order in which the subcontext lists are created: the higher the value of `index`, the later this list was created. The next available index number is kept in the variable `nextindex`. The reason for keeping track of this will be seen later.
- `outcome` is the index of the outcome shared by all subcontexts in the list, if there is one; otherwise, it is 0. (Note that the value of `outcome` can be 0 if all subcontexts in the list have also been assigned the value 0, which happens if the data items in each individual subcontext do not share the same outcome.)
- `data[]` is an array listing the subcontexts. `data[0]` is the number of subcontexts, and `data[1], …, data[data[0]]` contain the indices of the subcontexts.
- `count`'s value is the number of supracontexts sharing this list of subcontexts, or the number of arrows pointing to it from the supracontexts. We didn't keep track of this in our examples; rather, we waited until the end to count. However, keeping a running count is much more efficient when the algorithm is implemented as a computer program.
- `next` is a pointer which makes a circular linked list out of the lists of subcontexts. Using a circular linked list allows optimizations that we will see later.

Recall the table of subcontext lists at the end of Section 2.2:

SUPRACONTEXTS	COUNT	LIST OF SUBCONTEXTS
31– 3--	2	$(31\bar{2}, *)$
–12	1	$(\bar{3}12, r)$
--2	1	$(\bar{3}12, r) (\bar{3}\bar{1}2, r)$
312 3–2	2	<i>empty</i>
–1– ---	2	$\times$ ( <i>heterogeneous</i> )

The third row of this table could be represented as a variable of type AM\_SUPRA as follows:

- `index` equals 3, because the third row contains the third list of subcontexts.
- `outcome` equals 2, since all subcontexts share outcome  $r$ , the second outcome.

- `data[0]` equals 2, since there are two subcontexts in the list. `data[1]` and `data[2]` have the values 4 and 6, since these are just binary  $100_2$  and  $110_2$ , corresponding to subcontexts  $\bar{3}12$  and  $\bar{3}\bar{1}2$ .
- `count` equals 1, since only one supracontext ( $--2$ ) shares this list of subcontexts.

We say “could be represented” because the actual values depend on the order in which the subcontexts are added to the supracontextual lattice. Which list of subcontexts `next` points to is ignored for now.

The circular linked list of subcontexts always contains a structure with empty list and index 0. This structure is pointed to by the variable `supralist`. It is used to mark starting and ending points when the circular linked list is traversed. It is also used for other special purposes to be seen later.

The supracontext lattice links up with the subcontext lists via pointers. More precisely, each supracontext is assigned a pointer to type `AM_SUPRA`, which points to the appropriate list of subcontexts.

The list of pointers is kept in the array `lattice[]`. For each test item, `amc.c` allocates (and subsequently deallocates) enough memory for `lattice[]` to contain one pointer per supracontext. The array `lattice[]` is indexed as explained in Section 4.1. For instance, consider supracontext  $--2$ , with corresponding binary value  $110_2 = 6$ . Then `lattice[6]` will point to the list of subcontexts indexed by the number 3, as indicated in our table above. In other words, `lattice[6]->index` is 3, `lattice[6]->outcome` is 2, `lattice[6]->data[0]` is 2, and `lattice[6]->data[1]` and `lattice[6]->data[2]` are respectively 4 and 6.

This assumes, of course, that the supracontextual lattice has been completely filled. When `lattice[]` is first allocated memory, each element points to the same empty list of subcontexts that `supralist` does. As the lattice fills (how this happens is described in Section 6), these pointers change value until there are no more subcontexts to be added, at which point computation of the analogical set begins.

Notice that the actual structure of the lattice is never stored in memory – that is to say, there are no pointers between elements of the lattice indicating parenthood or childhood. The only way to tell that supracontexts indexed by  $010_2$  and  $110_2$  are in a parent-child relationship is by comparing the binary digits. However, there is no need to store such a relationship because of the iterator class defined in `progeny.c`. The functions `ancestor()` and `descendant()` take care of determining what parts of the supracontextual lattice need attention during any stage of adding a subcontext.

Though it did not happen in the example above, it is often the case that a list of subcontexts created at an earlier stage may find at a later stage that it has lost

all supracontexts pointing to it; that is, `count` is 0. To conserve memory, `amc.c` calls `cleansupra()` after each stage of adding a subcontext to all possible supracontexts. `cleansupra()` removes from the circular linked list all subcontext lists whose `count` has decremented to 0 (except, of course, the empty one pointed to by `supralist`).

As we saw in Section 2.2, once a supracontext has been deemed heterogeneous, there is no reason to attempt to add any more subcontexts to it. In `amc.c`, the variable `HETERO` of type `AM_SUPRA*` is allocated to point to an unused memory location; when a supracontext is marked as heterogeneous, its associated pointer in `lattice[]` is set to the value `HETERO`.

### 4.3 Data structures in `amc.pl`

Two types of data structures occur in `amc.pl`. One type contains data which are used throughout the run and are initialized during the subroutine `setup()`; these are described in Section 4.3.1. The other type contains data which change with each test item and are computed in either `beginTestItem()` or `count()`; these are described in Section 4.3.2.

