AUTOMATED INDUCTION OF A LEXICAL SUBLANGUAGE GRAMMAR
USING A HYBRID SYSTEM OF CORPUS- AND
KNOWLEDGE-BASED TECHNIQUES

By
Geert Jan Wilms

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Department of Computer Science

Mississippi State, Mississippi
December 1995
AUTOMATED INDUCTION OF A LEXICAL SUBLANGUAGE GRAMMAR
USING A HYBRID SYSTEM OF CORPUS- AND
KNOWLEDGE-BASED TECHNIQUES

By
Geert Jan Wilms

Approved:

Lois C. Boggess
Professor of Computer Science
Director of Dissertation

Julia E. Hodges
Professor and Graduate Coordinator
of the Department of Computer Science

Donald W. Dearholt
Professor and Department Head
of the Department of Computer Science

Bradley D. Carter
Professor of Computer Science

Julian E. Boggess, III
Assistant Professor
of Computer Science

W. Glenn Steele, Jr.,
Interim Dean of the
College of Engineering

Richard D. Koshel
Dean of the Graduate School
Porting a Natural Language Processing (NLP) system to a new domain remains one of the bottlenecks in syntactic parsing, because of the amount of effort required to fix gaps in the lexicon, and to attune the existing grammar to the idiosyncracies of the new sublanguage. This dissertation shows how the process of fitting a lexicalized grammar to a domain can be automated to a great extent by using a hybrid system that combines traditional knowledge-based techniques with a corpus-based approach.

An existing broad-coverage lexicon, the product of the expertise of seasoned linguists, is exploited to induce subcategorization features for new words, based on their paradigmatic relatedness to known words. The arbiter of paradigmatic relatedness is a category space, which can be bootstrapped from co-occurrence counts of content words in a training corpus. This dissertation uses a fixed-window approach which has been augmented with phrasal boundary information, and which has been finetuned by part-of-speech disambiguation of the input tokens. A smoothing technique called Singular Value Decomposition has been used to generalize the distributional information.

Proximity in this reduced space is then used to find for all the context digests a neighborhood of words that are paradigmatically related; the subcategorization frames of a
word are a “composite” of the features associated with these “similar” words. Experiments with PUNDIT, a broad-coverage symbolic NLP system, have shown that the category space can successfully be used to induce features like transitivity and subcategorization for clauses and infinitival complements.

The data-driven process not only expands the lexicon for new words, it also fits the grammar to the new domain by adjusting the feature set of the existing verbs, adding object options where appropriate to increase coverage and removing them to purge unwanted false positives from the solution space.

The advantage of combining data-driven mining with the existing lexical knowledgebase over other bootstrapping methods is that this approach does not require the manual identification of appropriate cues for subcategorization features, or the involved construction of a pattern matcher that is sophisticated enough to ignore false triggers.
DEDICATION

To my loving wife Wallika, for her patience and constant admonitions to “go study.”
ACKNOWLEDGEMENTS

I am greatly indebted to my dissertation director, Dr. Lois Boggess, for her guidance and thoughtful comments. Although much of the final stage of writing this dissertation was a race against the clock, she worked overtime to carefully review and meticulously edit this document, one time even staying on campus till 9 PM on a Friday night to babysit a reluctant printjob.

When I was applying to several schools to enter their Ph.D. program, Dr. Boggess wrote me a letter, introducing herself and the AI program at MSU, and this personal touch was a deciding factor in me choosing to attend Mississippi State. I will also forever remember several trips I took to conferences in the Boggesses van, and the lively discussions we had while on the road.

I also wish to express my sincere appreciation to the rest of my committee — Drs. Gene Boggess, Brad Carter, Don Dearholt, and Julia Hodges — for their suggestions and critiques.

Special thanks goes to people who helped me with computing resources; system administrators Tom Ritter and Gerhard Lehnerer, who helped me with network support questions; Dr. Ray Reighart of Chemical Abstracts Service, who graciously shared his powerful workstation for some heavy duty computing that would have overwhelmed walt, the departmental server; Don Lewis, of Jackson State Community College, for granting dedicated internet access to a faculty member of a “rival school.” A special note of gratitude is also extended to Union University, especially John David Barham, for providing me with the sole Unix workstation on campus, for purchasing Quintus Prolog, and for setting
up a dedicated telephone connection with JSCC. On the software side, I am indebted to Drs. Debbie Dahl and Marcia Linebarger for granting a license to use PUNDIT and for taking the time to answer my detailed questions; to Dr. Eric Brill for sharing the code to his bracketing program; and to Dr. Mike Berry for allowing access to his SVD implementation. Thanks are also due to my chair Dwayne Jennings, dean Dr. Jim Baggett, and vice-president Dr. Howard Newell, for giving me a reduced teaching load during several semesters and for making the completion of my degree a high priority.

This work would not have been possible without grants from the National Science Foundation, and Professional Development funds from Union University.

Finally, I am very grateful to my wife Wallika, for her understanding, patience, and prodding during the long gestation of this dissertation. Now we can finally take that long-promised trip to Gatlinburg.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td><strong>I. INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td>Theory-Driven vs. Empiricist Approaches</td>
<td>1</td>
</tr>
<tr>
<td>Robustness: Absolute and Relative Ungrammaticity</td>
<td>2</td>
</tr>
<tr>
<td>Porting</td>
<td>3</td>
</tr>
<tr>
<td>Objectives</td>
<td>4</td>
</tr>
<tr>
<td>System Overview</td>
<td>5</td>
</tr>
<tr>
<td>Outline</td>
<td>6</td>
</tr>
<tr>
<td><strong>II. LITERATURE REVIEW</strong></td>
<td>7</td>
</tr>
<tr>
<td>Sublanguages</td>
<td>8</td>
</tr>
<tr>
<td>Machine-Readable Dictionaries</td>
<td>10</td>
</tr>
<tr>
<td>Massive Data-Bashing</td>
<td>12</td>
</tr>
<tr>
<td>Part-Of-Speech Tagging</td>
<td>14</td>
</tr>
<tr>
<td>“Attuning” a Grammar</td>
<td>16</td>
</tr>
<tr>
<td>Probabilistic and Best-First Grammars</td>
<td>19</td>
</tr>
<tr>
<td>Preference through Rule Ordering</td>
<td>20</td>
</tr>
<tr>
<td>Probabilistic Context Free Grammars</td>
<td>20</td>
</tr>
<tr>
<td>Context Dependent Grammars</td>
<td>22</td>
</tr>
<tr>
<td>Inducing a Grammar</td>
<td>23</td>
</tr>
<tr>
<td>The Lexicon</td>
<td>27</td>
</tr>
<tr>
<td>Part-Of-Speech Information</td>
<td>27</td>
</tr>
<tr>
<td>Syntactic Features</td>
<td>28</td>
</tr>
<tr>
<td>(Semantic) Selectional Restrictions</td>
<td>29</td>
</tr>
<tr>
<td>The Lexicon and the Grammar Proper</td>
<td>30</td>
</tr>
<tr>
<td>Manual Approach: COMLEX</td>
<td>32</td>
</tr>
<tr>
<td>Automated Approach: Data-Driven</td>
<td>33</td>
</tr>
<tr>
<td>Fill in the Blanks with a Probabilistic Parser</td>
<td>34</td>
</tr>
<tr>
<td>Inferencing Lexical Syntax: Child’s Play</td>
<td>34</td>
</tr>
<tr>
<td>“Lerner” Grown Up: Partial Parsing</td>
<td>35</td>
</tr>
</tbody>
</table>
## Chapter Page

| The Company You Keep: Prepositional Phrase | 36 |
| Attachments by Lexical Association | 36 |
| SOLOMON: Syntactic or Semantic? | 38 |

### III. INDUCING A CATEGORY SPACE | 39

- The Physical Chemistry Corpus | 40
  - Tagging the Corpus | 42
  - Bracketing the Corpus | 45
- Bootstrapping the Category Space | 56
  - Syntactically-based Schemes | 56
  - Window-based Schemes | 58
- Singular Value Decomposition | 62
  - SVD for Improved Document Representations | 63
  - SVD for Compact Representations | 64
  - Formal Description of SVD | 66
  - SVD in the Physical Chemistry Domain | 67
- Cosine Similarity | 70
- Conclusion | 73

### IV. SUBCATEGORIZATION FRAMES | 74

- PUNDIT’s Architecture | 77
  - A Logic Restriction Grammar Implementation of Sager’s LSP | 77
  - Trial 1: Core Lexicon Only | 81
  - Trial 2: Lexicon Augmented with Stochastic POS Information | 86
- Backtracking to Find the Next Parse Candidate | 91
- PUNDIT’s Lexicon and Subcategorization Features | 94
  - Format of the lexicon | 94
  - Inducing Object Options | 99
  - Trial 3: Induction of Basic Features | 102
    - Attuning the Existing Lexicon | 103
    - Inducing new Features: an Iterative Process | 117
    - The Test Corpus Revisited | 120
  - Trial 4: Recognizing Finer Distinctions | 124
- Conclusion | 125

### V. CONCLUSIONS AND FUTURE RESEARCH | 127

- Future Research | 127
- Data-Driven Attuning | 128
- Induction Without Pattern Matching | 131
- Knowledge-Based Backbone | 132

## Appendix

| A. LSP FEATURES | 134 |
| B. LSP TERM GLOSSARY | 140 |
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.</td>
<td>Bracketing Accuracy</td>
<td>51</td>
</tr>
<tr>
<td>3.2.</td>
<td>Size of Context Digests</td>
<td>59</td>
</tr>
<tr>
<td>3.3.</td>
<td>Partial Context Digests</td>
<td>71</td>
</tr>
<tr>
<td>4.1.</td>
<td>Verbs in both Test Corpus and Core Lexicon</td>
<td>82</td>
</tr>
<tr>
<td>4.2.</td>
<td>Induced Features</td>
<td>100</td>
</tr>
<tr>
<td>4.3.</td>
<td>Near-Perfect Matches</td>
<td>107</td>
</tr>
<tr>
<td>4.4.</td>
<td>Contingency Table for Recall/Precision</td>
<td>113</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Word Usage Statistics</td>
</tr>
<tr>
<td>3.2</td>
<td>Transformation-based Error-driven Learning (Brill 1993b)</td>
</tr>
<tr>
<td>3.3</td>
<td>Sample Parse Tree</td>
</tr>
<tr>
<td>3.4</td>
<td>Example of Applying Bracketing Rule “b1_r nns 25703”</td>
</tr>
<tr>
<td>3.5</td>
<td>Cross-Bracketing Measure</td>
</tr>
<tr>
<td>3.6</td>
<td>The Distribution of Sentence Lengths in the Physical Chemistry Corpus</td>
</tr>
<tr>
<td>3.7</td>
<td>Sample False Positive</td>
</tr>
<tr>
<td>3.8</td>
<td>k-best Analysis</td>
</tr>
<tr>
<td>3.9</td>
<td>Sample Context Digest</td>
</tr>
<tr>
<td>3.10</td>
<td>Harwell-Boeing Format for Sparse Matrices</td>
</tr>
<tr>
<td>3.11</td>
<td>SVD Decomposition of Sample Matrix for $k=2$</td>
</tr>
<tr>
<td>3.12</td>
<td>Impact of SVD on Cluster Size</td>
</tr>
<tr>
<td>3.13</td>
<td>Proximity of Sample Context Digests in 3-d Space</td>
</tr>
<tr>
<td>4.1</td>
<td>Correct Parse After Updating the Lexical Entry of “range”</td>
</tr>
<tr>
<td>4.2</td>
<td>PUNDIT Architecture (Hirschman, Palmer et al. 1989)</td>
</tr>
<tr>
<td>4.3</td>
<td>PUNDIT Heuristic for New Words</td>
</tr>
<tr>
<td>4.4</td>
<td>False Positives due to Mistagged Verbs</td>
</tr>
<tr>
<td>4.5</td>
<td>Vanilla Entries in the Lexicon</td>
</tr>
<tr>
<td>4.6</td>
<td>First Parse, with two Misattachments</td>
</tr>
<tr>
<td>4.7</td>
<td>Correct Parse: Number 258</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

Theory-Driven vs. Empiricist Approaches

Much of Natural Language Processing (hence NLP) in the 90s is characterized by a renewed interest in corpus-based research. This revival has also spawned explorations into hybrid systems that combine symbolic and empirical information. A common goal of many such hybrid systems is to use statistical information to address two of the well-known bottlenecks: gaps in the lexicon (Zernik 1991) and structural/part-of-speech ambiguity (Weischedel et al. 1993).

Three factors help explain the resurgence of statistical methods in NLP systems, after this approach had been abandoned in the machine translation projects of the late 50s:

(a) disappointing performance of state-of-the-art rule-based parsing systems on “real-world” sentences. Black et al. speak of a “crisis” after observing an average correctness of only 22 percent in an experiment where seven “of the best-known and most respected English parsing systems” were tested on 100 randomly selected sentences from the Wall Street Journal (Black, Garside, and Leech 1993, 5). Claiming that the traditional development cycle will never converge, Magerman even advocates completely abandoning these types of knowledge-based grammars in favor of a parser produced from information-theoretic classification and probabilistic training (Magerman 1994).

(b) the increasing availability of
machine-readable dictionaries: Cobuild (Sinclair 1987), OALD (Hornby 1974), LDOCE (Procter 1978), and Webster (Gove 1977), and

large on-line corpora: one million words in the Brown Corpus (Francis and Kucera 1982) and a target of five to ten billion words in a collection from the Linguistic Data Consortium (Liberman 1991), which includes the annotated Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993).

(c) formal frameworks for combining stochastic and knowledge-based information in a principled manner (Joshi 1994); see, for example (Charniak 1993) and (Black, Garside, and Leech 1993) for a detailed description of Hidden Markov Models, the forward-backward and the inside-outside algorithm.

Robustness: Absolute and Relative Ungrammaticality

Because of the tedious manual encoding of linguistic knowledge in traditional rule-based systems, their two greatest shortcomings are lack of robustness and portability. The brittleness of knowledge-based systems is especially obvious at the syntactic level: they have problems coping with extra-grammaticalities both in the absolute sense (genuine errors like misspellings, subject-verb disagreement, . . .) and in the relative sense (error-free sentences beyond the capabilities of the parser). In 1921 Sapir wrote that “all grammars leak” (p. 38), and Jensen and Heidorn testified that this is still true today: “trying to write a grammar to describe explicitly all and only the sentences of a natural language is about as practical as trying to find the Holy Grail” (1982, 2). Even wide-coverage grammars like the ANLT system (Briscoe and Carroll 1991) and the CLE Unification Grammar (Rayner 1992) yield error rates around 50 percent when tested on noun definitions from Longman's Dictionary Of Contemporary English and on a subset of the Lancaster-Oslo/Bergen corpus (hence LOB: Johansson, Leech, and Goodluck 1978) respectively. This is especially bad because in both cases the testbed was written material.
that had undergone the discipline of publication, and because for the failed sentences no parse structure was generated.

Few existing systems fail gracefully when parsing ill-formed input. There have been several attempts to provide fall-back mechanisms for ATN grammars (Weischedel and Black 1980), pattern matchers (Hayes and Mouradian 1982; Woszcyna et al. 1993), and case frame models (Carbonell and Hayes 1983), but these extensions introduce many ambiguities of their own or employ grammar-specific heuristics. Recent research by Rosé and Waibel suggests that statistical information used to constrain repair hypotheses can be successfully employed to fit parse fragments together, without altering the grammar, as the flexible parse strategies of the above-mentioned symbolic approaches do (Rosé and Waibel 1994).

Back in 1968, Harris pointed out that sublanguages have a grammar of their own, a grammar which may be quite different from the grammar of standard language. Several efficient applications have been developed by exploiting the lexical, syntactic, and semantic restrictions of a sublanguage (Sager 1986; Kittredge and Lehrberger 1982; Grishman and Kittredge 1986), but all depend heavily on manual identification, and do not scale up very well because of their dependence on idiosyncracies. Because of the close correspondence between word distribution and information-bearing properties in technical sublanguages, such sublanguages are good candidates for the automatic acquisition of linguistic knowledge through statistical corpus analysis (e.g., Hirschman 1986a).

**Porting**

Hirschman et al. report that it took ten person-weeks to (manually) port the lexical and syntactic component of their PUNDIT system to a new domain of “resource management” (Hirschman, Lang et al. 1989). PUNDIT (Prolog UNDerstanding of Integrated Text) is a broad-coverage NLP system developed by Unisys, and is a logic restriction-
grammar implementation of Sager's string grammar (Hirschman, Palmer et al. 1989; Sager 1981). However, the designers have chosen to extend the vocabulary and grammar, rather than to use the customary “language modeling” approach of developing a minimal covering grammar. The advantage of their strategy is that as the coverage is increased over time, each successive port requires less effort (past domains include Voyager, ATIS, MUC, and directory assistance), but at the clearly undesirable expense of creating a “looser” grammar which overgenerates. The authors conclude their report by expressing the need for an approach which would use a training corpus to “tune the system (in some automatic way) to 'specialize' the lexicon and . . . to tune the parser to a new domain” (Hirschman, Lang et al. 1989, 282).

Objectives

The goal of this dissertation is to explore the use of a hybrid system of corpus-based and knowledge-based techniques to automate the process of inducing a sublanguage grammar. In particular, the following two objectives are addressed:

a) to automate the acquisition into the lexicon of syntactic subcategorization frames for unknown words, an important component of porting a broad-coverage parser to a new domain. The sublanguage of physical chemistry and the rule-based PUNDIT system mentioned above are used as a testbed. As will be explained in the next chapter, several grammar formalisms store a large portion of their syntactic knowledge in the lexicon as syntactic features, which help constrain the number of analyses generated. Missing or incorrect features can prevent a parse from being generated, or prevent the correct parse from being listed as the preferred reading (i.e., one of the top candidates; in the absence of domain-specific semantic selectional restrictions, PUNDIT often generates a large number of possible parses). As Daelemans puts it, the solution to reusability lies in the reusability of the (automated) acquisition method (i.e., a behavior-based model), not in
the competence-oriented approach of current computational lexicology, which concentrates on “linguistic engineering of theory-neutral, poly-theoretic, multi-applicable lexical representations” (Daelemans 1995, 98).

b) to customize the existing lexicon to the particular sublanguage, i.e., to generate a “minimal” lexicon in order to maximize the pruning of the search space (the language modeling approach mentioned earlier).

Both the addition of new word definitions and the modification of the feature sets of lemmas in the “generic” lexicon, when automated, obviously facilitate porting a parser to a new domain. Because the proposed discovery procedures are statistical, i.e., data-driven, they also tend to make the system more robust when parsing unrestricted text in a sublanguage domain, provided a large representative corpus is available for training.

Meteer and Gish argue that symbolic and statistical techniques should not be viewed as a dichotomy, but as complementary, and that their integration “leverages the a priori knowledge that can be represented in phrase grammars with the knowledge that can be acquired using statistical methods” (Meteer and Gish 1994, 70). This dissertation describes such a hybrid system, which is bootstrapped from a mixture of empirical (corpus-driven) and theory-driven (a priori features in the domain-independent lexicon) data, which together drive a traditional rule-based parser.

**System Overview**

The following section provides a brief overview of the approach taken to accomplish the above goals, in order to motivate the discussion of related research in the next chapter.

A large training corpus is used to cluster content words (verbs, nouns, and adjectives) together by using collocation statistics; the subcategorization features of a word then are a “composite” of the features associated with “similar” words. This is an iterative
process: for a given word, initially there are only a few words it clusters with that have an a priori feature set. A tentative frame is derived, which causes a chain-reaction of other words being re-evaluated based on this new information. Eventually the process converges. Words in the original lexicon undergo this same process, possibly resulting in features being added or dropped in their lexicon entries.

The corpus has been pre-processed with a part-of-speech tagger (Brill 1992) to avoid clustering, say, a verb with a noun which happens to also have a verb sense. In addition, the corpus has been “bracketed” with a stochastic parser (Brill 1993a), so collocation counts can be kept based not only on a fixed window, but also on medium-distance dependencies. To compensate for a sparse data matrix, a smoothing procedure called Singular Value Decomposition is used (Berry et al. 1993). Finally, the cosine similarity algorithm, often used in information retrieval (Salton 1983), is applied to form neighborhoods.