#### 4.3.1 Structures maintained throughout the run

Data used throughout the run is obtained from the data, test, and outcome files. The locations of these files and how they are to be parsed are determined by the configuration file `AM.config`. The formats of `AM.config` and the data, test, and outcome files are described elsewhere and not repeated here.

Each outcome consists of an abbreviated form, used to mark items in the data file, and a long form, listed in the “Statistical Summary” printed at the end of each test run. These two forms are in the arrays `@ocl` and `@outcomelist`, respectively. These two arrays begin with elements with values set to `undef`, so that each outcome has a positive integer for an index. There is also a hash `%outcometonom` which is used to convert an outcome in abbreviated form to its index.

The data items are parsed into three arrays: `@outcome`, `@data`, and `@spec`. For the  $i$ th data item, `$outcome[i]` contains the index of its outcome, `$data[i]` contains a reference to an array containing the values of the individual variables of the data item, and `$spec[i]` contains its specifier.

`@testItems` is just the contents of the test file, one line per array element, to be parsed later.

#### 4.3.2 Structures which change with each test item

The hashes `%subcontext` and `%subtooutcome` contain the information needed to create the supracontextual lattice and compute the analogical set. The keys of

`%subcontext` and `%subtooutcome` are the string equivalents of the decimal values of the binary values indexing the various subcontexts. (Recall that in Perl integer values must be converted to strings before they can act as hash keys.) The values of `%subcontext` are references to arrays listing the indices of the data items belonging to the subcontexts. The values of `%subtooutcome` are the indices of the outcomes of the subcontexts.

In our example, we have five data items: (310, *e*) (032, *r*) (210, *r*) (212, *r*) (311, *r*); and two outcomes: *e*, *r*. The correspondence between the nonempty subcontexts and the hashes `%subcontext` and `%subtooutcome` is given in Table 1.

Information about the analogical set is contained in the three variables `@datacount`, `@sum`, and `$grandtotal`. After the analogical set has been completely determined, `$datacount[i]` contains the number of pointers to the *i*th data item, `$sum[i]` contains the number of pointers to data items with outcome of index *i*, and `$grandtotal` contains the total number of pointers in the analogical set.

#### 4.4 Exchanging data between `amc.c` and `amc.pl`

The division of labor is such that `amc.pl` does the parsing and hashing, `amc.c` creates the supracontextual lattice and the lists of subcontexts, and together they do the counting. This requires some data to be passed between the two programs. Some of the data are single-valued variables, while other data comprise arrays. This section describes just how this is done.

First, `amc.c` retrieves the number of test items by finding the last index of array `@testItems` in `amc.pl` and adding 1:

```
num_test_items
    = av_len(get_av("testItems", FALSE)) + 1;
```

(`av_len()` is a bit of a misnomer, since it does *not* return the length of an array value.)

For scalars in `amc.pl` with integer values, the call `SvIV(get_sv("name", FALSE))` from `amc.c` returns the value of `$name`. This is used to retrieve for each test item the number of active variables (`$activeVar`), the size of the supracontextual lattice (`$activeContexts`), and the number of nonempty subcontexts (`$numsubcontexts`, which is computed in `amc.pl` by looking at the number of keys in `%subcontext`).

The subcontexts and their outcomes are passed as very long C arrays. Each array is first packed into a Perl string, storing the array elements as binary data in contiguous memory; calling `SvPV_nolen(get_sv("name", FALSE))` from `amc.c` then returns a pointer to the first element of the array, which pointer can then be recast accordingly. So, once the hashes `%subcontext` and

**Table 1.** The correspondence between the nonempty subcontexts and the hashes %subcontext and %suboutcome

SUBCONTEXT	DATA ITEMS	key	%subcontext {key}	%suboutcome {key}
312	(310, e) (311, r)	"1"	[0, 4]	0
3̄12	(212, r)	"4"	[3]	2
3̄12	(210, r)	"5"	[2]	2
3̄12	(032, r)	"6"	[1]	2

(Recall that %suboutcome {"1"} equals 0 because the data items in this subcontext do not share the same outcome. Data items are indexed starting with 0, outcomes with 1.)

`%subtooutcome` are ready, they are packed with the following code at the end of subroutine `beginTestItem()` in `amc.pl`:

```
my(@subcontexts) = keys %subcontext;
$subcontexts = pack "L*", @subcontexts;
$suboutcomes = pack "C*",
    map { $subtooutcome{$_} } @subcontexts;
```

(`map()` is used in the last line to ensure that the subcontexts and their outcomes match up in the right order when read in `amc.c`.) These arrays are then accessed in `amc.c` via

```
subcontext = (AM_CONTEXT *)
    SvPV_nolen(get_sv("subcontexts", FALSE));
suboutcome = (AM_OUTCOME *)
    SvPV_nolen(get_sv("suboutcomes", FALSE));
```

In our example, `keys %subcontext` returns the list `(1, 4, 5, 6)`, so these last two lines of code have the same result as if the following initializations had been made:

```
AM_CONTEXT subcontext[] = { 1, 4, 5, 6 };
AM_OUTCOME suboutcome[] = { 0, 2, 2, 2 };
```

(In `amc.c`, `numsubcontexts` is added to the pointers `subcontext` and `suboutcome` after the assignments are made, because `amc.c` goes through the subcontexts in *reverse* order. Thus, in our example, the subcontexts are added by `amc.c` in this order: 6, 5, 4, 1.)