Outline

The rest of this dissertation is outlined as follows: the next chapter attempts to put this research in the context of previous work on statistical and hybrid NLP systems. Some background on the lexicon and on syntactic subcategory features is also given. The approach taken in this dissertation is compared with three other systems that aim to automate the discovery of syntactic frames (McKee and Maloney 1992; Brent 1993; Manning 1993).

Chapter III explains the overall process in detail, and discusses the tools used to tag, bracket, collect co-occurrences, smooth, and cluster. Chapter IV describes the different experiments that were undertaken for inducing features, and discusses the schemes used to evaluate the outcomes. The final chapter summarizes the contributions of this research and suggests future research issues.
CHAPTER II
LITERATURE REVIEW

In a historical review of NLP Karen Sparck-Jones distinguishes four phases (Sparck-Jones 1994); around the end of the third phase, in the late 80s, problems of lexical transportability, customization, and knowledge acquisition led to the exploitation of machine-readable commercial dictionaries, which was followed by the recognition that text corpora themselves can be used as resources for validating lexical data (Jacobs 1991). The last stage Sparck-Jones labels the “massive data-bashing period” because of the increasing use of probabilistic approaches as supports (and substitutes) for model-based processing. A good overview of statistical NLP research can be found in a special issue of Computational Linguistics on using large corpora (volume 19, issues 1 and 2).

This chapter outlines some of the previous research on corpus-based and hybrid-model natural language processing systems. First the idiosyncratic nature of sublanguage grammars is described, suggesting the need for an empiricist, data-driven approach. A short section follows which describes some of the experiments with on-line dictionaries, and the limitations of such resources. The next segment characterizes linguistic vs. statistical approaches to NLP as a continuum, and indicates some of the successful contributions of stochastic research to syntactic parsing. Finally, a discussion follows on the lexicon, its structure, and how it fits into different grammar formalisms. A definition of subcategorization features is given, as well as a survey of previous approaches in automatic acquisition of such syntactic frames.
Sublanguages

In the *Mathematical Structures of Language*, Harris developed the idea that sublanguages have a grammar of their own, one which is not necessarily a subset of the standard language (Harris 1968). This suggests that because of “the presence in some sublanguages of structures which are unknown in the standard language . . . no single parsing grammar will be adequate for all types of text (dictionary problems aside)” (Kittredge 1982, 134). Kittredge adds that it is the conventionalization among speakers, their shared habits on word usage, that defines sublanguages. This shared knowledge can be inferred through distributional analysis, and thus a sublanguage grammar goes beyond syntax to express lexical co-occurrence restrictions and structural patterning. Thus, sublanguage grammars contain not only traditional rules about legal terminal strings in terms of parts-of-speech and features, but also rules of selection, which cannot be stated for the language as a whole. For instance, Harris’ example of “washing hydrochloric acid in polypeptides” is not “grammatical” in the biochemistry sublanguage, because the object wordclass of “wash” does not include “acid.” Sager, whose pharmacology program was one of the first practical sublanguage applications, stresses that this selection phenomenon is the distinguishing feature of sublanguages and that it is possible because language in technical domains is “stripped of literary niceties” to concentrate solely on communicating information:

In a sublanguage, selectional word classes have relatively sharp boundaries, reflecting the division of real world objects into classes that are sharply differentiated in the domain. Refined syntactic formulas stated in terms of these classes reflect the types of relations the objects named in these classes can have to each other and thereby provide a semantic characterization of the discourse in the given domain, using grammatical methods of description (Sager 1986, 3).

Friedman (1986) describes how an information retrieval system can use these syntactic formulas to automatically map the text from clinical records into structured formats like “treatment sentence type” or “test-result sentence type.” Another successful
sublanguage application is the Canadian METEO project, which was designed to automatically translate English weather reports to French (Lehrberger 1982). One interesting conclusion of this project was that the English weather-report sublanguage showed greater similarities to its French counterpart than to any other English sublanguage.

Such applications are possible because many ambiguities can be eliminated by exploiting the lexical, syntactic and semantic restrictions of sublanguages: Lehrberger estimates the vocabulary of an aircraft-maintenance sublanguage to be around 40,000, a far cry from the 450,000 entries in Webster's, with a corresponding reduction in polysemy. Fitzpatrick et al., in discussing telegraphic Navy messages, list several examples of how a domain-specific syntax rule (verbs are either transitive or intransitive, but not both) can handle missing data objects more elegantly than a standard grammar, which would have to consider a deletion operation and disambiguate using semantics (Fitzpatrick, Bachenko, and Hindle 1986).

Biber, in his comments on corpus design, presents empirical evidence on register variation (variation across text categories: fiction, academic prose, newspapers, etc.). He finds that

- individual linguistic features are distributed differently across registers, and
- similar features can have different functions in different registers (Biber 1993, 221)

He illustrates the grammatical differences by comparing the distribution of dependent clauses (relative clauses, causative adverbial subordination, and complement clauses), and shows how lexical patterns are distributed differently for “certainty” adjectives (“certain,” “sure,” “definite”). Because probabilistic techniques depend on accurate estimates of relative likelihoods, these differing distributions of linguistic features have important implications with respect to sample (training corpus) size and selection.

Though sublanguage systems are robust when applied to their restricted domain, the obvious tradeoff is that the idiosyncratic nature of the approach makes it extremely difficult
to scale these systems up. Moreover, like most knowledge-based approaches, the sublanguage grammars are difficult to come by; selectional word classes are manually identified from sample sentences. Hirschman (1986a), who takes a very semantic view of sublanguage grammars, has attempted to automate the acquisition of these classes using statistical clustering techniques, but has reported little success because of the circularity problem: distributional analysis requires correct syntactic analysis, but this in turn depends on the availability of sublanguage properties like semantic classes and class patterns. A similar bootstrapping problem exists for exploiting on-line dictionaries (see below). McKee and Maloney (1992), too, use a corpus-based approach to semi-automate the detection of the unique linguistic features of a domain: to identify words likely to have special domain meanings, word frequencies are compiled and compared with frequencies in other texts; this information is used to set the weight of word senses in the lexicon of domain items that are homonymous with general-lexicon lemmas.

**Machine-Readable Dictionaries**

Dissertation work by Amsler and Michiels in the early 80s sparked an interest in extracting information from machine-readable dictionaries to provide computational solutions to linguistic problems ranging from lexical acquisition, word sense selection, syntactic parsing, grammar development, and knowledge acquisition (Amsler 1980; Michiels 1981; for an overview, see Boguraev 1994). Early efforts regarded the dictionary source as a lexical database and attempted to extract primarily syntactic properties of words (like morphology and subcategorization information) that are explicitly stated in the entries. This is reflected by most of the position papers at the 1986 international workshop on “Automating the Lexicon” (Walker, Zampolli, and Calzolari 1986). More recent developments have shifted emphasis to uncovering lexical semantic information, which is encoded across the entire source. A good example of this is the investigation by Jensen and Binot
into automatically deriving selectional restrictions for disambiguating prepositional phrase attachments (Jensen and Binot 1987). Many of the papers that share this view of dictionaries as a lexical knowledgebase (Calzolari 1991) can be found in Briscoe, dePaiva, and Copestake (1993).

Lately, several researchers have taken a hard look at the feasibility and desirability of mining on-line dictionaries for lexical information. Concerns are raised with respect to (lexical) coverage; as Meijs points out, our vocabulary is finite but open-ended (Meijs 1986). In 1986, Walker and Amsler commented on the disappointing results of an earlier experiment at SRI International that compared word lists generated from an eight million word corpus of NY Times news stories with the Webster dictionary: the fact “that almost two thirds (61%) of the words in the dictionary did not appear in the text is not surprising; . . . That almost two thirds (64%) of the words in the text were not in the dictionary is more problematic” (Walker and Amsler 1986, 79). Ironically, another problem with coverage is that for many sublanguage applications a general-purpose dictionary often contains too much information, like obscure senses and rare parts-of-speech of a lemma, and does not provide assistance in selecting among them. This prompted McKee and Maloney (1992) to base the transitivity rating of verbs in their lexicon on corpus analysis rather than dictionary information: the American Heritage Dictionary marked 60 percent of their sample as both transitive and intransitive, whereas in their (MUC) domain they occurred almost exclusively with a transitive reading.

A second concern has to do with reliability; after all, these dictionaries were designed for a very different audience and purpose. Also, reliable identification of lexical information requires a full syntactic analysis of dictionary definitions because defining formulae often must rely on ambiguous words in their patterns (Ravin 1990; Klavans, Chodorow, and Wacholder 1990).
Because of the above concerns, much NLP research in the early 90s has shifted to corpus-driven acquisition of lexical information. In fact, even lexicographers have turned this direction; the Cobuild dictionary is one of the first to be compiled by examining a large collection of “English texts, spoken and written, running to many millions of words” so it can be “based on hard, measurable evidence” (Sinclair 1987, xv). However, on-line dictionaries remain a supplementary source that can be tapped (for example in stochastic part-of-speech tagging: Weischedel et al. 1993) and as a benchmark for recall/precision evaluations (Manning 1993).

Massive Data-Bashing

Back in the 50s and 60s, several researchers performed statistical analyses of natural languages (e.g., Guiraud 1959; Muller 1964). As early as 1948, Shannon proposed modeling sentence generation by a stochastic process. Chomsky, whose work in transformational grammars strongly influenced the direction of NLP research for the next few decades rejected any probabilistic view as inadequate and inappropriate, in part because it cannot account for dependencies that hold at unbounded distances (Chomsky 1957). He did, however, leave the door open to statistical research:

Given the grammar of language, one can study the use of the language statistically in various ways; and the development of probabilistic models for the use of language (as distinct from the syntactic structure of language) can be rewarding... One might seek to develop a more elaborate relationship between statistical and syntactic structure than the simple order of approximation model we have rejected. I would certainly not care to argue that any such relation is unthinkable, but I know of no suggestion to this effect that does not have obvious flaws... (quoted in Joshi 1994, 496).

The current interest in probabilistic approaches stems from their demonstrated utility in speech processing. Throughout the seventies, speech systems used hand-generated rules for their acoustic models (which translated a waveform, represented as a real-valued vector, to phonemes) and language models (which combined phonemes into word sequences). These brittle systems never scaled up to large vocabularies. Magerman (1994) accounts
how a seminar in 1980 at IDA on Hidden Markov Models revolutionized the way the recognition problem was approached: instead of manually encoding the phoneme classes, these intermediary representations can be viewed as “hidden,” and can be “discovered” automatically by using the Expectation-Maximization (hence EM) algorithm (Dempster, Laird, and Rubin 1977). To “port” the system to a new speaker simply requires retraining the statistical model by feeding it spoken and written versions of the same text.

Advocates of statistical NLP explain that linguistic knowledge, instead of being stored in the form of a lexicon and grammar rules, is represented as probabilities, or statistical parameters. Many of the traditional knowledge-based researchers remain skeptical, and some of their antagonism was voiced during the 1993 EACL conference (Solomon and McGee-Wood 1994). However, it may be more useful to view theory-driven and empiricist approaches as a continuum; Kanal and Chandrasekaran (1972) argue that a problem should be partitioned, and at each level a choice be made for a solution that is either linguistic, statistical, or in between. The decision should depend on the nature and quantity of a priori knowledge, and whether the data source is noisy or uncertain.

Hatzivassiloglou (1994) reports on a series of experiments to quantify the benefits of a combined approach: he measures the effects of augmenting his statistical-based semantic clusterer of adjectives with four shallow linguistic heuristics (e.g., co-occurrence counts restricted to “nouns in vicinity”), and finds that the improvements are statistically significant and can be generalized to other applications.

Distributional methods have been applied to automatically deduce lexical ambiguity rules, a.k.a. part-of-speech (hence POS) tagging. The high success rates that have been achieved show the feasibility of using probabilistic techniques for linguistic analysis. Several researchers are currently attempting the more difficult task of using probabilistic techniques to derive higher grammatical structures (parse trees) either by tuning an existing grammar to a specific corpus or by building one from scratch.
There are two different ways to calculate probabilities from a corpus; with unsupervised training, the probabilities are acquired automatically using Baker's (1979) inside-outside algorithm: the probabilities of an ambiguous, unanalyzed corpus are repeatedly reevaluated, until they finally converge. More precise probabilities can be achieved by manually creating an unambiguous training corpus, which is referred to as supervised training. Despite the extra preprocessing, this latter approach is more common.

Part-Of-Speech Tagging

POS tagging is the process of assigning to each word a label which indicates the linguistic category of that word in the context of a sentence. Several tagsets exist, the oldest of which is the Brown tagset (Kucera and Francis 1967). It has 184 different tags consisting of general wordclass categories (noun, adjective, article/determiner, verbal form, etc.) and occasional extended information like form, aspect, and case. Another popular tagset, that of the Penn Treebank (Santorini 1991a), contains a reduced set (48) that is less detailed. Hidden Markov Model (hence HMM) tagging was suggested by Mercer during a lecture in the mid-80s at MIT, which Church attended (Magerman 1994, 15), and a little known paper on the subject was published by Derouault and Merialdo (1985). The British team of Atwell, O'Donoghue, and Souter was one of the first to use a Markov model to automatically tag a corpus; their CLAWS system tagged the LOB corpus with a success rate between 93 and 94 percent, but their research was not published until much later (Garside 1987). One of the best known models is probably Church's stochastic parts program for AP news stories (1988); it selects part-of-speech tags that optimize both the lexical and contextual probabilities (estimated from the manually tagged Brown corpus). Since then, several probabilistic taggers have been developed (e.g., Boggess, Agarwal, and Davis 1991; see Dermatas and Kokkinakis 1995 for a list of stochastic, rule-based, and connectionist taggers). The probabilistic taggers are all reported to achieve success scores
that exceed 95 percent, rates which have never been matched by any non-corpus-based system (Atwell, O'Donoghue, and Souter 1991). The highest success rate so far is around 97.5 percent (Brill 1994).

In a fixed-order Markov model, a matrix of transition probabilities between two- or three-tag sequences (bigrams or trigrams) is used to find the most probable path through a sequence of words, thus disambiguating each to the most likely part-of-speech candidate. The transition probabilities are derived from a manually disambiguated training corpus. Church (1988) and Charniak et al. (1993) use this approach.

A Hidden Markov Model is a statistical model of a Markov process, where some component (the set of output probabilities associated with each node) is not explicitly represented in the data (hence hidden); estimating or “learning” the output probabilities can be accomplished using algorithms from information theory, like the EM algorithm or the Viterbi algorithm (Viterbi 1967). Although the training procedure is time consuming, the advantage is that this training can be done using an untagged corpus (e.g., probabilistic taggers by Jelinek 1990 and Kupiec 1992). Such unsupervised training is more common in speech-recognition systems (cf. Baker 1979).

The above taggers typically use a small order model, because the number of possible tag combinations grows exponentially; for example $184^3$ is over six million possible trigrams! Many of these combinations obviously do not occur in a typical training set. One smoothing algorithm that several researchers have adopted to cope with sparse data is to incorporate bigram estimates so unseen trigrams get a non-zero probability. To overcome the difficulty of training fixed-length Markov Models with long memories, Schütze and Singer (1994) have experimented with what they call a Variable Markov Model, which dynamically adapts its history length based on training data. They hope to boost their 95.81 percent success rate by the use of an MRD to help constrain words that did not occur in the training corpus (cf. Kupiec 1992 and Weischedel et al. 1993).
Brill’s dissertation work (1993b) tries a very different approach to incorporate higher-order information: he uses a rule-based learning approach which is able to explore a very large space of contextual factors and actually outperforms HMM techniques (Brill 1993a; 1993b; Ramshaw and Marcus 1994).

Probabilistic POS taggers have successfully been incorporated into rule-based systems (e.g., de Marcken 1990). Weischedel et al. (1993) report having used their POST tagger with Grishman’s Proteus system (Grishman and Sterling 1989), resulting in a speed increase of a factor of two.

“Attuning” a Grammar

Ideally, an NLP system should be designed so that domain-dependent information is isolated, to facilitate porting it to a new domain (e.g., Grosz 1983; Grishman, Hirschman, and Friedman 1983). Several researchers have investigated ways to take an existing broad-coverage grammar, and prune it to end up with an efficient minimal grammar for a specific sublanguage. Slocum (1986) describes an experiment to gather empirical evidence of such sublanguage grammars: by exploiting the built-in debugging routine of the parser they were using, a history was kept of the number of times each rule in the grammar was applied, how frequently it failed to satisfy subcategory features, and how often the phrase built was actually used in the final parse. An inspection of this record shows a marked difference between a corpus of technical phone maintenance manuals and a corpus of sales literature. The author mentions the potential of using these statistics to dynamically tune the grammar rules, but does not offer any concrete suggestions.

Grishman et al. (1984) and Hirschman (1990) report on a successful experiment that trims the PUNDIT grammar to fit two test samples from the medical domain and one from a sublanguage on failure-reports. A training corpus of parsed sentences for a particular domain was selected from the output of the original parser (since PUNDIT does
not rank its parses, selecting the “correct” parse is a labor-intensive task). From this
treebank a set of parent-children relations and their frequency was automatically extracted.
Then the generic grammar was traversed, and every disjunction in a grammar rule which
was unobserved was discarded, together with its restrictions. The lexicon was pruned in a
similar fashion, based on the features stored in the leaves of the parse trees in the training
set. Roughly 62 percent of the production rules and 49 percent of the non-terminals were
never used, and could be eliminated. For the reduced grammar, Grishman et al. found that
both medical samples shared a similar distribution of the remaining rules; the grammar for
the domain of failure reports was quite different, especially for prepositional phrase
attachments, which confirms Slocum's findings about sublanguage behavior. Hirschman
reports that for the reduced lexicon the ratio of word classes per word decreased from 1.37
to 1.08, a substantial decline in part-of-speech ambiguity. The authors also attempted to use
frequency data to control the search for best-first parsing, but reluctantly concluded that the
built-in depth-first backtracking engine in Prolog is ill suited for rule-reordering or
weighted rules.

Rayner (1992) applied a machine-learning concept called “explanation-based
generalization” (Mitchell, Keller and Kedar-Cabelli 1986) to tailor the grammar of SRI’s
Core Language Engine (Alshawi 1992); here, too, a treebank of parsed sentences was
used, but in the form of implicit trees, where each node is the mnemonic name of the
grammar rule that is being resolved at that point. These trees are indexed into and-or trees;
the or-nodes, which are alternate choices of grammar rules to expand, are cut up at
appropriate places to create new rules that consist of chunks of original grammar rules. In
Rayner (1992) the tree-cutting criteria were specified manually; Samuelsson (1994) was
able to automatically induce these criteria by cutting those or-nodes which exhibit high
entropy values: entropy is a measure of disorder, indicating nodes with a high expectation
of variation (i.e., where it is hard to predict what rule will be resolved next). Results are promising, though not as good as specialization with hand-coded criteria.

Simmons and Yu (1992) used an interactive program to create a minimal grammar: a new shift or reduce rule is added only if the rule-base so far does not predict its operation. They achieved an amazing 74 percent reduction in the number of rules, but needed to make several passes to include productions that initially were covered by other rules, but were interfered with as the result of later additions.

The two advantages of trimming a grammar are faster performance and fewer false positives. Because there are fewer productions and a smaller search space, there is a potential for significant speedup. Samuelsson (1994) mentions median speedup times of sixty times (with the hand-coded criteria). Grishman et al. (1984) report more modest improvements, depending on the size of the sentence: 25 percent for short sentences, 300 percent for longer ones, as these tend to backtrack more often. They found the improvement to be proportional to the reduction in productions. Being able to automatically produce a minimal grammar for the PUNDIT system is especially significant in view of the need to exhaust the parse search space, and because the broad-coverage grammar tends to become “bloated” over time, as each port to a new domain adds extensions (Hirschman, Lang et al. 1989).

The second benefit is that the reduced grammar may block some spurious parses that involve rules that were eliminated because they were never used in any final parse; Grishman et al., for example, report a 3 percent reduction in bad parses (1984). Samuelsson (1994) too claims a lower error rate but does not give specifics.