If types `AM_CONTEXT` and `AM_OUTCOME` are redefined in `amc.h`, the first arguments of the calls to `pack()` in the Perl code ("`L*`" and "`C*`") must be changed accordingly.

When it is time to compute the analogical set, `amc.c` sends the lists of subcontexts back to `amc.pl` one at a time, along with the number of supracontexts pointing to each list. In other words, for each element of type `AM_SUPRA` in the circular linked list of subcontext lists, it sends the array `data[]` and the scalar `count`. This is done by the function `countsupra()`, which is called within `amc.c` once all subcontexts have been added. For each list of subcontexts, it first creates “mortal” variables with the appropriate values and pushes them onto the Perl argument stack (`p` is a pointer of type `AM_SUPRA*` into the circular linked list):

```
XPUSHs(sv_2mortal(newSViv(p->count)));
XPUSHs(sv_2mortal(newSVpv(
    (char *) (p->data + 1),
    p->data[0] * sizeof(AM_CONTEXT))));
```

Then it calls `count()` in `amc.pl`, which begins as follows:

```
my $count = $_[0];
@list = ();
foreach (unpack "L*", $_[1]) {
    push @list, @{$subcontext{$_}};
}
```

The call to `unpack()` returns a list of subcontext indices; the `foreach` statement creates an array `@list` containing the indices of the actual data items in these subcontexts. (If `AM_CONTEXT` is redefined in `amc.h`, the first argument "L\*" of `unpack()` must be changed accordingly.)

In our example, `countsupra()` calls `count()` three times; the variables in `count()` take the successive values listed in Table 2.

## 5. Outline of the program

For each test item, three steps take place:

1. Each data item is compared with the test item and placed into the appropriate subcontext.
2. The subcontexts are placed into the supracontextual lattice one by one; while this is done, a running count is kept of how many times a given list of subcontexts occurs in the lattice.
3. The pointers within the homogeneous supracontexts are tallied to give the analogical set, which is then printed out or saved to a file.

Step 1 is handled completely by `amc.pl` in subroutine `beginTestItem()`, which is called from `amc.pl`. Step 2 is performed completely in `amc.c`, after it has received the subcontexts and their outcomes from `amc.pl`. Step 3 is done primarily in `amc.pl`, using data sent to it from `amc.c`.

The code for Step 1 is quite simple. For each data item, each variable is compared with the corresponding one in the test item, and the index `$context` of the data item's subcontext is computed. The index `i` of the data item is pushed onto `@{$subcontext{$context}}`, creating a hash of arrays, and `$suboutcome{$context}` is computed as explained in Section 4.3.2. When done, the indices of the subcontexts and their outcomes are passed to `amc.c`.

For Step 3, function `countsupra()` in `amc.c` loops over the subcontext lists, pushing the contents of each list along with its count onto the Perl argument stack and calling the Perl subroutine `count()`, as explained in Section 4.4. Note that the user can decide whether to count pointers linearly or quadratically. To optimize



**Table 2.** A chart of the arguments to the Perl subroutine `count()`. The rows give the values of these arguments for the three calls of `count()` from `countsupra()` in `amc.c`

<code>\$count</code>	<code>unpack "I*", \$_[1]</code>	SUBCONTEXTS	<code>@list</code>	DATA ITEMS
2	(1)	312	(0, 4)	(310, e), (311, r)
1	(4)	312	(3)	(212, r)
1	(4, 6)	312, 312	((3), (1))	(212, r), (032, r)

(Compare this with the results of Section 2.3.)

code, `amc.pl` creates two global variables, `@list` and `$tally`, and sets `$tally` during `setup()` with the following:

```
$linearOrSq eq 'linear' ?
  ($$tally = 0) : ($tally = \ $#list);
```

(`$linearOrSq` is set when `setup()` reads `AM.config`.) Then once `count()` creates the list `@list` of the indices of data items, they are counted by this routine:

```
foreach (@list) {
  $datacount[$_] += (1 + $$tally) * $count;
  $sum[$outcome[$_]] += (1 + $$tally) * $count;
  $grandtotal += (1 + $$tally) * $count;
}
```

For  $(1 + $$tally)$  will equal 1 if counting is to be done linearly and will equal the number of data items in the supracontext if counting is to be done quadratically.

The analogical set is printed out or saved to a file by the subroutine `endTestItem()` in `amc.pl`, which is called from `amc.c`.

All that remains is to describe Step 2: how the lattice is filled. This is the topic of the following section.

## 6. How `amc.c` fills the lattice

In Section 2.2, we gave an example of how to fill a supracontextual lattice. Whenever a new subcontext was to be added to a supracontext, the following steps took place:

1. If the supracontext was already marked as heterogeneous, it was skipped. If this heterogeneous supracontext was the *first* one this subcontext was to be added to, then by inclusive heterogeneity all other supracontexts to be added to were also heterogeneous, so this subcontext was skipped entirely.
2. The outcome of the new subcontext was compared with the outcomes of the subcontexts already in the list pointed to to see whether or not heterogeneity was introduced. (If there was no list to be compared with, then the supracontext was homogeneous trivially.)
3. If heterogeneity was introduced, the supracontext was marked as heterogeneous and nothing more was done with it.
4. If heterogeneity was not introduced, the supracontext was reset to point to a new list of subcontexts, consisting of the new subcontext appended to the list the supracontext previously pointed to.

Most of the difficulty in implementing this algorithm as a computer program is in the last step: `amc.c` must somehow keep track of all the subcontext lists, creating new ones only when necessary, and making sure that the right supracontexts point to them. It must not make duplicate copies of subcontext lists that already exist, lest memory be wasted (perhaps even to the extent of leading to program failure). Furthermore, as mentioned in Sections 4.2 and 5, the program maintains a running count of how many supracontexts point to any given list of subcontexts.

To this end, lists of subcontexts are always added into the circular linked list in a certain way: whenever a new list of subcontexts is created from an old one by appending a new subcontext, the new list is put into the circular linked list *immediately after* the one it is derived from; the old list links forward to the new one.

As mentioned in Section 4.2, the subcontext lists are each labeled with an index showing the order in which they were created. Right before a new subcontext is added to its corresponding supracontexts, the variable `baseindex` is set to the same value as `nextindex`. In this way, `amc.c` can tell which subcontext lists were added during the current stage and which were added previously just by comparing index numbers with `baseindex`.

With this all set up, adding a subcontext to a supracontext is quite simple. We look at the index of the subcontext list after the one the supracontext points to. If this index is less than `baseindex`, we know that it was created in a previous stage, so the new subcontext list has yet to be created. We create it, place it in the circular linked list immediately after the one the supracontext currently points to, set `index` to `nextindex` (which is then incremented) and `count` to 1, and reset the supracontext to point to it. If the index is not less than `baseindex`, we know that this is the new subcontext list we are looking for, so we increment its count by one and reset the supracontext to point to it – no creation is necessary.

Because this is a *circular* linked list, we don't have to worry about following a pointer that doesn't go anywhere – the “last” element will always point back to `*supralist`. Furthermore, since every supracontext starts out by pointing to the empty list of subcontexts (that is, `*supralist`), we don't have to write special code to take care of the case of creating a subcontext list with only one subcontext in it – we just append it to the empty list.

The actual process of adding a subcontext in `amc.c` essentially follows the five steps listed above, though in a manner differing in two small respects that help speed up the program. Recall that when a new subcontext is to be added to the supracontextual lattice, it is first added to the supracontext which shares its index, that is, the supracontext which contains the subcontext and is closest to the given. If this supracontext is already marked as heterogeneous, then by inclusive heterogeneity any other supracontext to which this subcontext would be added is also already marked as heterogeneous. Therefore, this *subcontext* need not be added to any supracontexts and is skipped over. Otherwise, `amc.c` attempts to add this subcontext to all appropriate supracontexts. This is done in the following code:

```

if (lattice[*subcontext] == HETERO) continue;
add(*subcontext, *subcontext, *suboutcome);
ancestor(*subcontext, activeVar);
while (d = descendant()) {
    if (lattice[d] == HETERO) continue;
    add(d, *subcontext, *suboutcome);
}

```

The function `add()` takes three arguments: the supracontext to be added to, the subcontext to be added, and the outcome of this subcontext; this last is used by `add()` to determine heterogeneity. It begins as follows:

```

void add(AM_CONTEXT supracontext,
        AM_CONTEXT subcontext,
        AM_OUTCOME outcome) {
    AM_SUPRA *p, *c;

    p = lattice[supracontext];
    if (p->count) --(p->count);

```

When `add()` attempts to add a subcontext to a supracontext (note that `add()` is called only if the supracontext is currently homogeneous), it does not first decide whether or not adding the new subcontext will introduce heterogeneity. Instead, it decides whether or not the list of subcontexts following the one pointed to by the supracontext was created earlier during this stage. This is determined by comparing its `index` with the current value of `baseindex`; if `index` is less than `baseindex`, `add()` creates and inserts the new list:

```

if (p->next->index < baseindex) {
    c = (AM_SUPRA *) malloc(sizeof(AM_SUPRA));
    c->index = nextindex++;
    c->count = 0;
    c->next = p->next;
    p->next = c;

```

If it was necessary to create a new list, heterogeneity is determined:

```

if((outcome and outcome == p->outcome)
    or !p->index) {

```

This line takes a little explanation. The first part means that the subcontext to be added has all data items sharing the same outcome (i.e., that `outcome` is nonzero) and that this common outcome matches the outcome shared by the subcontexts already in the list (i.e., that `outcome == p->outcome`). The second part means that the subcontext list is currently empty: the supracontext points to the empty list with `index` 0. These are the only two cases which preserve homogeneity. If homogeneity is preserved, `add()` copies the list, increments the count, appends the new subcontext, and points the supracontext to it:

```
c->outcome = outcome;
++(c->count);
c->data = calloc(p->data[0] + 2,
               sizeof(AM_CONTEXT));
memcpy(c->data, p->data,
       (p->data[0] + 1) * sizeof(AM_CONTEXT));
c->data[++(c->data[0])] = subcontext;
lattice[supracontext] = c;
```