There is however a major drawback to customizing these grammars: coverage suffers, in some cases dramatically, as the grammar becomes overtrained. Rayner (1992) reports that his simplified grammar has a coverage of 90 to 95 percent of the original grammar. Simmons and Yu (1992) did not fare so well; they confess that whereas the
radically pruned grammar correctly parsed all the sentences in the training corpus, only two out of fourteen sentences of new material were analyzed correctly. Their corpus consisted of newspaper articles, which may be too diverse to constitute a sublanguage; Hirschman (1990) reports only a 2 percent loss with the reduced grammar, and a 7 percent coverage loss when combining the pruned grammar and minimal lexicon. However, these appear to be figures for sentences that received a parse, not necessarily the correct one.

Probabilistic and Best-First Grammars

Another approach to customizing a grammar for a particular sublanguage domain is to train it on a sample corpus so that the order in which the space of possible parse trees is searched is “optimized.” All realistic natural language grammars are ambiguous to some degree, and when applied to unrestricted text may produce hundreds of parses. Church and Patil (1982) have demonstrated, for example, that the growth of the ambiguities for prepositional phrase attachment and conjunctions is essentially exponential (approximated by a Catalan number, the number of ways to insert parentheses). Stochastic methods can be used to estimate the likelihood of a parse. The hypothesis is that the most probable parse has a greater chance of being the correct one, which is corroborated by Briscoe and Carroll who assert that “probabilistic information concerning the frequency of occurrence of syntactic rules correlates in a useful (though not absolute) way with the semantically and pragmatically most plausible analysis” (1991, 6). The two most common approaches are either based on the premise that combining local probabilities can quantify the likelihood of entire parses, (e.g., Sharman, Jelinek, and Mercer 1990) or on the assumption that local context provides criteria for choosing among competing rules in the grammar (e.g., Simmons and Yu 1992).
Preference through Rule Ordering

One way to exploit the frequency information collected from a corpus is to reorder the rules of a parser, for example the outgoing arcs of an ATN, or the literal order of rules of a Prolog parser. Nagao and Nakamura (1982) have performed such an experiment on a Japanese parser, but reluctantly conclude that the learned order does not seem to affect performance very much, which is echoed by Hirschman (1990). Atwell, O'Donoghue, and Souter also question the benefit of such reordering; for systems that return all syntactic analyses, the changes will only affect the order and timing of the individual solutions. For systems that return only the first parse, the reordering does not necessarily cause the best parse to be returned: the depth-first search algorithm tends to focus first on trees with high frequency values for the upper nodes, but this may force rule matches lower in the tree which have low counts (1991, 7).

Probabilistic Context Free Grammars

Wetherell (1980) introduced the notion of probabilistic context free grammars for artificial (programming) languages: each production is assigned a probability which expresses the likelihood with which it, rather than other rules, is applied to rewrite a non-terminal. The probabilities of all the right-hand-side productions of a non-terminal must add up to one. The probability of a parse is simply the product of the probabilities of each rule in the derivation, because with context free grammars, the application of a rule depends solely on the grammar symbol on the left, not on context. In the same way that finite state machines can be generalized to the probabilistic case, a probabilistic Context Free Grammar (hence CFG) can be trained by using an extension of the forward-backward algorithm called the inside-outside algorithm (Baker 1979). Despite the fact that in many cases the independent rule assumption does not hold, several researchers have transported this idea
to natural languages and achieved different degrees of success depending on the training method and scoring function.

Fujisaki (1984) was one of the first to build a probabilistic CFG. He chose the unsupervised training approach (i.e., frequency statistics were drawn from both correct and incorrect parses), using the inside-outside algorithm: each grammar rule is assigned an *a priori* probability as an initial guess, and these probabilities are repeatedly approximated using the probabilities of the different parses of ambiguous sentences until they converge.

The intuitive idea is that if probabilities are continuously re-estimated in a manner which increases the overall probability that the set of sentences in the training corpus were generated (regardless of syntactic analysis) by the grammar, this will assign probabilities to individual rules which accurately reflect their frequency of use in the correct analyses of each sentence. (Briscoe and Carroll 1991, 8)

Fujisaki reports very high success rates: 91 percent for the magazines, 94 percent for the correspondence, up from 65 and 63 percent respectively for the non-probabilistic grammar; the better score for the correspondence may have been due to its lower number of average ambiguities per sentence (48.5 vs. 13.5).

Sharman, Jelinek, and Mercer (1990) did a similar experiment with a CYK parser, but used a different grammar and a treebank of AP newswire stories. They report results that are no better than those of Fujisaki (88 percent vs. 60 percent for the non-probabilistic grammar), which is surprising because of two reasons; first, they used supervised training which normally allows for more accurate probability estimates. Secondly, they used the Generalized Phrase Structure Grammar (Gazdar et al. 1985) which supports a more sophisticated language model. Briscoe and Carroll suggest that the high success rate of Fujisaki may somehow have been tied to the specific corpora chosen, and that it may also have benefitted from the Greibach Normal Form format, which allowed the grammar, and hence the probabilistic model, to integrate richer lexical information (1991, 8-9).

Chitrao and Grishman (1990) stuck with unsupervised training, but used a chart parser on a corpus of Navy messages. The relative improvement they achieve is similar to
the above experiments, but their original grammar did a poor job of parsing the selected sentences even though the grammar is based on the supposedly broad-coverage Linguistic String Grammar (32 percent correct before, 45 after training). Weischedel et al. (1993) used the same probabilistic CFG and found that with supervised training (using the Penn Treebank), as few as eighty sentences proved adequate for a grammar of approximately one thousand rules; they report a reduction in error rate by a factor of two.

Because the inside-outside algorithm is most naturally formulated using a bottom-up chart parser, there has been little research in Earley-based probabilistic parsers, even though they exhibit best-known complexity on several grammar subclasses (Earley 1970). Stolcke (1993) has developed such an efficient probabilistic top-down CFG, and addressed the problem of returning partial parses for ungrammatical input, a common stumbling block for predictive parsers (Magerman and Weir 1992). Two other advantages of Stolcke’s approach are that the grammar rules do not have to be specified in some normalized form, and that the parser can take advantage of partially bracketed input, which Pereira and Schabes (1992) have demonstrated can help EM-based estimators avoid suboptimal parameter solutions.

Context Dependent Grammars

Like Chitrao and Grishman, Magerman and Marcus (1991) opted for using a chart parser and unsupervised training, but in addition they preprocessed their corpus with Church's (1988) part-of-speech tagger. They concentrated on improving the scoring function by using context-sensitive conditional probabilities, which are calculated on the basis of the POS of the input, and the probability of the parent rules that predicted it. They achieved an accuracy rate of 88 percent when parsing a sample from the Voyager domain. This does not seem much of an improvement over the 87.5 percent score of the untrained
PUNDIT grammar, but the probabilistic parser achieved this success rate without the benefit of hand-generated adjustments and semantic post-processing.

Not only should a probabilistic parser recognize that the probability of productions may change according to context, but it should also distinguish the probability of different derivations when an identical rule can be applied multiple times in several ways. Therefore Wright (1990) and later Kochman and Kupin (1991) suggested using LR parsing as a more linguistically adequate grammar formalism. Wright describes a way of distributing the probabilities from a CFG to an LR parse table in such a way that the probabilistic ranking is preserved. Kochman and Kupin define a tractable algorithm for computing a scoring function by matrix manipulations (cf. Wetherell 1980), but do not discuss any practical tests. Briscoe and Carroll (1991) experimented with assigning probabilities directly to the parse table, which encodes information about the left and right context of a parse. In their testcase of noun definitions from the LDOCE they achieved success scores of 85.9 percent and a slightly lower score of 81 percent with the probabilities assigned to the actions and the transitions respectively.

Inducing a Grammar

Some researchers have abandoned existing grammars altogether, and have investigated ways of building a grammar from scratch. The idea is to take a large corpus and induce the grammar that is implicit in it. The methods differ in the amount of corpus preprocessing that is required, ranging from a tagged corpus, to bracketed training input, and a fully parsed treebank.

Simmons and Yu (1992) performed an experiment to acquire a context sensitive phrase structure grammar from a corpus of news stories. The corpus need not be preprocessed, as it is the linguist who provides the knowledge through focussed interaction. When parsing a corpus with such a grammar, a scoring function is used to select the
most appropriate rule, as there may not be an exact match considering the large number of possible contexts. The grammar created by Simmons and Yu was 96 percent correct when parsing the training corpus; the errors were all false positives; no results are reported for new material.

To validate the premise of Simmons and Yu that the use of context information can significantly reduce ambiguity, Miller and Fox (1994) designed a shift/reduce system that automatically acquires a context-dependent grammar, and compared its performance with a chart parser that uses a probabilistic CFG, also automatically learned. Instead of requiring human interaction, the context-dependent parser was trained using the Wall Street Journal Treebank of parsed sentences, which were converted into sequences of shift and reduce operations. Test results show that recall for both parsers is equal, but the probabilistic CFG has higher precision and fewer crossings (Black et al. 1991), leading the authors to conclude that this modeling is more resistant to overgeneration.

Magerman and Marcus (1990) experimented with a system which does not acquire an explicit grammar, but converts a corpus into a treebank of unlabelled parse trees. The corpus has to be preprocessed to annotate it with part-of-speech labels, but this could also be done using probabilistic methods (see above). The system deals only with part-of-speech n-grams which it views as stochastic events, not with the actual words. Fano’s (1961) mutual information statistic was generalized to include the left- and right-context of the n-gram and acts as a measure of continuity to find constituent boundaries; the mutual information statistic indicates the probability of two word(group)s belonging together. Magerman and Marcus hypothesized that if $a_1...a_n$ is a constituent, its probability is much larger than the probability of the same tag-sequence with one more element added ($\text{Prob}(a_1...a_na_m)$) because the constituent appears in a variety of contexts. Tests using the Brown corpus have proven this hypothesis to be correct most of the time. The algorithm
had an accuracy of two or fewer misplaced parentheses on short sentences (10-15 words), and up to five errors for longer ones (up to thirty tokens).

Using the same transformation-based error-driven learner that achieved very high success rates in POS tagging, Brill (1993a) acquired a transformational grammar that generates unlabelled bracketed trees similar to the ones of Magerman and Marcus (1990). Starting with a small training corpus that has been tagged and bracketed naively with right branching, a set of twelve transformations is repeatedly applied and compared against the proper phrase structure; the resulting parser is an ordered set of simple structural transformations to reduce error. The results of this symbolic system on a test set from the ATIS corpus shows an accuracy rate in terms of noncrossing constituents of 91.12 percent, which compares very favorably with the 90.36 percent performance of a statistical parser (Pereira and Schabes 1992), especially considering that the training corpus that was needed was only 21 percent as large as that used to train the inside-outside algorithm.

The title of Magerman’s (1994) dissertation work on acquiring a parser clearly shows his extreme position in the statistical vs. symbolic approach debate: Natural Language Parsing as Statistical Pattern Recognition. The goal of his SPATTER system is to collect statistical information from a large corpus of tagged text in order to automatically discover all the disambiguating criteria needed during the parsing process. SPATTER does produce complete parse trees, using a feature-based representation: given the input words, all of the other features (POS tags, constituent labels, and connecting edges) are predicted by the distributions assigned to a statistical model. A decision tree model (Breiman et al. 1984) is constructed which learns the disambiguation criteria based on the context of the decisions that led to the final parse tree; then an EM algorithm trains the hidden derivation model to maximize the total probability of the training corpus. A second EM algorithm finally smooths the decision trees to maximize the probability of the parse trees in a held-out test corpus. Two important conclusions came out of this research: existing statistical
models (decision trees) can use disambiguating long-distance context; and while SPATTER outscored a “highly respected grammarian” (p. iv) on the cross-brackets score, less than half of the parses were an exact match with the treebank. This last observation led Magerman to tone down his initial assertion that linguistics can be completely replaced by statistical analysis into a recognition that linguistic input is still crucial, especially in identifying more sophisticated disambiguation criteria for the parser to learn.

Finally, there are some systems that attempt to induce a grammar automatically from a bracketed corpus. Prolog has a built-in grammar rule notation called Definite Clause Grammar, so a grammar, when converted to the appropriate notation, can be run directly by the interpreter, without the need for building a parser. Atwell (1988) has written a program to extract a phrase-structure grammar/parser from the LOB corpus. However, the convenience of interpreting the grammar directly has serious computational drawbacks: Prolog performs an exhaustive depth-first search with backtracking, and spends a lot of time recomputing constituents, unlike the “well-formed substring” mechanism of chart parsers. Secondly, because Prolog treats grammar rules as code, it becomes impossible or unwieldy to fit these large grammars in memory; one of the training corpora yielded over eight thousand rules, which still left many constructs uncovered. Atwell suggested using the frequency information to threshold out some of the low-probability rules, but this would negatively affect the coverage of infrequent structures.

These limitations moved Atwell, O'Donoghue, and Souter (1991) to try a different approach, which is an extension of the CLAWS part-of-speech tagger. Their hybrid system used a first-order Markov model as an initial step to insert opening and closing brackets at tags that flag phrase boundaries. Because a Markov model is not powerful enough to capture long distance dependencies, a second, tree-closing phase was applied based on a more powerful stochastic context free model; heuristics and a constituent likelihood scoring function together select among the candidate parse trees from the combinations of multiple
possible insertion points and possible multiple missing boundary-brackets. Unfortunately, the bracket-insertion rules and the heuristics had to be manually handcrafted to fit a corpus.

The Lexicon

The important role the lexicon plays in NLP systems was epitomized by two recent workshops that discussed representational needs and acquisition strategies (American Association for Artificial Intelligence 1995; IFIP working group for NLP and knowledge representation 1995). While there is broad consensus about the types of information that should be contained in a lexicon (Atkins and Levin 1991), there is not much commonality among current systems with respect to representation schemas, especially for lexical semantics. Moreover, most lexicons still are hand-crafted and very modest in size. Pustejovsky, the “ensign” of the philosophical, theoretically sound approach to lexical semantics (Basili, Pazienza, and Velardi 1994, 22) reports on an experiment on automatic acquisition of lexical “qualia” structures (Pustejovsky, Bergler, and Anick 1993), but the results are limited and do not scale up (Basili, Pazienza, and Velardi 1994, 22).

With the increased availability of MRDs, there was a flurry of research in exploiting them for lexical information, but after the initial excitement, reality set in that these sources require extensive modifications to be accessible (Boguraev et al. 1987). Moreover, dictionaries too are hand-crafted, so this simply passes “the need for tools that acquired lexical information from the computational linguist to the lexicographer” (Manning 1993, 235). There have since been several advances in using corpus-based techniques for “learning” lexical data, a few of which are described below.

Part-Of-Speech Information

POS information is a mandatory item in the lexicon; as already mentioned, several statistical acquisition methods have been developed that can claim very high accuracy. But as Allen (1995) points out, even at 95 percent, the chance that every word is correct in a
sentence of twelve words is less than half, which can prevent a parser from finding a parse, or may result in false positives. Consequently some NLP researchers have shied away from relying on stochastic tagging, and use other heuristics instead (e.g., Brent 1992). On the other hand, coming full circle, several researchers (Kupiec 1992; Weischedel et al. 1993) have integrated on-line dictionaries in their stochastic taggers to help restrict the set of possible tags for unknown words.

Syntactic Features

The semantic arguments of words can be represented by different syntactic phrases. Compare for example the following sentence pairs:

- Mary wants a husband.
  * Mary wishes a husband.
- John considers the car beautiful.
  * John says the car beautiful.

Chomsky (1965) calls these properties the “subcategorization frames” of words. In rule-based NLP systems, they are heavily used to constrain the number of allowable parses. Sager’s Linguistic String Project (LSP) grammar, one of the early broad-coverage NLP systems, recognizes ten lexical subclasses for adjectives, twenty-three for nouns, and thirteen for verbs, plus two selection attributes and fifty-five object attributes for the latter category. These features are stored in the lexicon, and are tested by the restriction augmentations that are associated with the grammar rules (Fitzpatrick and Sager 1981, 322). In the above examples, “want” and “consider” have the LSP features NSTGO and SOBJBE respectively, which make the sentences grammatical. McKee and Maloney (1992) present some empirical evidence that these frames vary according to domain: using

1See appendix A for a brief explanation of the subset of LSP features used in this dissertation. NSTGO is roughly equivalent to the class of transitive verbs, including verbs that can drop their object (e.g., “he reads [a book]”). SOBJBE consists of verbs that can take the pattern noun plus predicate, where the latter can be a noun string (“they consider him their savior”), an adjective string (“we thought him interesting”), or an adverb string (“they assumed him on the premises”).
the mutual association ratio (Church and Hanks 1990) they analyzed the prepositions for which verbs subcategorize in the MUC corpus and the Dow Jones corpus and use this information to bias attachments in their SOLOMON system (see below).

(Semantic) Selectional Restrictions

There is a very strong correlation between these features, or subcategorization frames, and word meanings. Brent (1992) quotes an example from (Zwicky 1970) that all verbs with three semantic arguments, of which one is realized as a direct object, and another as a tensed clause, are verbs of communication (e.g., “he warned him he would be back.”). Child psychologists are divided whether lexical acquisition occurs by learning the meaning first and then inferring the frames next by exploiting the regularities between meaning and subcategorization (semantic bootstrapping hypothesis, e.g., Pinker 1989) or whether children use the frames to restrict their conjecture about new word meanings (syntactic bootstrapping premise, e.g., Gleitman 1990). Because of this semantic connection, investigations in automatic acquisition of such subcategorization features are sometimes classified under semantic research, while in other cases it falls under the heading of syntactic approaches to NLP. A case in point is the corpus-based approach of Hindle and Rooth (1993) to prepositional phrase attachment, a problem which probably cannot be resolved using either syntax or semantics alone. McKee and Maloney (1992) classify their own work on prepositional preference and transitivity features as syntactic knowledge, whereas they label mass/count nouns and partitives as semantic features.

The traditional approach to semantic disambiguation of using selectional restrictions can be traced back to the work of Katz and Fodor (1963). Word senses are grouped into semantic type hierarchies, sometimes generalized to semantic networks (Quillian 1968), which allow the system designer to state restrictions in terms of general classes. For example, in Sager’s (1981) Linguistic String Project grammar, the verb selection attribute
NOTNOBJ:NHUMAN rejects “the government abolished the boy,” and NOTNSUBJ:
NSENT3 throws out “the trouble studied.”

In the past, semantic restriction and abstraction hierarchies were typically hand-
crafted. Some software tools have been developed to facilitate this process (e.g., Ayuso,
Shaked, and Weischedel 1987; Grishman, Hirschman, and Nhan 1986), and the Wordnet
project has made a comprehensive lexical database available to the research community
(Miller et al. 1990). In his dissertation Resnik (1993) uses this resource to discover
selectional preferences and to identify appropriate word senses. Basili, Pazienza, and
Velardi too have employed Wordnet to overcome the sparsity of their data and to help
generalize their selectional patterns. Some of the selectional restrictions that their system
acquired semi-automatically are presented to illustrate how they vary markedly by domain,
especially for verbs (1994, 26).

Two other recent studies in automated sense disambiguation are works by
Yarowsky (1992), who applies the concept of mutual information, and Schütze (1993a),
who uses a vector-based approach. Thus a second strategy for resolving semantic
ambiguities is increasingly being explored, which is based on statistical analysis of corpora;
the frequency of semantic roles between senses is computed and used to give preference to
the more common combinations. Examples of such an approach can be found in (Grishman
and Sterling 1992; Weischedel et al. 1993; and Hastings 1994).

The Lexicon and the Grammar Proper

The division of responsibilities between the grammar (non-terminal productions)
and the lexicon (leaf node definitions) is somewhat fuzzy, and depends on the particular
grammar formalism the designers subscribe to. Some recent forms, like Lexical Functional
Grammars (Kaplan and Bresnan 1982) and Categorial Grammars (Wittenburg 1986) were
motivated by a desire to demonstrate how the solution to specific parsing problems could
be relegated to the lexicon, rather than “requiring omniscient grammatical rules” (Amsler 1994, 157). A good overview of some of the different approaches can be found in chapter four of (Allen 1995) and in (Sells 1985).

In order to build broad-coverage NLP systems, context free grammars must be extended by adding feature sets to the constituents, and by expressing rules in terms of those features structures rather than by simple grammatical categories. The lexicon must then specify the relevant feature value restrictions. Augmented CFGs have been popular in NLP since the introduction of attribute grammars by Knuth (1968) for programming languages. Using features allows the grammar writer to handle subcategorization and agreement in a concise and intuitive manner. If the set of features is finite, the augmented grammar is expressively equivalent to a CFG. Early systems like (Sager 1981) and (Robinson 1982) specified these annotations as arbitrary lisp code. The tradition of Generalized Phrase Structure Grammars has formalized many of these augmentation concepts, governed by the principle of feature propagation.

Feature structures can be generalized to such a degree that the CFG itself becomes redundant. The principle behind unification grammars (Kay 1982; Shieber 1984) is that almost the entire grammar can be specified as a set of constraints between feature structures, which are represented as Directed Acyclic Graphs; unification is implemented in terms of graph matching. One variant of unification grammars, called Lexical Functional Grammars takes this idea to an extreme, and encodes most of the grammar in the lexicon. Lexical entries not only specify the feature values, but also what slot in the constituent structure they will fill. Another grammar formalism where the entire grammar consists of lexical items and their associated structures is the Tree Adjoining Grammar (Joshi 1985). Because “really ‘meaningful’ statistical dependencies are between words (lexical items) mediated most likely by grammatical relations,” Joshi claims that lexicalized grammars like the ones mentioned above are better suited for integrating structural and statistical
information (Joshi 1994, 498). A final example of “radical lexicalism” (Karttunen 1989) is the Categorial Grammar. Solomon and McGee-Wood (1994) describe $L$, a prototype which induces such a grammar/lexicon from a corpus of children books, to show the viability of a totally lexical approach. Complex categories like “verb” are functions, and it is this visible information structure which makes Categorial Grammars ideal targets for corpus-based induction (p. 124).

Manual Approach: COMLEX

The Proteus Project at New York University has been working since 1993 on building a broad-coverage lexicon with detailed syntactic subcategorization information (Grishman, Macleod, and Meyers 1994; Macleod, Grishman, and Meyers 1994). Although this manual undertaking is based on information culled from various MRDs (and on introspection, supported by a ninety MB corpus of heterogeneous text), it has been designed specifically for computational applications, with a richer feature set and special attention to consistency in coding. In spite of Daelemans’ (1995) reservation about the desirability or feasibility of building such a theory-neutral lexicon, the designers felt this was the right approach, because the current state of automatic lexical acquisition of syntactic features (see below) deals only with a limited range of features, and does not handle low-frequency words well.

The COMmon LEXicon has subcategory information for nouns, adjectives, and verbs (nine, fourteen, and ninety-two complements respectively) for the 38,000 non-British words in the OALD. The verb codes in this MRD were supplemented with information from the Brandeis Verb Lexicon, which also influenced the lisp-like notation of the feature-value lists. Most of the initial information came from the detailed Linguistic String Project lexicon (Sager 1981), with additional material from the LDOCE and the
ACQUILEX project (Sanfilippo 1992). The lexicon is made available to the research community through the Linguistic Data Consortium.

Automated Approach: Data-Driven

One of the side-effects of aiming to be a broad-coverage lexicon is that COMLEX may not sufficiently capture the idiosyncracies of sublanguage domains; the designers admit that sometimes the acceptability of a proposed feature is “a matter of degree,” rather than absolute, and inclusion is left at the discretion of the lexical enterer: in certain contexts “dead” is gradable (“deader than a door nail”), and “apple” can be a mass noun (“Apple is a wonderful flavor”) (Grishman, Macleod, and Meyers 1994, 4). These “contexts” have a lot to do with sublanguages, and in some domains certain features associated with a word would be appropriate while in others they would just be “noise” or, worse, lead to incorrect analyses. Data-driven automated lexical acquisition of syntactic features may be a superior approach, even if less comprehensive.

Calzolari and Bindi (1990) come to this very conclusion in their paper on large lexical database requirements: even if an on-line dictionary like the LDOCE were available for Italian, nothing “can be achieved with the same ‘objectiveness’ [as statistical corpus analysis], and certainly not to the same extent, by other means such as e.g. lexicographers’ intuition” (p. 56). The authors used the mutual association ratio and the dispersion measure (Bortolini, Tagliavini, and Zampolli 1971) on the twelve million word Italian Reference Corpus (Zampolli 1988) and observed how this gives statistical evidence to syntactic relationships between word-pairs. These relationships include sentential complementation and arguments marked by prepositions: for example, for their Italian corpus, “risk” (noun and verb) subcategorizes for only the preposition “of,” and “speak” is associated with “about” and “with” but not “to.” The authors conclude that “these quantitative data can be
associated to the different subcategorization frames” (Calzolari and Bindi 1990, 56), but do not offer any concrete method for automating this mapping.

**Fill in the Blanks with a Probabilistic Parser**

Weischedel et al. (1993) report on some preliminary experiments that acquire lexical syntax (and limited semantics) by using a probabilistic CFG in combination with a stochastic tagger. The experiment consisted of 160 sentences from the Penn Treebank in each of which one word was deliberately undefined to the system. Then their POST tagger hypothesized possible parts-of-speech, and rule probabilities acquired through supervised training selected the most likely parse from the multiple trees, from which the system concluded syntactic and semantic facts. For example, in “Who in division four is the contact for MIT?” the system learned that “contact” is a count noun, with third person singular agreement, and ISA “person” class. The authors are currently exploring heuristics for combining the learned sets of lexical features from different examples: sometimes they can be unified, but there are also cases where they must be listed as alternative options.

**Inferencing Lexical Syntax: Child’s Play**

In the area of acquisition of subcategorization frames, the work by Michael Brent (1992, 1993) is probably best known, and most referenced. He approaches the problem from a unique perspective, trying to make it a plausible model for the way children perform syntactic bootstrapping. His hypothesis is that children first learn the syntactic function of a few extremely common and highly frequent and informative words, like prepositions, determiners, pronouns, function morphemes, etc. They then use these approximate cues to discover relevant syntactic structure without needing to know the meaning of all the words. Because of the nature of the string-local surface cues, the parses will be incomplete and often incorrect, but the effects of misconstrued input (or ungrammatical input, for that matter) will be reduced by applying a statistical inferencing algorithm.
The first step in Brent’s algorithm is to use function-morpheme cues to locate verbs and identify phrases that might be their complements. The heuristic used is a compromise of simplicity, frequency, and reliability, and works by considering a word a verb if it has been observed to occur both with and without the “-ing” ending, except if it ever appears immediately following a known determiner or preposition. Brent uses similar cues for identifying four complement types (NP, PP, VP, and Complementized Phrase), again qualifying the matches with an approximate heuristic (the complements must occur “near” enough the verb, and the possibility of the NP complement being subject must be ruled out). Because of the heuristic nature of the identification phase, it is followed by statistical error reduction to minimize the effects of misconstrued input. The author developed a statistical model to estimate the error rate of each frame; each verb is analogous to a biased coin, with each occurrence of a verb corresponding to a coin flip; i.e., each occurrence of a verb without a cue for a frame counts as a small bit of evidence against that verb subcategorizing for that frame. Once the miscue rates are known, the evidence for each frame can be weighted using standard methods of hypothesis testing from binomial frequency data, “much as one would consider the number of heads turned up by a coin to determine whether or not it is fair” (Brent 1992, 447). The above approach was implemented in a system called “Lerner,” and tested using a simulation on the CHILDES corpus (MacWinney 1991) which discovered eight verb subcategorization frames (the four complement types mentioned above, plus combinations of those same complements with NP).

“Lerner” Grown Up: Partial Parsing

In an earlier paper, Brent’s Lerner acquired five different subcategorization frames (Brent and Berwick 1991). While the authors concluded optimistically that they “foresee no impediment to detecting many more” (p. 344), later work (Brent 1992, 1993) still only
reports on eight frames; the approach is not truly extensible because the small amount of false triggers is a consequence of a strong dependence on cues that are very accurate predictors of subcategorization frames; but for many features no such reliable cues exist (Hearst 1992). To overcome this limitation, Manning (1993) tried a slightly different approach: by pre-processing the corpus (four million words of NY Times news wire) with Kupiec’s stochastic POS tagger (Kupiec 1992) and by using a finite state parser, he is able to infer nineteen different subclasses. The partial parser has a simple NP recognizer, but otherwise skip-scans the input until it finds a verb and subsequent complements. Heuristics still play a role, for example in discovering the end of the complements, and when the parser finds itself in uncertain waters it skips ahead to the next sentence. Manning uses the same statistical filtering method as proposed in Brent and Berwick (1991), but because his data are much less reliable, he selects a much higher value for the upper bound on the probability that a verb which does not subcategorize for X will nevertheless appear with a cue for X.

Manning reports learning an average of 1.6 frames for 3,104 verbs, with a recall of 82 percent. To get a pessimistic lower bound, the acquired frames for forty random verbs were compared against those in the OALD, which yielded a recall of only 43 percent, but a high precision score of 90 percent (which is attributable to the high upper bound setting mentioned above).

The Company You Keep: Prepositional Phrase Attachments by Lexical Association

Hindle and Rooth (1993) tackle one specific subcategorization feature: prepositional phrase attachment. Since structure-based methods like right association (Kimball 1973) and minimal attachment (Frazier 1978) do not appear to be good predictors, the authors hypothesize with Whittenmore, Ferrara, and Brunner (1990) that the solution may lie in lexical preference, and they propose to discover this lexical association through
distributional analysis of a large corpus. The authors assume even more *a priori* grammatical sophistication than Manning (1993): the corpus is preprocessed with Church’s (1988) PARTS tagger and parsed by the Fidditch parser (Hindle 1983). From thirteen million words of AP news stories they extracted over 220 thousand ambiguous verb-noun-preposition triples. These were converted into bigram counts by using heuristics: e.g., if the noun was a pronoun or the verb was passive (and the preposition was not “by”), the preposition was attached to the verb; if the noun phrase was in a pre-verbal position like subject, the noun claimed the preposition. Because many triples cannot be resolved with these simple heuristics, and because the source, as in the above approaches, is imprecise, log likelihood ratio statistics (Mosteller and Wallace 1964) were used to decide the remaining cases. The reason the input was noisy was that the Fidditch parser often yielded imperfect parses, and because sometimes the preposition attached to a higher locus than the noun or verb in the triple. The Lexical Association score was the log of the ratio of the probability of verb attachment over the probability of noun attachment, where the probabilities are estimated from the table of co-occurrence counts. The sign indicated which attachment was more likely, and the magnitude indicated the strength of confidence.

Both authors manually disambiguated one thousand random triples (without looking at the context), with an overall error rate of around 15 percent, which indicates the limitations of an approach that does not consider semantics or discourse analysis. The Lexical Association procedure fared moderately well, with overall precision of 80 percent. By increasing the LA score threshold, this was improved to 91 percent, but at the expense of a severe decline in recall for verb attachments. As in Manning’s paper, a final experiment compared the lexical association procedure with the features stored in the Cobuild dictionary; the latter had similar precision results but poor recall figures because in most cases the dictionary provided insufficient information, proof again of the limitations of MRDs and the superiority of using a data-driven approach.
SOLOMON: Syntactic or Semantic?

In an attempt to equip SOLOMON, SRA’s large knowledge-based NLP system, with statistically acquired knowledge, McKee and Maloney (1992) experimented with learning subcategorization frames from the Dow Jones and MUC corpus, both of which have been tagged with POS information. Using the mutual association ratio and a window of two to the right, prepositional preferences are learned, and for values exceeding a threshold of 2.0, preference is given to verbal attachment. Another feature being inferred is transitivity. A heuristic is used to determine if a verb is used in a transitive way (when followed directly by a surface object, or when used passively), and the ratio of transitive use divided by total number of occurrences is entered in the lexicon, to weigh the likelihood of different parses.

McKee and Maloney also investigated two subcategorization frames for nouns, which they curiously categorize as semantic knowledge: mass/count and partitives. Mass nouns are detected by high ratios of singular to plural uses, corroborated by a low score for the mutual information statistic with preceding indefinite articles; count nouns conversely have a low ratio and high score respectively, and do occur occasionally preceded by numbers. The authors conclude that all the above information was easily assimilated in the parser, without requiring major changes, and helped to improve performance. No specifics however were given about the latter.
CHAPTER III
INDUCING A CATEGORY SPACE

This chapter describes the steps involved in computing a category space for the verbs in the sublanguage domain of physical chemistry. A category space is a multi-dimensional space in which the syntactic category of words is represented by a vector of co-occurrence counts (Schütze 1993b). Proximity between two such vectors, or context digests, can be used to measure the paradigmatic relatedness of the words they represent (Schütze and Pedersen 1993). Paradigmatic relatedness indicates how well two words can be substituted for each other, i.e., how similar their syntactic behavior is. This is not the same as the synonym relationship, which is based on semantic similarity. For example, tests for the chemical domain show a high paradigmatic score for the following pairs: “established-calculated” (0.927), and “maintained-understood” (0.896), but a low score for “established-verified” (0.017), and “maintained-claimed” (0.002), which indicates that the verbs in the latter pairs occur in quite different syntactic environments.

The sample scores above are calculated from a cosine similarity metric, which will be explained below. This chapter first describes the testbed corpus and the pre-processing steps of tokenization, POS tagging, and bracketing. The next segment discusses the bootstrapping of the category space, followed by a description of the “smoothing” of that space by singular value decomposition. Finally, an explanation is given of how proximity in this space is used to compute the paradigmatic relatedness between context digests.
The Physical Chemistry Corpus

One of the prerequisites for doing corpus-based NLP research is to have a “sufficiently-large” training sample, although few formal investigations have been performed on the impact of the size of the data set on performance (Weischedel et al. 1993). The Brown corpus, the first widely-used machine-readable corpus, contains just over one million words from fifteen carefully selected categories of text. The LOB corpus is a similar effort for British English. Another popular source is several months’ worth of on-line news stories from the NY Times and the AP newswire. The Brown corpus is also available in a version completely tagged by hand with POS information. Other augmented corpora, called treebanks, have been created with automated tools and some human post-processing. The Linguistic Data Consortium has made available the Penn Treebank, which consists of approximately four million tagged words and thirteen thousand parsed sentences from the Wall Street Journal (Marcus, Santorini, and Marcinkiewicz 1993). The training corpus used in this dissertation consists of approximately 340,000 sentences from the Journal of Physical Chemistry, consisting of about eight million words.

Many decades ago Zipf discovered that 70 percent of token usage is accounted for by the most frequent 20 percent of word types, mostly short function words (Zipf 1949). This has become known as Zipf’s Law, and is explained by the principle of least effort, since “it [is] easier for a writer or speaker of a language to repeat certain words instead of coining new and different words” (Salton 1983, 60). It turns out that for the technical sublanguage of physical chemistry, as few as 0.6 percent account for 70 percent of the words in the training corpus (even less, 0.5 percent when words are disambiguated by part of speech; see figure 3.1). A possible explanation is that the text is rife with chemical

---

1The number of types in a corpus corresponds to the number of distinct words; the number of tokens indicates the number of running words (Kucera and Francis 1967, 294). For the Brown corpus, 69 percent of the word tokens is accounted for by 2 percent of the word types (reported in Brill 1993b, 44).
names and formulae, each instance occurring but a few times. At any rate, even in the case of generic English, Zipf’s law implies that a large percentage of the vocabulary occurs very infrequently, and this poses a potential problem for stochastic approaches to NLP, such as gathering collocation information. Doug Paul found, for example, that only 58.6 percent of his test set trigrams occurred in a forty million word training set of Wall Street Journal text (reported in Dagan, Pereira, and Lee 1994, 277).

![Figure 3.1. Word Usage Statistics](image)

There are two different ways to tackle the problem of sparse data; the solution chosen by the IBM/Lancaster team was a gargantuan effort to amass a half-billion-word archive for their research in continuous speech recognition (Black, Garside, and Leech 1993). A different path is to use statistical smoothing techniques, like class-based (Brown, Della Pietra, deSouza, Lai, and Mercer 1992) or similarity-based (Dagan, Pereira, and Lee 1994) estimation of co-occurrence probabilities. Magerman and Marcus (1991) used only the POS tags, rather than the actual words, an approach which has been modified and
adopted in this dissertation research for expanding the context digests. But the primary statistical tool used here is called singular value decomposition, a technique which Deerweester et al. successfully introduced to the domain of information retrieval to “estimate the underlying latent . . . structure in the data that is partially obscured by the randomness of word choice . . . .” (1990, 391).

The raw text of the articles from the Journal of Physical Chemistry is formatted using SGML (Standard Generalized Markup Language), a descriptive markup format that reflects the document structure. This text must be “cleaned up” prior to processing, by stripping it of the codes that indicate layout (e.g., `<bdy><sec><st>`), formatting (e.g., `<it>` or `<bd>`), and embedded footnotes and tables. This is part of a tokenization process to identify the relevant lexical units (Charniak 1993), which also includes stripping the punctuation off words, flagging the ends of sentences, and identifying multi-word expressions (e.g., “due_to,” “carried_out,” “as_well_as”) and complex chemical name patterns and formulae. Whereas the latter pattern recognizer is highly domain-specific, the identification of multi-words can be automated using the mutual information statistic (see, for example, Church and Hanks 1990 or McKee and Maloney 1992).

Tagging the Corpus

Context digests contain information about the syntactic environment in which a word occurs. Verbs and nouns obviously occur in different “neighborhoods,” so it is important to disambiguate words that have multiple POS in order to maintain separate vectors for them. Such ambiguity is common for words with an “-s” suffix (e.g., “contracts” is used both as a plural noun and as a third person singular verb), but often there is no morphological clue, and the noun sense (or verb sense) may be particular to the

---

2Written by Dr. Ray Reighart of Chemical Abstracts Service, Columbus, OH.

3Implemented by Lynellen Perry Smith. The physical chemistry articles are used in the AIMS project at Mississippi State University. For a description, see Agarwal (1995).
sublanguage. Some examples of noun/verb ambiguity in the physical chemistry domain are “fit, spread, log, mask, abstract,” and “approach.” Of the 95,421 unique words in the training corpus of physical chemistry, 1,177 occur with both noun and verb POS.

Failure to disambiguate by POS can adversely affect the context digests, a conclusion echoed by Schütze who hypothesizes that “the two components of an ambiguous vector can add up in a way that makes it by chance similar to an unambiguous word of a different syntactic category” (1993b, 154). For example, in his corpus of NY Times news stories, the nearest neighbors in the category space of “transports” were “spokesman, copyboy,” and “staffer” (Schütze did not disambiguate his training corpus by POS because his goal was precisely to induce the tags by using a connectionist approach with context digests as input).

The tagset used to annotate the domain corpus is the one introduced in the Penn Treebank, with a few extensions: auxiliaries and gerunds have their own tags, and subordinate conjunctions are distinguished from proper prepositions (see appendix C). A 160,000 word subset of the corpus was hand-tagged\(^4\) and used as the “gold standard” to train Brill’s rule-based tagger (Brill 1992). Brill’s tagger is an application of what he calls transformation-based error-driven learning (see figure 3.2, taken from Brill 1993b, 34); each word in the training corpus is initially tagged with the tag with which it occurs most often in the hand-tagged corpus, i.e., without regard to context. Tags for unknown words are guessed based on heuristics like suffix, capitalization, etc. A greedy search is then applied, which tries six transformations in turn and keeps the contextual rule which results in the greatest error reduction (calculated by comparing the effect of the transformation on the training corpus with the hand-tagged standard). This learning process is repeatedly applied to correct initial tags and to compensate for newly introduced errors, until the error

\(^4\)Performed by Lois Boggess and Lynellen Perry Smith, as part of the AIMS project.
reduction falls below some predefined threshold. For example, “NN VB PREVTAG TO” is a rule which corrects the tag of the current “noun” to verb, if it follows a word tagged with “TO.” The system learned 277 such contextual rules for the physical chemistry training corpus before converging, after almost two weeks of processing on a Sun 4/690, with 70 MB of RAM and 400 MB of swap space. The rule-acquisition phase of the transformation-based system is slow due to the nature of the greedy search, and because most of the implementation is based on interpreted perl code. See Roche and Schabes (1995) for a much speedier implementation using a deterministic finite-state transducer.