However, if heterogeneity is introduced, `add()` does not yet delete this newly created list of subcontexts. Instead, it marks it as having no elements and marks the supracontext as heterogeneous:

```
} else {
    c->outcome = 0;
    c->data = calloc(1, sizeof(AM_CONTEXT));
    lattice[supracontext] = HETERO;
}
```

This sets a flag: *any* supracontext with the same list of subcontexts as that currently under consideration will become heterogeneous when the new subcontext is added. This is used to test future cases of possible heterogeneity within the same stage instead of comparing outcomes.

Now suppose that the list of subcontexts pointed to by the supracontext is followed by one created previously during this stage. If the following list is non-empty, all we have to do is repoint the supracontext and update the count:

```
} else {
    if (p->next->data[0]) {
        ++((lattice[supracontext] = p->next)->count);
    }
}
```

If the following list is empty, that is the flag set earlier indicating that adding the subcontext to the list currently pointed to by the supracontext will introduce heterogeneity, so the supracontext is marked heterogeneous:

```
} else {
    lattice[supracontext] = HETERO;
}
}
```

After the subcontext has been added to all possible supracontexts, `cleansupra()` is called to remove empty subcontext lists.

## Appendix

### A. The inner workings of `ancestor()` and `descendant()`

#### A.1 The theory

Given a binary number, say 1001011, an algorithm is needed to produce the following:

1001111 1011011 1011111 1101011 1101111 1111011 1111111

There is an obvious one-to-one correspondence between these binary numbers and those in the range  $001_2-111_2$  (underscores indicate affected bits):

001 → 1001111  
 010 → 1011011  
 011 → 1011111  
 100 → 1101011  
 101 → 1101111  
 110 → 1111011  
 111 → 1111111

An algorithm to produce this list would consist of the following steps:

1. Starting with a binary number of  $n$  digits, create a list of binary numbers which show where the 0s occur. Using the example 1001011 above, one possible list would be 0000100, 0010000, 0100000.
2. Create a counter going from 1 to  $2^z - 1$  in binary, where  $z$  is the number of 0s in the original binary number.
3. For each value of the counter, compute and return the new binary number. For instance,  $101_2$  would create the number  $1001011_2 + 1(0100000_2) + 0(0010000_2) + 1(0000100_2) = 1101111_2$ . (Instead of adding, bitwise OR operations could be used.)

Step 3 results in a total of  $O(z2^z)$  1-bit shifts to the right and bitwise AND operations (to ascertain where the 0s and 1s are in the counter) and another  $O(z2^{z-1})$  bitwise OR operations (to evaluate the new number). Thus, this algorithm is  $O(z2^z)$  in running time.

To improve upon this, recall that it doesn't matter in what order these binary numbers are returned. The above list could be reordered as follows:

001 → 1001111  
 011 → 1011111  
 010 → 1011011  
 110 → 1111011  
 111 → 1111111

101 → 101111  
100 → 101011

The thing to note here is that each row differs from the previous by exactly one bit. In other words, only one bitwise operation is needed at each step, assuming the result of the previous step is stored. (In practice, two are actually used: one to determine the value of the bit, and one to flip it.) However, the entries in the first column are no longer easily computed by simply adding 1 to the previous value.

To see the pattern of this first column, the previous table is repeated with an extra column in front. The values of this column represent which bit has been flipped, 0 representing the least significant:

0 001 → 100111  
1 011 → 101111  
0 010 → 101101  
2 110 → 111101  
0 111 → 111111  
1 101 → 110111  
0 100 → 110101

For larger values of  $z$ , the sequence is 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 4, ... The pattern of this last sequence becomes obvious when another column containing the positive binary integers in ascending order is prepended:

001 0 001 → 100111  
010 1 011 → 101111  
011 0 010 → 101101  
100 2 110 → 111101  
101 0 111 → 111111  
110 1 101 → 110111  
111 0 100 → 110101

The value in the second column is precisely the number of zeros the binary number in the first column ends with.

Thus, Step 3 above can be replaced by

3. Find the least significant 1 in the counter. Flip the corresponding bit in the previously returned number and return the new number.

Finding the least significant 1 in all the counters takes a total of  $O(2^z)$  1-bit shifts and  $O(2^{z+1})$  bitwise ANDs, while computing the new number requires a total of  $O(2^{z-1})$  additional bitwise operations. Thus, this algorithm is  $O(2^{z+1})$  in running time, which compared to the earlier  $O(z2^z)$  is a great improvement.

## A.2 The implementation

The algorithm explained above is handled by two functions. `ancestor()` creates the list of Step 1 and the counter of Step 2, which counter is decremented and used to create the return value as explained in Step 3 through repeated calls to `descendant()`. (It can be shown that an increasing counter and a decreasing one give the same sequence of flip digits.)