Figure 3.2. Transformation-based Error-driven Learning (Brill 1993b)

One advantage a rule-based tagger has over stochastic approaches is that the linguistic knowledge is made explicit in a relatively small set of rules, which can be inspected and manually finetuned to adjust for idiosyncracies in the hand-tagged corpus. In
previous testing on Wall Street Journal articles, Brill has reported a success rate of over 97 percent, one of the highest in any of the automated tagging approaches (Brill 1994).

The large held-out part of the physical chemistry corpus was automatically tagged by initializing the tags without regard to context, and by subsequently applying each transformation in the contextual rule-set in turn. The overall POS ambiguity (ratio of word classes per word) for the training corpus is 1.30 without correction for occasional mistaggings.

Bracketing the Corpus

Another application of transformation-based error-driven learning is to augment a corpus with bracketing information, that is, to generate implicit trees with binary branching, but whose nonterminals are unlabelled. Figure 3.3, for example, shows the PUNDIT generated parse tree that corresponds to the following bracketed sentence: “((these measurements) revealed (spectral features (that (were (assigned (to (vibrational modes (of (the perturbed NH<4</d><u>+</u>ion)))))))))."

A training corpus is created by stripping off the brackets from a treebank with proper phrase structure and by naively initializing each sentence with a right-linear structure. As with the POS tagger, a set of structural transformation rules is then learned by repeatedly applying twelve possible transformations that add or delete parentheses to the training corpus and by keeping those rules that lead to the greatest overall reduction in parsing error compared to the treebank. But unlike the tagger, the triggering environment for the rules is based on the POS of the words, rather than the words themselves, and each rule application requires a few extra steps to preserve the binary branching and balanced parentheses condition. For the sample sentence above, the initial structure looks like the following: “((these (measurements (revealed (spectral (features (that (were (assigned (to (vibrational (modes (of (the (perturbed (NH4+ion)))))))))))))))).” The
Figure 3.3. Sample Parse Tree
result of applying one such rule can be seen in figure 3.4: add a “)” to the right of plural noun tags.

The scoring metric used to judge which transformation rule leads to the best improvement in conforming to the proper phrase structures is the cross-bracketing score described in Pereira and Schabes (1992) and Black, Lafferty, and Roukos (1992); it measures the percentage of constituents (words in between matching brackets) which do not cross any constituents in the target treebank. For example, in the following sentence pair, the two constituents marked by heavy parentheses are structurally inconsistent (see figure 3.5): “(( these measurements ) revealed ( spectral features ( that ( were ( assigned ( to ( vibrational modes ( of ( the perturbed NH4+ ) ion ) ) ) ) ) ) ) ) . )” vs. “(((( these measurements ) ( revealed ( ( spectral features ) ( that ( were ( assigned ( ( to ( vibrational modes ) ) ( of ( ( the ( ( perturbed NH4+ ) ion ) ) ) ) ) ) ) ) ) ) ) ) ) . )”

Brill used the same cross-bracketing score to evaluate the accuracy of his rule-based “transformational grammar” against a stochastically induced grammar that uses the inside-outside algorithm (Pereira and Schabes 1992), and reports comparable performance while needing a much smaller training corpus and far fewer rules. Another advantage over Context Free Grammars, traditional or probabilistic, is that the trained parser is somewhat

---

5This metric was the outcome of the PARSEVAL workshop on parser evaluations (Black et al. 1991). In order to facilitate the comparison of parser performances, the “Treebank Recognition Problem” was intentionally defined with a limited scope: rather than accepting any “reasonable” parse, the one and only correct analysis is the one in the treebank. Magerman argues however that the state-of-the-art in (statistical) parsing has improved in the last few years to a point where the cross-bracket error scores are no longer informative enough to differentiate between parsers; he proposes instead an even stricter evaluation criterion of “exact match,” where not only constituents, but also labels and tags must be identical to the treebank parse (1994, 94).

6150 sentences in a training corpus from the ATIS domain, and 64 transformations, vs. 700 sentences and 4096 context-free rules and probabilities (Brill 1993a, 262). Note that while the learner is weakly statistical in nature, the resulting parser is purely symbolic.
Figure 3.4. Example of Applying Bracketing Rule "b1_r nns 25703"
these measurements revealed spectral features that were assigned to vibrational modes of the perturbed NH₄⁺ ion.

Figure 3.5. CrossBracketing Measure
more robust because not all the input tokens are necessarily considered by the transformation rules (Brill 1993a, 259).

Preliminary code provided by Eric Brill was used to implement a transformation-based learner. Using 32,000 bracketed sentences from the Penn Treebank, 82 transformation rules were induced, which were subsequently used to bracket the corpus of physical chemistry. To assess the accuracy of the bracketing, the standard cross-bracketing precision formula was used (Magerman 1994, 92):

\[
\frac{\text{# of parse constituents which exactly match treebank constituents}}{\text{total # of parse constituents}}
\]  

(3.1)

The “treebank” standard in this case is a special held-out section of the chemistry corpus, consisting of 170 sentences that were input to the PUNDIT parser.\(^7\) Table 3.1 confirms the findings of Brill that performance is a factor of sentence length (in addition to size of the training corpus, and number of rules learned; Brill 1993a, 263): in one experiment, only sentences of fifteen words or fewer were selected from the test corpus. The last two rows show the results of comparing against all the sentences in the test corpus that received a parse.\(^8\) Figure 3.6 shows the distribution of sentence lengths in the physical chemistry corpus. The high count of very short sentences is due to headers and SGML placeholders for non-text entries like figures. Although it is generally true that longer sentences are more difficult to parse (and bracket) than shorter ones, headings pose a unique problem because

---

\(^7\)This testbed is used to evaluate the effects of customizing the lexicon on parser performance; see chapter IV for details. Run 1 in table 3.1 used the unaltered PUNDIT system (forty-two sentences received a parse); In run 2, the lexicon was updated minimally with the POS for each word, rather than relying on PUNDIT’s simplistic heuristic which guesses the POS of unknown words based on suffixes (sixty-seven sentences received a parse).

\(^8\)Not all of the sentences that received a parse in run one and two are included because a mismatch in length prevented direct comparison; for instance, Brill’s transformational learner saw multi-word expressions like “rather_than/in” as a single tag, whereas PUNDIT considered both words separately in the parse; and PUNDIT turned figures like “3.4” into multiple tokens (“qvar-q == 3^ ^4”).
Table 3.1. Bracketing Accuracy

<table>
<thead>
<tr>
<th>max sentence length</th>
<th>average sentence length</th>
<th>average # of constituents</th>
<th>% accuracy</th>
<th>% of sentences with ≤2 errors</th>
<th>% of sentences with ≤5 errors</th>
<th>% of sentences with ≤10 errors</th>
<th>% of sentences with ≤15 errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>run 1 &lt;15</td>
<td>9.62</td>
<td>6.21</td>
<td>0.70</td>
<td>0.50</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>run 2 &lt;15</td>
<td>10.86</td>
<td>7.21</td>
<td>0.65</td>
<td>0.29</td>
<td>0.93</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>run 1 unlimited</td>
<td>20.64</td>
<td>13.89</td>
<td>0.66</td>
<td>0.18</td>
<td>0.39</td>
<td>0.75</td>
<td>0.96</td>
</tr>
<tr>
<td>run 2 unlimited</td>
<td>23.15</td>
<td>15.57</td>
<td>0.65</td>
<td>0.11</td>
<td>0.39</td>
<td>0.70</td>
<td>0.91</td>
</tr>
</tbody>
</table>

Figure 3.6. The Distribution of Sentence Lengths in the Physical Chemistry Corpus
they are often elliptic sentences or standalone nounphrases. There is also a long tail of
sentences that are over one hundred words, the longest one being 271 words.

One obvious reason why Brill achieves higher accuracy scores is that both his
training corpus and his test corpus come from the same sublanguage domain, whereas the
experiments described in table 3.1 applied the transformational rules learned from the Penn
Treebank to a different corpus. Moreover, the accuracy scores are misleading because the
standard against which the bracketing program was measured was produced by PUNDIT,
a knowledge-based NLP system whose linguistic foundation may be quite different from
the model used to generate the Penn Treebank. The bracketing in the Penn Treebank was
created by Hindle’s Fidditch deterministic parser, and hand-corrected by human annotators
(Hindle 1983). Such differences have lead some advocates of non-stochastic NLP systems
to argue against the evaluation measure of the Treebank Recognition Problem as inappro-
priate (Magerman 1994, 89). For example, noun-noun modifiers are systematically handled
differently in PUNDIT and Fidditch: compare “nearly free internal rotation of the solvent
molecules about ( their ( local c3 ) ) axes ( in ( ( the first solvation ) shell ) ) has been
observed in the smaller clusters” with “nearly free internal rotation of the solvent molecules
about ( their ( local ( c3 axes ) ) ) in ( the ( first ( solvation shell ) ) ) has been observed in
the smaller clusters.”

However, the following scenario can be used to bootstrap a system, as has been done
with supervised training of stochastic POS taggers: induce a tranformational grammar for a
domain for which a bracketed treebank is available (for example the Wall Street Journal).
Apply these rules to bracket text in a new domain; the output is then corrected by a human
annotator,9 possibly guided by the trees of those sentences in the new domain that were
successfully parsed by an NLP system like PUNDIT or Fidditch. This newly created

9The Penn Treebank, release 2, is making tree manipulation tools like tgrep
available. Alternatively, “patches” can be manually added to the transformational rule set to
achieve the desired corrections.
“treebank” for the target domain can then be used to induce a sublanguage-specific transformational grammar, possibly taking a few iterations where the output, after correction, becomes additional input for the next training cycle.

Even if the transformational grammar was learned from a different corpus, it can be a useful tool in evaluating the output of the PUNDIT parser; PUNDIT does not return partial parses for input it cannot handle, but when a parse tree is generated there is no guarantee either that this is a correct or even plausible analysis. Occasionally, false positives are produced, as for example when the wrong POS is selected for an ambiguous word like “shift” and “bind” in figure 3.7. A high cross-bracketing error score (25 in figure 3.7) is a possible indicator of such errors. In a similar fashion cross-bracketing scores can be used to choose among the multiple parses PUNDIT can produce for a sentence; it is normally the responsibility of the semantic component to select among the unranked syntactically viable alternatives, but porting this module to a new domain is still very much a laborious manual task (see chapter IV for more detailed background on the PUNDIT system). In an experiment where PUNDIT was configured to produce the top ten parse alternatives for each sentence, 30 percent of the cases were found to have a parse with fewer cross-bracketing errors than the one offered as a first alternative. Figure 3.8 shows two examples; the first analysis of each pair is the one returned first by PUNDIT. Notice that both analyses in the first pair are actually false positives, due to the fact that a wrong or incomplete feature set is associated with “range” in the lexicon. This point will be made again in chapter IV.

---

10PUNDIT uses a very simplistic heuristic to guess the POS of new words, based solely on the morphological ending. See chapter four for more details.

11Not all of these, though are necessarily a “better” parse, in part because the transformational rules were not optimized for this domain or parser. It should also be noted that not all of the ten surface structures returned by PUNDIT are necessarily unique: the system also generates a deep structure which regularizes the surface parse before passing it on to the semantic component. The cross-bracketing metric, however, is based on the surface tree.
Figure 3.7. Sample False Positive
the discharge current under these conditions ranged from 40 to 200 mgr.

a shielding grid [...] prevented stray fields from affecting the ion trajectories in this region.

Figure 3.8. kbest Analysis
Augmenting the data corpus with phrase structure information not only helps in analyzing the trees produced by a knowledge-based NLP system, but can also be used to generate more informative context digests (see next section).

**Bootstrapping the Category Space**

The main philosophy behind corpus-based approaches to NLP is that the data source itself is a resource that can be mined for linguistic information. Frequency information can be gathered about individual words, from which conclusions can be drawn about their syntactic behavior (for that domain), which in turn can be used to aid in processing the text. McKee and Maloney (1992), for instance, distinguish between count and mass nouns based on the finding that a high ratio of singular to plural uses generally indicates a mass nature. However, statistics on words in isolation can be misleading; for example, words in non-final position in compound noun phrases are not inflected for number (e.g., “electron densities” or “transmission spectra”), and context should be used to filter out this “noise.” Additional evidence for assigning the “mass” feature would be a low mutual association ratio with preceding indefinite articles or numbers.

**Syntactically-based Schemes**

There are two general approaches in the literature to collecting distributional information: window-based and syntactically-based (Charniak 1993). In the latter scheme the text is scanned until a section is found that is deemed to be relevant. These methods differ in the amount of linguistic sophistication they presume; Brent (1992;1993), who attempts to model children’s language acquisition, relies solely on a list of prepositions and pronouns and a few simple heuristics: any word that occurs in the corpus both with and without the “-ing” suffix is taken to be a verb, unless it is ever found following a determiner or a preposition other than “to.” A phrase is assumed to be a complement of the verb only if it follows immediately or when the sole intermediary word is a pronoun or
proper noun. Brent achieves very high accuracy in learning subcategorization frames of verbs, but because his method is highly selective, a very large corpus is needed.

The majority of the syntactically-based schemes, however, do exploit POS tags assigned by a stochastic tagger, even if these are not perfect. McKee and Maloney (1992), for example, discover three kinds of clausal attachment complements by examining the tags following a verb. Because they consider only unambiguous cases, a lot of potential information remains unexamined, as in Brent’s case. To investigate the wider context, the “rough” structure of the sentence must be computed, a process known as partial parsing. This produces a flat tree with phrase boundaries marked and identified by type, but without much internal detail. Manning (1993), for example, wrote a small finite state parser to improve on Brent’s design: it parses the complements following a verb until the end of the subcategorization arguments is discovered. When the parser encounters input it cannot handle, it simply skips to the next sentence, but even so Manning recognizes that several false positives are introduced; to separate the valid feature frames from the miscues statistical filtering is used which weighs the number of times a verb co-occurs in the corpus with a particular cue against the total frequency of the verb.

Instead of building an ad-hoc system for partial parsing, some researchers have turned to full-blown robust NLP systems, which return detailed analyses for that part of the input they can handle, and a more summary partial parse for the “extra-grammatical”

---

12 Brent only considers proper names and pronouns, which are easily detected noun phrases, because noun phrases in general are hard to isolate. However, Weischedel et al. (1993) report on two successful experiments with two different programs that were able to identify core noun phrases (sans post-modifiers) with 90 percent success using local syntactic information only.

13 ”that” complement: _/vb? that/sc infinitive complement: _/vb? to/to _/vb gerund complement: _/vb? _/ger

14 Flagged by a period, subordinate conjunction, or clausal complement. The parser contains a noun phrase recognizer and several other rules which seem to have been handcrafted to fit the specific domain.
portion. Hindle and Rooth (1993), for example use the Fidditch parser to identify ambiguous verb-noun-preposition triples to induce preposition attachment frames for the lexicon. Because the partially parsed output is an imperfect source, like Manning they use statistical filtering to estimate the lexical associations.

Window-based Schemes

A second approach to collecting relevant distributional information is to keep co-occurrence counts of the nearest lexical neighbors of a word, usually within a fixed distance or “window.” Markov models, for example, predict the POS of a word based on the tags of the two or three words preceding it (bigrams and trigrams respectively). A window can also include neighbors to the right: McKee and Maloney (1992) used a fixed two-word window to the right to collect information for preposition attachment decisions for verbs. Schütze has experimented with window lengths of four words (Schütze 1993b), two hundred letter fourgrams (Schütze 1993a) and two thousand characters (Schütze 1992).15

For this dissertation, a window of four was adopted, i.e., for words of interest in the domain of physical chemistry, co-occurrence counts were kept between those words and their immediate left neighbors ($w_{i-1}w_i$), immediate right neighbors ($w_iw_{i+1}$), and left and right neighbors that are two words away ($w_{i-2}w_i$ and $w_iw_{i+2}$ respectively). Or in other words, for selected word pairs ($v w$), separate counts were kept of how many times $v$ occurs in the training corpus at most two words away from $w$ in either direction. As table 3.2 shows, separate runs were done for adjectives, nouns, and verbs. For example, 655 word types tagged with GER occurred frequently enough in the training corpus to warrant

15The vectors thus created were input to a neural network for use in POS assignment and word sense disambiguation.
constructing a vector or context digest. As Zipf’s law would predict, there is a long tail of word types which occur too infrequently to permit gathering useful statistics; (this seems especially true for nouns; as table 3.2 shows, almost 96 percent are disqualified: only for 2,585 out of 64,486 noun types that occurred in the training corpus was a context digest constructed. This despite the fact that chemical names like “ammonium” and “NH<sub>4</sub>” which have been tagged with /NN//SUBST, have been collapsed in the class “substance”).

Table 3.2. Size of Context Digests

<table>
<thead>
<tr>
<th></th>
<th># of context digests (f ≥ 10)</th>
<th>total # of types</th>
<th># of neighbor types (f ≥ 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JJ</td>
<td>2,090</td>
<td>12,239</td>
<td>6,036</td>
</tr>
<tr>
<td>NN</td>
<td>2,585</td>
<td>64,486</td>
<td>5,366</td>
</tr>
<tr>
<td>GER</td>
<td>655</td>
<td>1,861</td>
<td>5,165</td>
</tr>
<tr>
<td>VB?</td>
<td>1,430</td>
<td>5,791</td>
<td>5,509</td>
</tr>
</tbody>
</table>

Each window of the context digests tracks co-occurrence counts with word types of any POS, provided these types have a minimum frequency of one hundred in the training corpus (last column of table 3.2). For “rare” neighbors, the algorithm simply records the neighbor’s POS, a compromise to keep the size of the arrays manageable, while providing some information on the syntactic context. Additionally, some tokens like ordinal numbers have been collapsed into a single class “CD.” Another approach for handling low frequency words would be to perform morphological reduction to their root form (cf. Brown, Della Pietra, Della Pietra, Lafferty, and Mercer 1992).
be verbs (e.g., “shielding”), frequent word types of another POS (e.g., “expansion”), or tags for less frequent types (e.g., “CD” and “JJ”).

Figure 3.9 further illustrates how a context digest is formed by combining matrices of co-occurrence counts. Entries are shown for the words with verb tags in the following sample sentence from the training corpus: “((((a/dt shielding/vbg) grid/nn ((surrounding/vbg ((the/dt expansion/nn) region/nn)) (maintained/vbn (at/in (350/cd v/nn)))))) prevented /vbd (stray/jj fields/nns) (from/in (affecting/ger ((the/dt ion/nn) trajectories/nns) (in/in (this/dt region/nn)))) ./.)” Windows never cross sentence boundaries. This positional information is also tallied for the digests.

The first four matrices correspond to the fixed windows. The last two windows introduce some long(er)-distance information to the vector: based on the bracketing information with which the corpus was preprocessed, the phrasal boundaries within which the verb occurs are marked (indicated by a double line in figure 3.9). These boundaries are of variable length, and can in fact span the whole sentence, as is the case for “prevented.” Ideally, the name of the type phrase that the verb occurred in should be used as a clustering feature, but since this information is unavailable (the non-terminals in the trees implicit in the bracketing are unlabelled) the next best thing is used, and each boundary is marked by a pair of tags occurring on either side of the bracket. For example, in figure 3.9, “maintained” is bounded by NN-VBN on the left and NN-VBD on the right.

Each context digest for verbs, then, contains 27,654 possible entries. The resulting matrix is very sparse, however; the density for the verb category space is only 1.5 percent. To save space, the sparse matrix is saved in a compressed column storage format called Harwell-Boeing: a first vector stores the nonzero elements only, in column

\[18\]
Density is the ratio of number of nonzero elements over total number of slots in the matrix. The figure for verbs seems typical: the category space of adjectives has a density of 1.2 percent; noun density is somewhat higher (2.1 percent), gerund density lower (0.7 percent).
a shielding grid surrounding the expansion region maintained at CD prevented JJ fields from affecting the ion trajectories in this region.

Figure 3.9. Sample Context Digest
major order; a second vector of equal length holds the index of each value in its column. A third and final vector indicates the location of the first value in each column. See figure 3.10 for an example, taken from Duff, Grimes, and Lewis (1989). This is the format expected by the SVDPACKC program for dimensionality reduction, which will be explained next.

\[
\begin{bmatrix}
1 & -3 & 0 & -1 & 0 \\
0 & 0 & -2 & 0 & 3 \\
2 & 0 & 0 & 0 & 0 \\
0 & 4 & 0 & -4 & 0 \\
5 & 0 & -5 & 0 & 6
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th>values</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>-3</th>
<th>4</th>
<th>-2</th>
<th>-5</th>
<th>-1</th>
<th>-4</th>
<th>3</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>row index</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>column pointer</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.10. Harwell-Boeing Format for Sparse Matrices

**Singular Value Decomposition**

After bootstrapping the category space based on co-occurrence counts in the training corpus, the distributional information is generalized by means of a matrix manipulation method called Singular Value Decomposition (SVD). This technique is often used in factor analysis, because reducing the representation to a low dimensionality allows one to better visualize the space. It is exactly this compactness of representation that has led Schütze to apply SVD to the field of NLP, to reduce the number of input parameters to a neural net, without sacrificing too many of the fine distinctions in the original text (Schütze 1992; 1993a; 1993b). Schütze stresses, though, that the objective of dimensionality reduction is not visualizing the underlying factors, or “rotating the space in order to position the axes in an intuitive way” (1992, 9). This echoes the comments of Deerweester et al., who pioneered the use of SVD in information retrieval, that their goal is “not to be able to
describe the vectors verbally, but merely to be able to represent terms, documents and queries in a way that escapes the unreliability, ambiguity and redundancy of individual terms as descriptors” (Deerweester et al. 1990, 395). This improved document representation follows from the dimensionality reduction, in that “documents with somewhat different profiles of term usage” can now be “mapped into the same vector of factor values” (1990, 395). The next three sections will briefly elaborate on word-by-document matrices, word-by-word matrices, and on the formal characteristics of the SVD model.

SVD for Improved Document Representations

The goal of information retrieval is to locate documents that match a user’s query. Relevance decisions are traditionally made on the basis of lexical matches between the terms in the query and document base, but because of variability in word usage this approach often leads to poor recall and precision: the presence of synonymy in natural language means that an author may have described his or her ideas using different words than the ones used in the query; conversely, the presence of polysemy means that a match may be a false positive because a word may refer to several things. The traditional solution to these problems has been to require the use of restricted vocabularies in indexing and querying, and to augment the query transparently with terms from a domain-specific thesaurus.

Deerweester et al. introduced a novel way of looking at the retrieval question as a statistical problem, on account of the unreliability of the observed association between terms in the query and the documents: their assumption is that the “underlying latent . . . structure in the data . . . is partially obscured by the randomness in word choice” and that statistical techniques can “estimate this latent structure and get rid of the obscuring ‘noise’” (1990, 391). The statistical technique in question is SVD, which decomposes the original term-document matrix into linearly independent factors, many of which are very small. An
approximate model with fewer dimensions can be constructed by ignoring these small components. Figure 3.11 shows an example of a hypothetical term-document matrix, taken from Deerweester et al. (1990). The fifth row in C shows, for example, that that term ("system") occurs once in the second and third document, and twice in the fourth document. SVD breaks down this matrix into three matrices $T_0, S_0$ and $D'_0$ such that the original matrix $C$ can be reconstructed (with occasional rounding errors) by the product of $T_0 \cdot S_0 \cdot D'_0$. But by combining only the first $k$ linearly independent components (two in this example) a reduced model is built which disregards lesser terminology variations, because $k$ is smaller than the number of rows (terms). Consequently, "the similarity of terms and documents is determined by the overall pattern of word usage in the entire collection," rather than by the exact surface-level words (Dumais 1991, 230).

The choice of $k$ determines the amount of dimensionality reduction and hence the degree to which the reduced matrix $\hat{C}$ fits the original matrix, or in other words, the degree to which generalizations are captured. Both Deerweester et al. and Dumais found that a relatively small number of dimensions worked best;\textsuperscript{19} they settled on $k=100$, and found that this allowed on average a 30 percent performance improvement over the word-matching method in discriminating relevant documents.

**SVD for Compact Representations**

Inspired by the success of SVD in information retrieval, Schütze introduced this technique to the field of NLP. There are two main differences between his approach and the one mentioned above: the first is that Schütze’s main concern is to exploit the

\textsuperscript{19}Relatively small compared to the number of terms or rows, that is. Dumais experimented with increasing sizes of $k$, and found that performance sharply increased around ten dimensions, and slowly degrades after a peak somewhere between seventy and one hundred dimensions. She concludes that performance increases “only while the added dimensions continue to account for meaningful, as opposed to chance, co-occurrence” (1991, 232).
\[
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

**Original matrix \( C \)**

\[
\begin{bmatrix}
0.22 & 0.11 & -0.29 & -0.41 & 0.11 & -0.34 & -0.52 & -0.06 & -0.41 \\
0.20 & 0.07 & -0.14 & -0.55 & -0.28 & 0.50 & 0.07 & -0.01 & -0.11 \\
0.24 & -0.04 & 0.16 & -0.59 & 0.11 & -0.25 & 0.30 & 0.06 & 0.49 \\
0.40 & -0.06 & 0.34 & 0.10 & -0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\
0.64 & -0.17 & -0.36 & 0.33 & 0.16 & -0.21 & 0.17 & 0.03 & 0.27 \\
0.27 & -0.11 & 0.43 & 0.07 & -0.08 & -0.17 & -0.28 & -0.02 & -0.05 \\
0.27 & -0.11 & 0.43 & 0.07 & -0.08 & -0.17 & -0.28 & -0.02 & -0.05 \\
0.30 & -0.14 & -0.33 & 0.19 & -0.11 & 0.27 & -0.03 & -0.02 & -0.17 \\
0.21 & -0.27 & 0.18 & -0.03 & 0.54 & 0.08 & 0.47 & -0.04 & -0.58 \\
0.01 & -0.49 & -0.23 & 0.02 & -0.59 & -0.39 & 0.29 & 0.25 & -0.23 \\
0.04 & -0.62 & -0.22 & 0.00 & 0.07 & 0.11 & -0.16 & -0.68 & 0.23 \\
0.03 & -0.45 & -0.14 & -0.01 & 0.30 & 0.28 & -0.34 & 0.68 & 0.18 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.34 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 2.54 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 2.35 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 1.50 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
\end{bmatrix}
\]

\( T_0 \)

\( S_0 \)

\( D_0' \)

\( ^\text{Reduced matrix} \)

(only the first two singular values are kept, plus the corresponding columns from \( T \) and \( D \))

**Figure 3.11. SVD Decomposition of Sample Matrix for \( k=2 \)**
dimensionality reduction to obtain a compact representation of the co-occurrence data. Consequently he experimented with much more aggressive values for $k$: 15 and sometimes even as low as 10 (Schütze 1993b). Some of the trials involved character-based and letter fourgram windows, which resulted in rather dense matrices, so no smoothing “to deal with the ‘bumpiness’ of small counts” was really necessary (Schütze 1992, 9).

A second difference is that Schütze uses word-by-word rather than word-by-document matrices. In the latter case recall and precision is improved because synonyms in the reduced document vectors are collapsed onto the same underlying dimension; in the case of word-by-word matrices the synonym vectors are already similar because their context vectors are alike (they tend to occur with the same words).

**Formal Description of SVD**

Given a $m$-by-$n$ matrix $C$, there exist two unitary matrices $U$ and $V$ such that

$$S = U' C V$$

(3.2)

where $U'U=V'V=VV'=I_n$ and $S=\text{diag}(\sigma_1,\ldots,\sigma_n)$ (Golub and Reinsch 1971).

$U$ contains $n$ orthonormalized eigenvectors that are associated with the $n$ largest eigenvalues of $CC'$, and $V$ holds the orthonormalized vectors of $C'C$. The diagonal elements of $S$ are the singular (or principal) values of $C$; these are the non-negative square roots of the eigenvalues of $C'C$ (or $CC'$, whichever has fewer rows and columns). The following properties also hold, for $r = \text{rank}(C)$:

$$\sigma_i > 0 \text{ for } 1 \leq i \leq r$$

(3.3)

and

$$\sigma_i = 0 \text{ for } i \geq r+1$$

(3.4)

Berry et al. state the theorem of (3.2) slightly differently to make clear the decomposition into major axes of variation:

$$C = U \ \text{diag}(\sigma_1,\ldots,\sigma_p) \ V' \ \text{ where } p=\text{min}\{n,m\}$$

(3.5)
The dyadic decomposition property then allows the description of a matrix as the sum of $r$ rank-one matrices of decreasing importance (as measured by the singular values). The important generalization property of SVD follows from the following theorem: (Berry et al. 1993, 3)

given

$$\mathbf{C} = \mathbf{C}_k = \sum_{i=1}^{k} \mathbf{u}_i \sigma_i \mathbf{v}_i$$

with $k < r$,

then

$$\min_{r(B)=k} \| \mathbf{C} - \mathbf{B} \|_F^2 = \| \mathbf{C} - \mathbf{C}_k \|_F^2 = \sigma_k^2 + 2 + \sigma_p^2 .$$

(3.6)

$\mathbf{C}_k$, being the product of $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ restricted to their first $k$ columns, is the best least-square approximation of rank $k$ to the original matrix $\mathbf{C}$. Or, in other words, SVD selects the most cautious projection by choosing among all possible matrices in a $k$-dimensional space the one that is least different from the original co-occurrence matrix, thus preserving much of the original similarity structure while mapping rare context digests in the directions to which they are most similar.

**SVD in the Physical Chemistry Domain**

To generalize the associational patterns in the category space that was bootstrapped from the physical chemistry corpus, SVD was applied with a conservative value for $k$ of 350. The tool used for this purpose was a slightly modified version of the las2 module from the SVDPACKC package (Berry et al. 1993). The las2 module is an implementation of the single-vector Lanczos algorithm, with selective reorthogonalization (Golub and Underwood 1977). This iteration-based method is about five times faster than any of the other seven modules in determining the largest singular triplets (singular values and corresponding left and right singular vectors). Even so, it took typically between five and
seven days for the program to make a single run on a 50 MHz SPARCstation-10, with 250 MB of RAM and one GB of swap space.\textsuperscript{20}

The generalizing effect of SVD causes the category space for verbs to become much less sparse: 35.4 percent of the entries now have non-zero “counts,” up from 1.5 percent. Most of these are new counts, i.e. SVD infers context similarities between words that may not be apparent in the original co-occurrence matrix due to the natural randomness in any corpus sample. For example, “established” and “calculated” are syntactically similar verbs that share several features like SNWH and THATS.\textsuperscript{21} However, in the original category space their context digests are dissimilar because the arguments used in the training sample are somewhat different. For example, “calculated” tends to occur with an adverb before the embedded sentence: “it was calculated \textit{previously} that the rate constant for tunneling reaction CD is about CD times larger than that for tunneling reaction CD at CD k” or “by using figures CD and CD , it is calculated \textit{easily} that m* = 0.28me even if v0 = CD ev.” Consequently the pair of verbs originally has a low (0.51) cosine similarity measure (cosine similarity will be explained in the last section of this chapter). The training corpus contains other verbs, though, like “observed” and “evaluated” which share agents with both “established” and “calculated.” The dimensionality reduction of SVD uses this fact to infer the similarity of both “established” and “calculated” (their new cosine score is 0.93).

Although it is clear that the generalizing effect of SVD makes the category space less sparse, the non-zero values in the original and reduced matrix cannot be compared directly;

\textsuperscript{20}The time complexity of SVD is quadratic in the rank of the collocation matrix (Schütze 1993b, 252).

\textsuperscript{21}See appendix A for a brief explanation of the subset of the Linguistic String Project features used in this dissertation. Following are two sample sentences from the training domain that illustrate the SNWH and THATS feature: “however , it has not been established \textit{whether} the t1 state can also be generated by intersystem crossing .” “we have established \textit{that} spin-rotation dominates both CD and CD relaxation in ch4 in the gas phase .”
in the former they are integer frequency counts, in the latter real numbers, sometimes negative (see for example figure 3.11). A more informative view of the impact of SVD on the clusters can be seen in figure 3.12, which plots the average number of verbs in a cluster for a given cosine similarity threshold. The average number of context digests that are very similar (greater than 97 percent confidence) remains fairly constant after SVD, but the dimension reduction provides a lot more information about syntactic behavior when a less strict cutoff value is adopted (say 90 percent). This is the now familiar tradeoff between using limited but highly accurate data and examining a broader array of “noisier” information, as was the case in syntactically-based approaches to bootstrapping a category space (Brent 1992 and Manning 1993).

![Figure 3.12. Impact of SVD on Cluster Size](image-url)
Cosine Similarity

Once the category space has been bootstrapped and generalized, the syntactic similarity of any two words can be computed by using the cosine similarity measure, which was a major feature of the SMART information retrieval system (Salton 1983). This measures the cosine of the angle between two context digests, which can be viewed as vectors in a $s$-dimensional space. Each axis in this space is a high-frequency word type, disambiguated by POS and window position, or a tag (pair) for low frequency words and phrase boundaries. The position of a context digest in this space is determined by the weight of the types in that vector. When two digests are identical (they both occur with the same neighbors a proportional number of times and there are no neighbors that appear in only one of the digests), their vectors are superimposed in the space, and the angle between them is zero.\(^{22}\) Hence the similarity measure is an inverse function of the angle between two vectors.

Table 3.3 shows three partial context digests for verbs; for demonstration purposes, only three of the 27,624 dimensions are shown, all from the immediate right-neighbor window. The string “calculated from” occurs 1,052 times in the training corpus, and “established” co-occurs 127 times with “that” (for both verbs this particular neighbor happens to be the one they occur most often with). “Verified” is much less frequent in the

\(^{22}\)Such identical vectors would be extremely rare for a large corpus, and in fact there are none in the training corpus of physical chemistry for verbs, adjectives, or nouns. However, in one experiment the corpus was checked for duplicate tag sequences that span an entire sentence but differ in at least one word token. For example, there are three instances of “/dt /jj /nn /aux /vbn /in /dt /nn /in /dt /nn”: “a similar lens was used for the collection of the fluorescence,” “the main problem is related to the nature of the tls” and “a similar mechanism is considered in the field of the electrochemistry.” There are 892 such tag sequences in the corpus with an average length of 10.29 tags. Many of these instances seem to be motivated by the stylistic need for variation in writing, as in “these are also given|listed|plotted in table CD,” but they do reveal syntactic as well as semantic information that can be tapped. For example, any features currently in the lexical entry of “purify” that govern its relationship with “by” may have to be extended to other prepositions upon encountering multiple instances like “SUBST was purified by|with a millipore super-q system.”
corpus, and is never followed by “from.” Figure 3.13 plots those three vectors in the three-dimensional category space.

Table 3.3. Partial Context Digests

<table>
<thead>
<tr>
<th></th>
<th>from/in &lt;1</th>
<th>that/sc &lt;1</th>
<th>./ &lt;1</th>
</tr>
</thead>
<tbody>
<tr>
<td>calculated/vbn</td>
<td>1052</td>
<td>25</td>
<td>70</td>
</tr>
<tr>
<td>established/vbn</td>
<td>13</td>
<td>127</td>
<td>47</td>
</tr>
<tr>
<td>verified/vbn</td>
<td>0</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

![Figure 3.13. Proximity of Sample Context Digests in 3-d Space](image)

The cosine similarity measure is defined as

$$\text{cosine}(d_i, d_j) = \frac{\sum_{c=1}^{s} (d_{ic} \cdot d_{jc})}{\sqrt{\sum_{c=1}^{s} (d_{ic})^2 \cdot \sum_{c=1}^{s} (d_{jc})^2}}$$

(3.7)

The numerator is the sum of the products of the counts that both vectors have in common. The denominator normalizes the equation so all vectors are of equal length, by dividing the expression by the products of the lengths of the digests. For the limited data of table 3.3,
the cosine similarity between “established-calculated” is 0.14 and between “established-verified” 0.97. Note that it is the distribution of the neighbors which affects the similarity measure, not the absolute frequency of each verb; if “verified” were to occur twice as often, with counts of 0-30-4, the similarity with “established” would remain the same.

The category space can be clustered by comparing pairs of context digests using the cosine similarity measure; such clusters contain words whose syntactic behavior is substantially similar. The degree of similarity depends on the adopted threshold value. For example, a high threshold of 0.95 for the verb category space results in an average cluster size of 6.87, with 1,170 verbs not belonging to any cluster. A more liberal value of 0.80 broadens the clusters to an average of 88.75 verbs, with 963 singletons (see figure 3.12).

However, these neighborhoods are not traditional clusters; each verb has its own individual representation in a multi-dimensional space, i.e. is the center of its own neighborhood. Typically any given verb is a vector which simultaneously belongs in several neighborhoods. Schütze emphasizes the benefits of this approach over arbitrary classification:

Any clustering into classes introduces artificial boundaries that cut off words from part of their . . . neighborhood. In large classes, there will be members “from opposite sides of the class” that are only distantly related. So any class size is problematic, since words are either separated from close neighbors or lumped together with distant terms (1993a, 896-97).

Salton points out this same advantage of individual spaces, from the perspective of information retrieval; whereas conventional library classification systems place related documents together in a single common class, the overlapping classes in the “clustered document file . . . represent a substantial advantage” by permitting “documents [to] appear in more than one class” (1983, 124).
Conclusion

This chapter has shown how a category space can be bootstrapped from co-occurrence counts in a training corpus. A fixed-window approach has been augmented with phrasal boundary information, and has been finetuned by POS disambiguation of the input tokens. A smoothing technique called SVD has been used to generalize the distributional information. Proximity in this reduced space is then used to find for all the context digests a “cluster” of verbs that are paradigmatically related. Chapter IV explains how this information can be used to infer syntactic knowledge in the form of lexicon features.
CHAPTER IV
SUBCATEGORIZATION FRAMES

Two issues dealing with porting a parser to a new domain will be addressed in this chapter, and a solution will be proposed which employs the category space that was derived in chapter III. The first issue involves dealing with new words, i.e., input outside the current knowledgebase. The second, related issue has to do with adjusting the existing grammar knowledge to fit the “idiosyncracies” of the new domain.

One way of testing a possible solution and of evaluating its effectiveness is an “existence proof,” which involves comparing the output of a broad-coverage NLP system on some held-out text from a new domain before and after the parser was modified with “induced” information. One broad-coverage NLP system is Unisys’ PUNDIT, which was chosen because of its long-time presence in the research literature, because it was designed to be portable and broad-coverage, and because of its availability in source form for academic research. Much of the grammatical knowledge of PUNDIT is stored as subcategorization frames in the lexicon, in particular for verbs, and specifying this information is a crucial but “daunting . . . task,” as one of PUNDIT’s developers puts it:

PUNDIT requires very detailed information about what types of complement a verb can take, and what prepositions and particles the verb requires. This information is necessary to get correct parses and avoid incorrect parses, but it is difficult to specify. Many of the distinctions amongst complement types may be obscure to the non-linguist, but they are all significant. (Riley 1988, 9)

A good example of the above can be seen in figure 3.8. “Range” is classified in PUNDIT’s standard lexicon as a noun, but because of the “-ed” suffix the system correctly identifies
the use in this sentence as a verb. The only feature which PUNDIT hypothesizes for new verbs is transitivity, which not only is incorrect in this instance, but the lack of the subcategorization for a prepositional phrase complement yields a false positive, with an empty verb slot. Figure 4.1 shows that after the lexical entry for “range” has been updated to indicate a strong selectional dependency with the prepositions “from” and “to,” PUNDIT’s native grammar is comprehensive enough to correctly parse the sentence.

The outline of this chapter is as follows: the first section provides some background information on the PUNDIT system. A brief analysis follows on the performance of the unmodified parser on a testbed from the physical chemistry domain. The next section illustrates some easy gains that can be had by including stochastic information instead of relying on the set of simple heuristic rules PUNDIT uses for POS disambiguation. Then the PUNDIT system’s architecture will be revisited for a discussion of its lexicon and of the subcategorization frames its designers adopted from Sager’s Linguistic String Project (LSP) grammar. Finally, there will be a discussion of experiments in inducing two sets of frames from the category space: a subset of features that encompasses those found in the studies by Brent (1993), Manning (1993), and McKee and Maloney (1992), and a larger set consisting of those complements from the String Grammar which the PUNDIT developers chose to implement in their system (described in Linebarger 1988).