`ancestor()` takes two arguments, the original binary number `context` (passed as an integer) and the number of active variables `numvar`. It then concurrently computes the number  $z$  of 0s in the binary number, storing it in `numgaps`, and produces the list of Step 1 and places it in the array `gaps[]`. The counter `t` is originally set to  $1 \ll \text{numgaps}$ , i.e.,  $2^z$ , and is decremented at the beginning of every call to `descendant()`, as required in Step 2. The variable `a` keeps track of the last returned result, the binary number which will have a bit flipped on the next call to `descendant()`; its value is first set to that of the original binary number.

The variables `t`, `a`, and `gaps[]` are static with file scope; thus, they are encapsulated from the rest of the program, and their values persist over each call to `descendant()`, obviating the need to pass any values and speeding up the code.

On each call to `descendant()`, `t` is decremented, the least significant place containing a 1 is computed, and the corresponding bit in `a` is flipped in these lines:

```
for(i = 0, tt = t; !(tt & 1); tt >>= 1, ++i);
flip = gaps[i];
a = (a & flip ? a & ~flip : a | flip);
return a;
```

If decrementing `t` gives a value of 0, this indicates that there are no more values to return, and `descendant()` returns 0 to signify this.

### Note

1. The word *set* is not to be taken in its mathematical sense. In AM, a set may contain repeated occurrences of the same element.





# Index

## A

access to occurrences 322–324  
agreement 329  
agreement density 329  
algorithm for computer program  
365–383, 385–409  
Alzheimer’s disease 89–94  
amnesic effects 83–87  
amplitude 326–328, 335, 337–344  
analogical modeling (AM) 2–8,  
11–25, 27–47, 51–56, 73–79,  
89–96, 113–121, 127–136,  
141–151, 159–160, 162–171,  
173–177, 183–201, 209, 220–221,  
228–246, 249–258, 266, 276,  
283–292, 295–297, 308–315,  
319–346, 349–363, 365–383  
analogical modeling algorithm  
385–409  
analogical quantum mechanics  
345–346  
analogical set 13, 20–22, 27–35, 119,  
163, 341–343, 376–383, 391–392  
analogous categories 301–315  
analogy  
four-part proportional  
225–226, 265–266, 271–276  
traditional 25, 181, 276  
*and*-operator 330–332  
Anttila’s theorem 274–283  
aphasia  
anterior 89–91  
posterior 89–94  
Arabic plurals 367–374, 376–382  
artificial grammar learning (AGL)  
55–56, 79–89

artificial neural networks (ANN)  
1–4, 11, 47, 113, 201, 231–234,  
320  
associative chunk strength 80–81  
associative network 4  
autoassociator 58–60, 231

## B

backpropagation 60–61, 113, 174,  
288  
bias 186–196, 200–201

## C

case-based reasoning 226  
categorical behavior 3, 24, 25,  
238–240, 286–287  
categorization 7, 43–45, 51–96,  
301–315  
CELEX database 113–114, 164–165,  
182, 184, 187  
chi-squared analysis 19  
children’s errors 6, 22–24, 38  
class-prediction strength (CPS)  
213–215  
classification strategies 254–257  
competitive learning 237–238  
configuration file 373–376  
connectionism 2, 5, 11, 52–56,  
58–64, 201, 226, 276–283, 288,  
320  
constituent family 183–201  
constraint conjunction 271–272  
constraint ranking 274–276  
constraints in optimality theory  
faithfulness 266–271, 292, 294

input-output (IO) 267–271  
output-output (OO) 267–271  
parochial 267–271  
structure 266–271, 291–296  
contain 335–337, 340–343  
contexts vs. subcontexts 309–315  
control-control-not (*cnm*) 330–332  
corpora 362  
correspondence theory 268–271  
cryptography 319  
cue extinction 61–62

## D

Danish nominal compounds 7,  
246–253  
data file 349–363, 368–371  
data-oriented parsing 158, 226  
dataset 12, 333, 349–363  
decision rule 77–78, 82  
decision trees 211  
declarative approaches 1–4  
decoherence 321, 327–328, 329, 340  
derived interpretation 197–199  
deterministic behavior 12–13, 16,  
162–163, 182, 320–321, 326, 387  
dialectal change 6, 24, 38  
directed acyclic graph 226–228  
disagreement 12, 19–20, 329, 340  
disjunctivity 211–215  
dissimilation 292  
distance 160–161, 210–211, 247  
distributed processing 2  
dual-route approaches 4, 22–23,  
89–94, 96, 164  
Dutch  
diminutives 216  
lexical stress 23–24, 143, 177  
noun-noun compounds 6,  
181–201

## E

eager learning 158  
editing 211

## English

artificial sentences 303–306  
base-NP chunking 217  
Georgian dialect 293–296  
grapheme-phoneme conversion  
216  
part-of-speech tagging 217  
past tense 4, 22, 43–44, 89–95  
prepositional phrase (PP)  
attachment 217, 355  
spelling 13–22  
entanglement 326  
entropy 3, 25, 190–192, 328  
episodic memory 65–70  
error analysis 168–170  
evidence 5–6, 23–24, 38, 150–151  
example-based machine translation  
(EBMT) 158–159  
exclude vs. include 375–382  
exemplar-based approaches 1–7, 11,  
52–56, 64–73, 141–151, 226, 320,  
332  
experimental materials 154–155,  
204–206, 263–264  
explicitness in analogy 2, 11, 25  
exponential explosion 5, 7, 14,  
45–47, 164, 176, 235, 319–320,  
345, 359

## F

family expressions 211–215  
feature hierarchy 192–200  
Finnish past tense 12, 24, 27–35,  
38–44, 173–174, 241–246,  
283–287  
first intersect 335, 337–343  
first outcome 335, 337–343  
formatting options, spaces vs.  
commas 369–371, 375  
frequency 37–39, 241, 284, 293–296,  
362, 376, 378  
fuzziness 3, 11, 24, 25, 38, 56–57,  
160

## G

- gain ratio 172–173
- gang effect 25, 33–34, 229
- gender, grammatical 110, 143–145
- generalized context model (GCM) 72–73, 113
- German
  - grammatical gender 110
  - plural 5, 23, 109–121, 164–175, 216
  - umlauting 5, 109–112
- given context 12, 25, 333–334, 375
- global significance 34–36
- grammaticality 79–83

## H

- heterogeneity 12, 16–21, 229–231, 326, 335, 337–344, 387–392
- hidden units 232–233
- historical change 6, 23–24, 38
- homogeneity 2–3, 5, 16–21, 25, 27–34, 163, 220, 228–231, 233–235, 284, 288–289, 326, 344
- Huntington's disease 89–94

## I

- idiosyncratic behavior 238–240
- include 335–337, 340–343
- inclusive heterogeneity 17–18, 230–231, 388
- information gain (IG) 3, 6, 25, 35, 161, 172–176, 183, 210, 217
- information measure
  - logarithmic 20, 328
  - quadratic 20, 321–322, 329
- information theory 320
- infrequent words 38–39, 244–246
- inhibition
  - conditional 61–62
  - latent 61
- instance families 6, 209–221
- instance-based approaches 2, 5, 11, 23–24, 52–56, 64–73, 141–151

- instance-based learner 6
- interactive activation and competition (IAC) 232–234, 236
- intersect 335, 338–343
- issues in analogical modeling 27–47

## K

- k* nearest neighbors (*k*-NN) 209–221

## L

- lattice 15, 325, 388–391, 403–406
- lazy learning 11, 158, 226
- leakage 3, 11, 24, 38, 160
- learnability 166–168
- learning times 62
- leave-one-out 168–170
- linear associator 279–281
- linear selection 20–22, 163, 231, 375, 378–379, 392
- linearity, underlying 321, 326–328
- loan words 5, 38, 124–125, 132–135, 145
- local determination 3, 25, 35, 164
- local representations 236
- local significance 5, 34–35

## M

- matching 306–308, 314
- maximizing gain 21, 76
- memory, declarative vs. procedural 84–85
- memory, imperfect 5, 20, 37, 70, 85–87, 164, 175, 322–324, 340, 361, 375, 376, 380–382
- memory, short-term vs. long-term 306
- memory-based language processing (MBLP) 157–177
- memory-based learner (MBL) 210, 306–308
- merging 211–215, 219–220
- minimizing loss 21

monomorphemic interpretation  
197–199  
morphological families 201  
multiple trace model (MINERVA)  
70–72

## N

natural statistics 20, 164, 321–324  
nearest neighbors (NN) 2–3, 5, 11,  
25, 27–34, 73, 157–158, 161,  
233–234  
negation 330  
neurological impairments 89–95  
non-declarative approaches 320  
non-deterministic behavior 12–13,  
16, 162–163, 182, 201, 248–249,  
320–321, 326, 387  
non-distributed processing 4  
non-neighbors 163, 171–173  
non-rule approaches 1–4, 11  
nonce words 5, 23, 24, 38  
norming 326, 329, 340  
novel compounds 183–184,  
186–199

## O

observation 321, 327–328, 329  
occurrences 321, 326, 340, 344–346  
operating systems 365–367  
operators, reversible 329–332  
optimality theory (OT) 7, 247,  
265–297  
outcome file 371–372  
outcome 5, 12, 42–45, 368, 387  
overlap metric 160–161, 210–211

## P

parallel distributed processing (PDP)  
2, 11  
parallel processing 176, 319  
parameter specification 42–43  
Parkinson's disease 89–94  
Pascal program 365

pattern associator 58–60  
perceptual learning 61  
performance errors 24  
Perl/C computer program  
ancestors and descendants  
407–409  
data structures 393–402  
filling the lattice 403–406  
indexing 393–394  
outline 401, 403  
plurality of intersect 335, 337–343  
plurality of outcome 335, 337–343  
pointers to occurrences 5, 19–22,  
231, 289–290, 321, 327–328, 340,  
344–346, 392  
power of statistical test 20, 175  
predicting on the fly 3, 25, 46  
prime factorization 319  
probability estimation 322–323  
procedural approaches 1–4, 11  
processing times 22–23  
proportional principle 272–275  
prototypes 5, 52–57, 62–64, 74  
proximity 229  
pseudo-words 183–184, 186–199  
psycholinguistics 5