1PUNDIT’s designers have made it a major goal to keep the domain dependent and the domain specific components of their system separate, for instance by maintaining a distinct core lexicon for generic English and a secondary lexicon that is particular to an application domain. However, there are traces in the core lexicon of entries that are biased towards the ATIS (Air Travel Planning Information) domain, such as the nominal sense of “range” and “pass,” the verbal use of “bank,” and “head,” and the use of “return” as an adjective.
Figure 4.1. Correct Parse After Updating the Lexical Entry of “range”
PUNDIT’s Architecture

A Logic Restriction Grammar Implementation of Sager’s LSP

PUNDIT (Prolog UNDerstanding of Integrated Text) is a broad-coverage system for the syntactic, semantic, and pragmatic analysis of natural language text. Its design was intentionally modular to facilitate porting to different domains, and considerable effort was made to separate domain-dependent from domain-independent knowledge, and to keep procedural and declarative components apart. Figure 4.2, taken from Hirschman, Palmer et al. (1989) shows the architecture of the system; the domain-independent part of the system consists of about 65,000 lines of Prolog code, which specify the procedural component for the three phases of analysis, and most of the grammar plus the core lexicon. The domain-specific portion consists mainly of data, except for the application module; it contains additional lexical items, specialized grammar rules, and the knowledgebase and domain-dependent semantic and focusing rules for semantic and pragmatic analysis.2 PUNDIT has been ported to several domains, including military, medical, and air travel planning applications. The latter consists of about 45,000 lines of additional source code.

The grammar component produces both a detailed surface parse and a deep structure analysis, called Intermediate Syntactic Representation (ISR). The latter reduces surface variants to a single canonical form by regularizing the relationship between predicates and arguments, with the added advantage that details that are irrelevant for semantic analysis are removed. For example, since PUNDIT builds surface parse trees first, before “undoing” transformations like passivization, the lexicon must explicitly store subcategorization features for objects both in active and passive use. Because there is a predictable correlation

---

2For more information on the semantic component, see Palmer et al. (1986) and Dahl, Palmer, and Passonneau (1987). The pragmatics module handles reference resolution (Dahl and Ball 1990), discourse integration (Ball 1989 and Norton et al. 1990), and temporal analysis (Passonneau 1988).
Figure 4.2. PUNDIT Architecture (Hirschman, Palmer et al. 1989)
between such active and passive objects, tools are available for the developer which automatically derive the passive object options from the active ones (see Linebarger 1988).

A typical BNF grammar rule consists of three components, as is evident in the following sample rule (adapted from the developer’s guide, Paramax System Corporation 1992, 3):

\[ pn ::= lp,*p,nstg, \{w_pval\}, \rightarrow pp[p,nstg,lp] \].

1) The first component is a context-free production, based on Sager’s Linguistic String Project (the current version has about 100 such rules). The example above can be paraphrased as a prepositional phrase being rewritten as an optional preposition-modifying phrase, followed by a preposition, then a noun phrase.

2) The second component is a context sensitive restriction (currently there are about 75 such restrictions). PUNDIT’s grammar belongs to the class of restriction grammars. Unlike most logical grammars, which provide contextual information through parameters to the BNF definition, a restriction grammar isolates all context sensitivity in the constraints so the rewriting rules remain uncluttered. The restrictions in curly brackets are calls to Prolog predicates, which have access to two implicit parameters, the word stream and the parse tree constructed so far. The restrictions perform two functions, depending on their position in the production; in end-position, they enforce the well-formedness of the partial tree (e.g., subject-verb agreement); in initial position, the restrictions enforce optimization strategies by dynamically pruning the inapplicable object options to focus the search. This is especially true for verbs, which have a rich set of subcategorization features; for example, in the sample BNF rule above, if the prepositional phrase being built is the object of a verb, the “w_pval” restriction will

\[ \text{\textsuperscript{3}}\text{All logic grammars have in common the concept of parsing as proving sentencehood, where the parse tree is a sort of explicit proof tree. For a detailed discussion of logic grammars, and restriction grammars in particular, and the implementation of the latter in the PUNDIT system, see Hirschman and Dowding (1990).} \]
check the lexicon to see if that verb is subcategorized for the preposition in question. If it isn’t, the rule is eliminated from consideration. Such on-the-fly pruning yields a 8-9 fold speed-up, in particular when all parses are requested (Dowding and Hirschman 1987, 7). 4

3) The last part of the production is a lambda-calculus based rule for building the deep structure called Intermediate Syntactic Representation (see Lang and Hirschman 1988). PUNDIT uses a meta-grammar approach to achieve wide coverage of linguistic phenomena like conjunction and wh-constructs. This meta grammar transparently creates new rules from definitions of unconjoined elements and augments production rules with parameters that track the need for or the filling of a gap. 5 The advantage of this approach, as is the case with restrictions, is that the original BNF grammar remains uncluttered and compact to facilitate development and maintenance, while parsing efficiency is promoted by translating the grammar so the “expansion” takes place at compile time.

This section has described the modular nature of the PUNDIT NLP system. Of interest to this dissertation are its lexical and grammatical analysis components, which combine to exhaust the search space to produce all parsing variants that satisfy the syntactic requirements. Both surface structures and deep structures are passed to the domain-dependent semantic and pragmatic modules, which among other things must choose the

---

4The issue of efficiency becomes very important as the size of the grammar keeps growing, because of the design decision to add extensions to the grammar to increase the coverage when porting, rather than rebuilding a minimal covering grammar (see Hirschman, Lang et al. 1989). Much of the execution time is spent building partial trees that allow easy traversal. By compiling the grammar rather than interpreting it, the designers of PUNDIT achieved a 2-3 fold speed-up. The run-time pruning of rules is accomplished by adding code which calls a dynamic translator once the object options are known.

5Below is an example of a conjunction meta rule (taken from Hirschman and Dowding 1990, 152):

\[
\text{lnr}::=\text{ln},\text{nvar},\text{rn} \Rightarrow (\text{lnr} ::= \text{ln},\text{nvar},\text{rn}; \text{ln},\text{nvar},\text{rn},\text{conj_wd},\text{lnr})
\]

For more information on the conjunction meta-rule mechanism, see Hirschman (1986b); for wh-constructions, consult Hirschman (1988).
parse alternative that has the correct prepositional phrase attachment. The next segment
describes how well the syntactic components of the unmodified PUNDIT system fare when
parsing text from a new domain, to set a baseline against which later improvements can be
measured.

Trial 1: Core Lexicon Only

A section of text from the physical chemistry domain, consisting of 170 sentences,
was held-out for testing purposes. This segment has a total of 4,581 word tokens, or an
average sentence length of 26.94 words; the shortest sentence contains 5 words, the
longest 94. The number of unique word types is 1102, of which 172 are verbs other than
auxiliaries. Word types traditionally correspond to the number of distinct words, whereas
tokens indicate the number of running words (cf. Kucera and Francis 1967, 294). Thus in
the extract from the training corpus below, there are 64 tokens but fewer types, because for
instance the determiner “the” occurs multiple times (i.e., four tokens but one type).

Usually, the model for the process assumes that the solvent forms a dielectric
continuum surrounding the reactant and product species as well as any transition-
state structures. . . . On high-temperature annealing, trimers or perhaps other forms
of dimer give rise to peaks at 1.1 and 1.3 cm-1. . . . If numerous small fragments
are formed, scattering can be high and much ion current can be lost.

However, in this dissertation, the term “type” deviates from the standard definition in two
ways; because the corpus has been disambiguated by POS by a stochastic tagger, a word
with multiple parts of speech is considered multiple types. In the extract above, there are
two types for “forms” because it is used once as a verb and once as a noun. Secondly,
morphological variants like “formed,” “forming,” etc. are not considered distinct types.

---

6The longest sentence is actually an enumerated list in disguise, and should perhaps
be broken up into fragments for parsing purposes: “Under this notation, x takes_on the
values 0, 1, 2, ... and indicates whether the subgroup is in the core, first solvation shell,
second solvation shell, etc., y is the number of equivalent n-h oscillators of a particular
bond type in the local subunit, z is either bd or fr to indicate whether the h of the n-h bond
is involved in hydrogen bonding, and p is sstr, astr, or bend depending_upon whether the
vibration can be considered as a symmetric or an antisymmetric stretch or a bending mode
in the local subunit.” The second longest sentence is only 54 words.
The domain-independent lexicon distributed with the PUNDIT system contains 823 root forms (again, disambiguated by POS). This core lexicon holds definitions for very common, “generic” English words, including function words such as auxiliaries, determiners, quantifiers, etc. Roughly 30 percent are tokens that are not nouns, adjectives or verbs. Of interest for this dissertation are the 164 verbs that have detailed subcategorization information, several of which properly belong in a domain-specific lexicon (travel, depart, descend, schedule, . . .). Not unexpectedly, only 17 of the 172 verb types in the test corpus are defined in the lexicon (table 4.1).

PUNDIT has two options for dealing with unknown words, depending on the setting of the switch “enter_new_word”; if “on,” the user will be queried by the system to enter the necessary lexical information interactively, clearly an impractical task except for limited input files. The alternative approach, used in this dissertation, is to let PUNDIT

---

To make the process somewhat more tolerable, a tool is provided which speeds up the lexical entry by “elicite[ing] relevant linguistic information from the user, comput[ing] dependencies between attributes” (e.g., passive objects), “and prompt[ing] for morphological related forms” (Riley 1988, 1). Wherever possible it offers a guess, and sample sentences are available to illustrate the meaning of a feature. Besides speed, this utility promotes consistency and completeness.
guess the part of speech of unknown words, based on some very simplistic heuristics that check for morphological cues (see figure 4.3). Anything without a marked suffix defaults to being categorized as a single noun. This heuristic may generate parses that are false positives, or more frequently, may prevent legitimate parses from being considered. Another reason for parsing failures is that suffixes, without regard to context, are sometimes ambiguous, as the following examples from the test corpus illustrate: -ss (business vs. nonetheless), -es (engines vs. indicates), -s (evenings vs. various, perhaps, appears), -ed (accepted vs. infrared), -ing (allowing vs. during), -ly (approximately vs. apply).\textsuperscript{8} Lastly, even if the correct POS is determined, there are only a few subcategorization frames that are assigned by default (plural/singular and mass/count for nouns, and transitivity for verbs). This is a serious obstacle because the dynamic translator prunes all rules that involve features that are not listed explicitly in the lexicon or hypothesized for the known and new words respectively in the input sentence (see page 80).\textsuperscript{9} For example, PUNDIT fails to parse the following sentence although the heuristic correctly identifies the main verb, because it hypothesizes the verb is transitive, and hence all BNF rules in the grammar base that require the NULLOBJ (intransitive) feature are eliminated from consideration even before the parsing of the sentence has begun: “the low wavenumber component at 1 cm\textsuperscript{-1} \textbf{disappeared} first following evacuation.”

\textsuperscript{8}The relationship between suffix and POS is really many-to-many: not only can certain endings flag multiple POS, some categories have more than one morphological marker: irregular (Latin) plurals like “spectra, data, maxima” are quite common in the domain of physical chemistry. Verbs with irregular part participles, like “bend” and “set” are candidates for being added to the core lexicon.

\textsuperscript{9}There is, of course, another reason why PUNDIT may fail to produce any parse, which falls outside the scope of this dissertation: the construct may truly fall outside the coverage of the grammar, i.e., new grammar rules must be introduced, rather than updating the lexicon with POS and subcategorization information.
check_suffix(Word, Drow, Defns) :- ... suffix(S, Word, Rest, Defns), !.
check_suffix(Word, _, Word: [n: [root:[Word], new_word, singular]]). %default
suffix("ss", Word, _, Word: [n: [root:[Word], new_word, ncount1, singular]]) :- !.
suffix("es", Word, Root, Word: [n: [root:[Root], new_word, ncount1, plural]]) :- !.
suffix("s", Word, Root, Word: [n: [root:[Root], new_word, plural]]).
suffix("ed", Word, Root, Word:
    [ven: [root:[Root], new_word, objlist:[nstgo], pobjlist:[nullobj]],
     tv: [root: [Root], new_word, past, objlist:[nstgo]])].
suffix("ing", Word, Root, Word: [ving: [root:[Root], new_word, objlist:[nstgo]]]).
suffix("ly", Word, _, Word: [d: [root:[Word], new_word]]).

Figure 4.3. PUNDIT Heuristic for New Words

Of the 170 sentences in the test corpus, out-of-the-box PUNDIT returned a parse for 42 strings. The main cause of this limited success is that most verbs get mistagged as nouns because of a lack of morphological cues. If such sentences receive a parse at all, they are always false positives: they are labeled as fragments because of an apparent lack of a main verb (see figure 3.8), or an unrelated word with ostensible verbal suffix is “elevated” to status of main verb (see figure 4.4). There are six such obvious false positives among the sentences parsed.

There are two instances of false positives that are noteworthy because they involve members from closed classes: “beyond” and “rather_than”; it happened that neither was in the core lexicon, even though such prepositions and predeterminers really belong there since they are common to almost all domain languages. It is also worth pointing out that the verbs in table 4.1, which are defined with subcategorization frames, feature as main verb in thirty-three sentences of the test corpus, but only ten of these receive a parse.

A final category of error consists of adjectives mistagged as nouns. This is a fairly large group, but the deviance from the proper parse structure is minimal, and may be useful “as-is” for applications that do not require parses with detailed internal structure (compare “in the studies of the n=1 cluster, ( ( no ( vibrational/NN predissociation ) ) signal ) was observed if the pulsed laser was not present” with “… ( ( no vibrational/JJ predissociation )
after passing through the discharge region, the ionized gas **undergoes** collisionally **induced** vibrational relaxation in the 1.0-mm-long by 3.0-mm-diameter drift region before expanding through the 70&mgr expansion nozzle.

because of the large binding energies of the $n = 1$ and 2 ammoniated ammonium ions, absorption of sufficient energy from the tunable **infrared** laser to cause predissociation is not a facile process.
signal) was observed…”) and “((the most intense/NN) component (of 2&ngr;4)) is probably a parallel band” with “((the (most intense/JJ) component (of 2&ngr;4)) is…”). In some cases PUNDIT eventually provided the “correct” analysis when asked to generate all possible parses, despite the mistagging, but with the correct POS for the adjective, this parse was ranked much higher. For example, the parse of the conjunction below was the 26th alternative for “perpendicular” tagged as noun, but ranked 17th when corrected to adjective: “(it also includes ((the (&ngr;4 type)) fundamental (of NH3) and (perpendicular components (of ((overtone and combination) bands))))).”

This section has illustrated that porting an NLP system to a new domain requires large amounts of domain-specific information, most visibly in the area of lexical definitions. Without such knowledge, most sentences cannot be parsed, and many of the sentences that do receive a parse are actually false positives. PUNDIT, which is an all-symbolic system, identifies the POS of new words using a very simple heuristic that does not take context into account. The next section describes how stochastic information can be used to augment the knowledge-based parser to boost the number of sentences that receive a successful parse.

Trial 2: Lexicon Augmented with Stochastic POS Information

The stochastic POS information that was described in chapter III for the construction of the category space can be used to augment the knowledge-based PUNDIT system: instead of the simplistic heuristic that relies on suffix information alone for dealing with new words, a much more informed statistical model can be applied as a pre-processor. It is

10To some extent the different internal groupings are debatable anyway, and largely dependent on the particular linguistic theory subscribed to. The reason these sentences received a parse despite the mistagging is that PUNDIT has very liberal rules about noun-noun modifiers.
not only much more accurate, but can identify categories that have no distinct morphological ending.

\[(\text{newnoun}, \text{root: newnoun}, [n:[]]).\]
\[(\text{newnouns}, \text{root: newnoun}, [n:[\text{plural}]]).\]
\[(\text{newadj}, \text{root: newadj}, [\text{adj:[]}]).\]
\[(\text{newadv}, \text{root: newadv}, [d:[]]).\]
\[(\text{newprep}, \text{root: newprep}, [p:[]]).\]
\[(\text{newquant}, \text{root: newquant}, [q:[]]).\]

Figure 4.5. Vanilla Entries in the Lexicon

To measure the improvements of such a hybrid system, six “vanilla” entries were added to the lexicon (see figure 4.5). Based on the POS tag assigned by the stochastic tagger, words in the text to be parsed that are not in the lexicon are given the features of these vanilla entries, so that for all intents and purposes an input sentence like “gas-phase spectroscopic measurements for the ammonium ion and the ammoniated ammonium clusters have only recently been carried_out” could have been “newadj newadj newnouns for the newnoun newnoun and the ammoniated newnoun newnouns have only newadv been carried_out.” It is worth pointing out however that while the vanilla lexicon entries indicate the POS, no subcategorization features can be assigned to them because there are no defaults. Whereas the stochastic tagger is much more accurate in identifying the POS of a word than PUNDIT’s suffix heuristic, neither one indicates, for example, whether a noun is countable, or animate.

PUNDIT returned a parse for 67 sentences from the pre-processed corpus. All but three sentences that received a parse in trial one received a parse in this trial, although their assigned structure was not always the same as before.\(^{11}\) Not all of the remaining 28 parses were correct either, primarily because the vanilla lexicon entries were missing necessary

\(^{11}\)This helps identify false positives in trial one that are due to PUNDIT’s simple heuristic. The three sentences that received no parse in the second trial predictably turned out to have been false positives that were completely off target in the first trial.
subcategorization features. In three cases, there were still false positives due to mistagged verbs, either because of glitches in the stochastic parser, or because the word is known to the core lexicon, but under a different POS.¹²

Unlike the false positives of trial one, most of the “incorrect” parses in this trial were attachment problems. For example, in the following first parse, “observed” should be raised to main verb position: “(( ( the ( numerous low-frequency ) ) bending_motions ( available ( to ( the cluster ) ) ) ) are ( in ( the case ( of ( ( this ( ( work observed ) only ) ) ( as ( overtone absorptions ) or ( in ( combination ( with ( ( a high-frequency ) stretch ) ) ) ) ) ) ) ) ) ).” As figures 4.6 and 4.7 show, PUNDIT often does find the right parse when allowed to continue the search for alternative analyses. It will systematically try all combinations that satisfy the syntactic constraints, and delegate to the semantic component to choose the most likely parse among all the candidates. Although the correct parse¹³ shown in figure 4.7 is the 258th, PUNDIT did “correct” the misattachment of “species” early on (10th parse), but tried many other unsuccessful variations before raising the prepositional phrase “on the H5O2+ cluster.” The absence of subcategorization frames in most of the input words, which are responsible for imposing syntactic constraints (as well as permitting certain constructs), explains the sometimes huge number of variations PUNDIT considers.

The previous section has shown how one aspect of the lexicon, part of speech information, plays a major role in determining which syntactic rules apply to the input, and how a stochastic process which takes a larger context into account is more successful in disambiguating words for POS than a heuristic based on morphology alone. Although

¹²For example, the nominal entry for “peaks” allowed a successful parse for “this structure . . . could lead to peaks located roughly halfway between . . . ” but prevented a correct analysis of a previous sentence (“the weak feature . . . peaks in the correct position for . . . ”).

¹³Dr. Ray Reighart, a domain expert, was consulted to verify the proper parse of the sentence.
for the H502+ system studied previously by a messenger technique, the small perturbation by a H2 molecule on the H502+ cluster is thought to lead to the formation of an asymmetric structure involving species similar to H30+ and H20.

Figure 4.6. First Parse, with two Misattachments
Figure 4.7. Correct Parse: Number 258
knowing the correct POS is one step in the direction of avoiding false positives, other
syntactic information that is also stored in the lexicon is needed to avoid more subtle types
of false positives. The next section illustrates how the semantic component’s search for
correct attachments is hampered by a large solution space because of lexical entries with
unwanted or incorrect subcategorization information.