## Q

quadratic selection 20–22, 163, 231,  
392  
quantum analogical modeling  
46–47, 332–346  
quantum bit (qubit) 324  
quantum computing (QC) 7, 46–47,  
319–346  
quantum mechanics 7, 46, 320, 326

## R

random selection 13, 20–22, 75–77,  
163, 234–236, 289–291, 313–314,  
322, 378, 383  
random walk model 79

randomness 324–325  
 reaction time (RT) 78–79  
 regular vs. exceptional behavior 3,  
 24, 25, 37–39, 229, 238–240  
 regular vs. irregular 89–90, 160, 164  
 regularization 150–151  
 remembering, probability of  
 322–324, 375, 380–382  
 results file 376–383  
 reverse contain 337, 340–343  
 reversibility 7, 320, 326, 329–332  
 rime family 183–201  
 robustness 11, 36, 160, 164,  
 173–175, 359, 380–382  
 rule approaches 1–4, 11, 52–56,  
 125–126, 181–182, 201  
 rule-governedness 25  
 rules of usage 75–78, 234–236,  
 289–291

## S

Schrödinger's wave equation  
 321–322, 327–328, 329  
 selection by majority 234–236  
 selection by plurality 13, 20–22,  
 75–77, 163, 234–236, 313–314,  
 323–324, 378, 383  
 self-prediction 6, 37–39  
 separability, linear vs. non-linear 66  
 Shannon's information 20, 328, 340  
 Shor's algorithm 319  
 similarity 78, 160–161, 234, 284,  
 302–303  
 simple neural network model  
 (SimNet) 6–7, 236–246,  
 257–258  
 simultaneity 319, 326–328, 329  
 single-route approaches 4, 22–23,  
 96, 164  
 Spanish  
 diminutive formation 6,  
 146–147  
 grammatical gender 6, 143–145  
 lexical stress 6, 23, 148–151

specifier 368–369  
 spin-up vs. spin-down 321, 324,  
 326–328  
 squaring 321–322, 326–328, 329,  
 340, 375, 378–382  
 statistics 19–20  
 subcontexts 15–19, 162–163, 326,  
 385–394  
 suffix family 183–201  
 sum 335, 337–344  
 superposition 326–328  
 supracontexts 2–3, 12–22, 25,  
 74–75, 162–163, 228–231,  
 289–290, 320–321, 325–328, 334,  
 385–394  
 symbolic rule systems 52–56

## T

test file 361, 372–373  
 Tilburg Memory-Based Learner  
 (TiMBL) 6, 141–151, 183–201,  
 233–234, 258  
 token vs. type 360  
 tractability 7, 45–47, 319, 345  
 training period 3, 25, 46–47,  
 174–175, 320  
 transfer condition 87–89  
 transitional behavior 24  
 translation, automatic 23, 158–159  
 Turkish  
 exceptional classes 124–125,  
 131–135  
 /k/-0 alternation 5, 123–136  
 nominal system 5, 123–136  
 tutorial 4, 13–22

## U

unbiased estimators 322–324  
 uncertainty 20, 229, 320, 328–329

## V

variable selection 359–360

variable types 164, 350–353,  
355–358  
variables 5, 12, 42–43, 114–116,  
129–130, 246–248, 284, 368  
variables, significant vs.  
non-significant 5, 25, 39–40,  
46–47  
variance estimation 323  
version spaces 225–228

**W**  
weighting of variables 2, 5, 39–40,  
142, 164, 172–175  
word association 78

**Z**  
zeros, redundant vs. non-redundant  
40–42, 360, 371, 375, 376

In the series HUMAN COGNITIVE PROCESSING (HCP) the following titles have been published thus far or are scheduled for publication:

1. NING YU: *The Contemporary Theory of Metaphor. A perspective from Chinese*. 1998.
2. COOPER, David L.: *Linguistic Attractors. The cognitive dynamics of language acquisition and change*. 1999.
3. FUCHS, Catherine and Stéphane ROBERT (eds.): *Language Diversity and Cognitive Representations*. 1999.
4. PANTHER, Klaus-Uwe and Günter RADDEN (eds.): *Metonymy in Language and Thought*. 1999.
5. NUYTS, Jan: *Epistemic Modality, Language, and Conceptualization. A cognitive-pragmatic perspective*. 2001.
6. FORTESCUE, Michael: *Pattern and Process. A Whiteheadian perspective on linguistics*. 2001.
7. SCHLESINGER, Izchak, Tamar KEREN-PORTNOY and Tamar PARUSH: *The Structure of Arguments*. 2001.
8. SANDERS, Ted, Joost SCHILPEROORD and Wilbert SPOOREN (eds.): *Text Representation. Linguistic and psycholinguistic aspects*. 2001.
9. GRAUMANN, Carl Friedrich and Werner KALLMEYER (eds.): *Text Representation. Linguistic and psycholinguistic aspects*. 2002.
10. SKOUSEN, Royal, Deryle LONSDALE and Dilworth B. PARKINSON (eds.): *Analogical Modeling. An exemplar-based approach to language*. 2002.