**Backtracking to Find the Next Parse Candidate**

Church and Patil (1982) have demonstrated that the number of possible groupings
for prepositional phrases and conjunctions grows exponentially, and this is certainly
evident in the large number of alternative parses PUNDIT returns for such constructs. The
system’s strategy is to start with local attachment of prepositional phrases to the nearest
noun (right association principle, Kimball 1973), and if additional parses are requested, to
progressively try attachments higher up the tree. Sometimes this allows the correct parse to
be ranked near the top, as in the following sentence, where all prepositional phrases are
attached correctly on the first try, with the exception of “in a pulsed radiolysis:” “the only
measurements of the spectra of the ammoniated substance ions have been limited to the
direct absorption measurements performed by Schwarz in a pulsed radiolysis of a gas cell
containing ammonia in a helium carrier.” In other cases many variations must be tried
before the right attachments are found, as in the following parse, which is ranked number
1,032: “(( the spontaneous ) decay ( ( of ( ( metastable parent ) ions ) ) ( ( to ( daughter
ions ) ) ( in ( the trap ) ) ) ) ) was ( as high ) (( as 5% ) ( for ( ( the largest ) ions ) and ( ( as
low ) as ( 10^-2 % ( for NH4+[NH3] ) ) ) ) ) ).” Occasionally, the right association

---

14 It should be stressed that several of the intermediate parses are false positives for
reasons other than prepositional attachment, primarily the result of missing or incorrect
subcategorization frames. And in quite a few cases the surface parse is generated more than
once, but with different deep structures. Another cause for the proliferation of parses in the
example sentence is the presence of a conjunction, which initially linked “largest ions” and
“10-2%.” In general, though, PUNDIT is much more adapt at handling conjunction than at
attaching prepositional phrases, because of the presence of syntactic cues in the former
case; in both of the two examples below, the first parse PUNDIT returned was correct with
principle is taken to extremes, and a missing noun is hypothesized as anchor, before a verbal attachment is considered (figure 4.8).

When all syntactic constraints have been exhausted in limiting the search space, the semantic component must choose/promote an interpretation using domain knowledge, world knowledge, and pragmatics. Witness to the difficulty of this task is the fact that sometimes it is hard even for human annotators to make a call, for example in a sentence like the following, where it is unclear whether “in the first solvation shell” modifies “rotation,” “molecules,” or “axes:” “nearly free internal rotation of the solvent molecules about their local c3 axes in the first solvation shell has been observed in the smaller clusters.”15 Consequently, it is very important for the syntactic component to maximize the syntactic constraints to limit the number of parse alternatives that the semantic subsystem needs to consider. Secondly, ranking the parse candidates passed to the semantic processor would help it focus on the most likely parses first. One Linguistic String Project subcategorization frame that can be used for this purpose is NPN, which indicates a strong selectional preference between a verb and a particular preposition, in essence directing the parser to attempt verbal attachment before defaulting to a local attachment to an intermediate noun. The effects can be illustrated with the following sentence: “( a shielding grid ( surrounding the expansion region ) ( maintained at 350 v ) ) prevented ( stray fields ) ( from affecting the ion trajectories in this region ).” PUNDIT correctly guesses that the

regard to the conjunction, despite the long distance gap: “the distribution of cluster sizes produced by the source could be easily monitored by (sweeping the field of the 60&degree sector magnet) and (counting the ions that arrive at the daly detector) with the second mass_filter set to transmit all ions”; “when (the source is cooled with liquid freon_22 instead_of freon_12), and (a higher pressure is used behind the nozzle), larger cluster ions could be favored as shown in figure 4c.”

Hindle and Rooth comment too about the semantical indeterminacy of many preposition attachments. To evaluate the performance of their system which uses distributional information to resolve attachments, both authors manually disambiguated a 1,000 sentence test sample, and found this “a surprisingly difficult task” with “more than 10% of the sentences . . . problematic” because “the attachment decision is not clear theoretically” (1993, 112).
The experimental apparatus *used in* this work has been described previously.

**Figure 4.8. Right Association Principle Run Amok**
unknown word “prevented” is the main verb and assigns it one feature: transitive. The first parse returned attaches “from” to “fields;” correct attachment to “prevented” is ranked fifth, but only the 75th parse alternative is totally correct (“maintained” attached to “grid”). By adding the NPN(from) feature to “prevent;” the very first parse correctly handles the prepositional phrase, and the desired analysis is boosted to 41st rank. Unfortunately the NPN feature only applies to V NP PP constructs and also does not function in favoring distant verbal attachment if there are intermediate prepositional phrases. Thus, the first parses returned for the following sentences attach the prepositional phrases headed by “in” to the local nouns (“Schwarz” and “series” respectively) even if the main verbs are subcategorized for NPN(in): “the direct absorption measurements were performed by Schwarz in a pulsed radiolysis of a gas cell containing ammonia” or “ab_initio molecular orbital calculations determined the structure and binding_energies of the series in reasonable agreement with high-pressure equilibrium measurements.”

Figure 4.9. Format of a Lexical Entry in PUNDIT

PUNDIT’s Lexicon and Subcategorization Features

Format of the lexicon

The importance of the subcategorization features in the lexicon can be illustrated with “show,” one of the verbs in the core lexicon, whose features were defined by PUNDIT’s experts. Its core entries are listed in figure 4.9. There are fourteen sentences in
the test corpus that use “show” as a verb. For most of these, the correct parse is among the list of candidate parses, after the unknown words were disambiguated for POS (trial two). Some of the constructs involve a that clause (e.g., “it will be shown that correlation of spectral features with cluster size can provide information about solvation shell structure, the geometry of the clusters, and the onset of a condensed-phase environment around the ion core”), others a direct object (e.g., “the spectra of the ammoniated ammonium ion series show a number of common features”); see figure 4.10 for some additional examples. In a test where the features were removed, none of these sentences received any parse, except the one where “show” was used in a subclause as a passive verb (second example in figure 4.10).

It is worth pointing out that the generic lexical entry for “show” provided by PUNDIT may contain features that are inappropriate for the sublanguage of physical chemistry, and these features have the effect of “bloating” the number of candidate parses that the semantic component has to choose from. For example, the first parse returned for sentence two in figure 4.10 is mostly correct, but misattaches the prepositional phrase locally to “region”: “this extra averaging is reflected in ( the better signal-to-noise_ratios (___________________________

16 The irregular plural of the subject kept PUNDIT from parsing this sentence in trial one because of a perceived number conflict with the main verb.

17 Some of the features in the core lexicon, like “h-*” or “NCOUNT2” were inherited from NYU, where the Linguistic String Project grammar was developed, but never used in PUNDIT (Marcia Linebarger, personal communication). Unfortunately, some of the functionality of these missing features has been hard-coded into the grammar. Occasionally, the grammar rules have also been “patched” to cover idiosyncratic behavior, triggered by sample sentences in a particular sublanguage. For example, PUNDIT cannot correctly parse the following sentence, although any preposition other than “of” does work: “such a system treats the vibrational modes of each subunit separately from the rest of the complex.” The “culprit” is a “d_of” restriction in the grammar, unrelated to the lexical features of any particular word.

We wrote d_of to cut down on spurious pp attachments in sentences like “he bought a picture of Mary”, where the “of” pp cannot be a sentence adjunct. There are many semi-idiom counterexamples (“of late”, “to talk of kings and cabbages”, etc.) and probably it occurs productively, but it’s worth losing a few cases to cut down on the junk parses. (Marcia Linebarger, personal communication).
Kebarle and co-workers showed that in the larger clusters there was a discontinuity in the $\Delta H$ for the $n=4 \rightarrow n=5$ clustering step.

This extra averaging is reflected in the better signal-to-noise ratios in the higher frequency region for some of the traces shown in figure 9.

The ion core frequencies associated with extensive hydrogen bonding show a gradual blue shift.
in (( the higher frequency ) region ( for some of the traces ( shown ( in figure 9 ) ) ) ) ) .” The correct attachment is parse number 85: “this extra averaging is reflected in ( the better signal-to-noise ratios (( in (( the higher frequency ) region ) ) ( for some of the traces ( shown ( in figure 9 ) ) ) ) ) .” As figure 4.11 shows, some of the parse candidates PUNDIT returns before arriving at the correct analysis are clearly false positives. Pruning the NSTGO feature from the lexicon eliminates these and promotes the right solution to parse number 67.

![Figure 4.11. Effects of Pruning Feature Set](image)

Another characteristic of the lexicon that can be seen in figure 4.9 is the structure sharing among morphological variants, by the use of pointers. For example, “shown” has some features in common with the root form (e.g., [3]), so the object options need be updated in only one place, but the entry also lists some object options of its own (e.g., VMOD).

There are three types of subcategorization frames that can be associated with words in the lexicon:\(^{18}\)

\(^{18}\)For a complete listing with examples, see appendix 3 of Sager (1981). Only a subset of these features are adopted in the PUNDIT implementation, as detailed in Linebarger (1988). In a few instances, PUNDIT actually makes finer distinctions than the Linguistic String Project (e.g., EQTOVO vs. TOVO in table 4.2), but these are exceptions.
(a) features that impose (selectional) restrictions on legal constructs. NOTNSUBJ (NHUMAN), for example, rules out “*the man occurred,” and NOTNOBJ(NSENT1) rejects “*he ate the fact.” Sometimes the restrictions involve specific words rather than classes, as with ASTG, which dictates the set of adjectives that can occur with particular verbs (compare “this category tested low in reading” vs. “*this category tested easy in reading”). Features of this first type are especially useful for language generation; for language recognition, the assumption is that the input is grammatical, and, if necessary, the grammar/lexicon rather than the input should be adjusted for sublanguage idiosyncracies. The value in parsing is that these restrictions help prune the number of candidates that are otherwise syntactically viable.19

(b) features that license certain constructs. Verbs marked as C1SHOULD, for example, can take a that-complement with an untensed verb, as in “for equilibrium, we require that the atomic chemical potentials be equal.” Strong selectional dependencies between verbs and certain prepositions can be indicated with PN, as in “direct momentum transfer satisfactorily accounts for the broad features of collisional activation.” Missing features of this type will definitely prevent the parser from finding an analysis (except for false positives).

(c) features that affect the interpretation of the sentence. For example, certain verbs followed by “as” parse their object as ASOBJBE (“in the capacity of”), rather than the generic NSTGO object (“while”); compare “enzymes function as catalysts” vs. “he lived abroad as a teenager.” Another example is a semi-closed class of verbs (VSENT4) with special treatment of “for” clauses: “it remains for future work to establish whether the troe or zewail approach gives better estimates” vs. the generic adjunct “for a typical

---

19The category space, when applied to nouns, is especially useful for finding broad classes like NHUMAN, NTIME, NSENT. Unfortunately, this is one of the features the designers of PUNDIT chose not to implement in their system.
experiment, signals were averaged for about 400 laser shots at each wavelength in the 2600-3200-cm⁻¹ region.”

**Inducing Object Options**

Table 4.2 shows the subcategorization frames that are induced by the systems of McKee and Maloney (1992), Brent (1993), and Manning (1993). The first column shows the matching feature labels from Sager’s Linguistic String Project; in many cases her grammar allows for more fine-grained distinctions (column three). Most of these, however, only affect the deep structure analysis, not the surface structure parse, so a superclass (shown in italics) was used in inducing trial number three. The technical term for EQTOVO is an infinitival complement with subject controlled equi, which means that the subject controls both the main verb (called matrix verb) and the infinitive. This is reflected in the Intermediate Syntactic Representation, which shows the subject with the same ID variable copied into the infinitive:

\[
\begin{align*}
\text{OPS: } & \text{assrt,present} \\
\text{VERB: } & \text{want} \\
\text{SUBJ: pro: } & \text{he (sing), }._705374 \\
\text{OBJ: } & \text{OPS: untensed} \\
\text{VERB: } & \text{repair} \\
\text{SUBJ: pro: } & \text{he (sing), }._705374 \\
\text{OBJ: } & \text{the disk drive (sing), }._14715 \\
\end{align*}
\]

Thus there are selection restrictions between the subject and both verbs, as the following examples demonstrate: “*the number 12 wants to be divisible by 3” and “*the cat wants to be divisible by 3” (taken from Linebarger 1988, 6). TOVO, on the other hand, is a

\[\text{OPS: assrt,past} \]
\[\text{VERB: say} \]
\[\text{SUBJ: pro: he (sing)} \]
\[\text{OBJ: OPS: past} \]
\[\text{VERB: be} \]
\[\text{SUBJ: the disk (sing)} \]
\[\text{PREDN: inoperable (sing)} \]

---

\[\text{20In the case of THATS and ASSERTION, the exact opposite is true: the surface differences (absence or presence of the complement) are regularized to a common deep structure (ISR):} \]
\[\text{OPS: assrt,past} \]
\[\text{VERB: say} \]
\[\text{SUBJ: pro: he (sing)} \]
\[\text{OBJ: OPS: past} \]
\[\text{VERB: be} \]
\[\text{SUBJ: the disk (sing)} \]
\[\text{PREDN: inoperable (sing)} \]
Table 4.2. Induced Features

<table>
<thead>
<tr>
<th>1:McKee/Maloney</th>
<th>2:Brent</th>
<th>3:Manning</th>
<th>finer distinctions in LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3</td>
<td>Nullobj</td>
<td>intransitive</td>
<td>the pump failed</td>
</tr>
<tr>
<td>1,2</td>
<td>Nstgo</td>
<td>transitive</td>
<td>she found the book</td>
</tr>
<tr>
<td>1,2,3</td>
<td>THAT</td>
<td>clause</td>
<td>he said that the disk was inoperable</td>
</tr>
<tr>
<td></td>
<td>ASSERTION</td>
<td></td>
<td>he said the disk was inoperable</td>
</tr>
<tr>
<td></td>
<td>C1should</td>
<td>tense-less</td>
<td>he suggested that it be replaced</td>
</tr>
<tr>
<td>1,2,3</td>
<td>INF</td>
<td>infinitival complement</td>
<td>he wants to repair the disk drive</td>
</tr>
<tr>
<td></td>
<td>EQTOVO</td>
<td>(equi)</td>
<td>he told the factory that the sac was inoperable</td>
</tr>
<tr>
<td></td>
<td>TOVO</td>
<td>(raising)</td>
<td>the pump seems to be failing</td>
</tr>
<tr>
<td>2,3</td>
<td>NTHATS</td>
<td>direct object + that compl</td>
<td>she told the factory that the sac was inoperable</td>
</tr>
<tr>
<td>2,3</td>
<td>NN</td>
<td>double object</td>
<td>he told her the result</td>
</tr>
<tr>
<td></td>
<td>OBJBE</td>
<td></td>
<td>I consider her a good dancer</td>
</tr>
<tr>
<td>2,3</td>
<td>NTOVO</td>
<td>(ecm)</td>
<td>the factory expects him to repair the sac</td>
</tr>
<tr>
<td></td>
<td>TRIAL 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,2</td>
<td>PNpval</td>
<td>prepositional phrase</td>
<td>the surgeon operates on the patient</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td></td>
<td>the surgeon operates on the table</td>
</tr>
<tr>
<td>2,3</td>
<td>TV-P</td>
<td>direct object +</td>
<td>they returned the disk drive to the factory</td>
</tr>
<tr>
<td></td>
<td>PNP</td>
<td>prepositional phrase</td>
<td>they returned to the factory the disk drive that almost caused a fire.</td>
</tr>
<tr>
<td></td>
<td>PNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OBJBE</td>
<td></td>
<td>I found her in a state of euphoria</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td></td>
<td>I found her on the fourth try</td>
</tr>
<tr>
<td></td>
<td>TRIAL 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
semi-closed class of aspectual verbs where the subject of the infinitive has been “raised” to the level of the matrix verb, but the sole argument of the latter is the entire infinitive, not the TOVO object:

OPS: assrt,present
VERB: seem
SUBJ:OPS: untensed,prog
VERB: fail
SUBJ: the pump (sing)

OBJTOVO, or object controlled equi, is an argument of both matrix verb and infinitive, similar to EQTOVO. NTOVO, finally, is known as exceptional case marking, where the NP following the matrix verb is subject of the infinitive but not a direct object of the main verb; instead the entire construct following the matrix verb is its argument, as was the case for TOVO and SVEN. Hence there are no selectional restrictions between the NP and the matrix verb, only between the NP and the infinitive (thus permitting “I believe the table to be quite attractive”).

The features shown in the shaded area are actually quite different object options, but their manifestation in the source text is similar to the target feature to the extent that it might trick systems that are based on localized pattern matching. SVEN, for example, flags a predicative small clause which in its entirety is the argument of the verb; “the book” is (obviously) not the argument of “found,” and hence its selectional restrictions do not affect the main verb (but they do impact “missing”).

The authors of the studies mentioned in table 4.2 deal with these extraneous features in several ways, ranging from distancing themselves from the argument-adjunct distinction, to being very conservative in specifying cues, to using statistical measures to filter

21 NTOVO but not OBJTOVO allows a pleonastic subject as in “I expect there to be unlimited champagne” (Linebarger 1988, 7).

22 Adjuncts are prepositional phrases that fill a temporal or locative slot that can occur with most any verb (e.g., “on the table”); arguments on the other hand, are licensed by the lexical entry of a particular verb (e.g., “operate on a patient”). Although some
out this noise. It should also be noted that some of these rogue options occur with a semi-closed class of verbs only.

Trial 3: Induction of Basic Features

In this trial, seven object options for verbs were induced (column one of table 4.2). Four of these are basic features in the Linguistic String Project (LSP); the other three (THAT, INF, NP-INF) are virtual superclasses that ignore some of the finer syntactic nuances of LSP.\(^{23}\) For testing purposes with the PUNDIT system, the induced superclasses for new words result in a lexical entry for each of the features they encompass, an overgeneralization which may “pad” the solution space but should still allow the correct parse to be among the candidates.

The information gathered from the category space described in chapter III addresses two issues: it directs the assignment of appropriate features to new words; and secondly, it governs the adjustment of the lexical entries of known words, correcting errors in some cases, but more often attuning the entries for the idiosyncrasies of the sublanguage. It turns out that solution to the second task helps steer the induction process for new words. The next section will look at adjustments to the existing lexicon and efforts to expand the research has been done to provide a formal basis to distinguish between the two (e.g., Jackendoff 1977, referenced in Brent 1993, and Pollard and Sag 1987, referenced in Manning 1993), “in practice, it gets murky” even for trained linguists (Manning 1993, 236). Brent abandons the distinction in his Learner system, because “it is not clear how a machine learning system could . . . [separate] the nonspecific cooccurrences of verbs with PPs (adjuncts) . . . from the specific ones (arguments)” (1993, 257). The bracketing guidelines for the Penn Treebank project similarly do not distinguish between the two constructs (Santorini 1991b).

\(^{23}\)The THAT superclass encompasses three LSP subcategorization features, one of which (THATS) is actually a superset of another member (ASSERTION); all verbs where the complement is optional also allow clauses with explicit “that.” PUNDIT does not have such a generalization rule in its grammar, but requires instead that the lexical entry list both features explicitly. For some verbs the “that” complement is mandatory, as in “*the director added Susan was a good scientist,” and hence only the THATS feature is listed. The only entry in the core lexicon with ASSERTION but not “THATS” is “tell,” but this is an apparent omission.
knowledge-based pool from which to draw inferences. Then an iterative process will be described that hypothesizes features for new words from partial information, and continually re-adjusts and finetunes the proposed lexical entry as coverage increases, until the system converges. Finally, the results of the induction will be analyzed by reparsing the test corpus with the updated lexicon.

**Attuning the Existing Lexicon**

In the hybrid system for NLP discussed in this dissertation, the category space developed in chapter III constitutes the statistical, or data-driven component, whereas the grammar rules and the core lexicon make up the knowledge-based part. The original lexicon is hand-crafted, a result of introspection by one or more linguists and possibly of manually introduced incremental “fixes” triggered by exposure to new linguistic phenomena.24 The goal is to let the features of a new word be a “composite” of the subcategorization frames of verbs it is syntactically close to (in category space). By the same token, the feature set of related words may “suggest” that the object options of a known verb need updating, i.e., that for the sublanguage being studied, an additional feature should be added, or an existing option can be removed.

When looking at known verbs only, with tight neighborhoods (i.e. a high cosine similarity index of 0.98), the feature sets are fairly homogeneous. For example, “provide”

---

24As Hirschman et al. explain, the philosophy of the PUNDIT development team in lexicon construction is to keep extending the system in order to attempt to “capture the general English definitions” rather than just “domain-specific usages encountered in the training data.” Rarely does any pruning occur:

The rationale is . . . similar to that used in the syntactic component: generation of lexical entries is a time-consuming process; our goal is to develop a broad-coverage system, so when entering a word in the lexicon, we enter the general English categories for the word. In many cases this provides a much more general definition than what is specifically required by an application. . . . The disadvantage of this approach is that as the coverage grows, the grammar becomes “looser” -- the number of parses for any given word sequence tends to increase and also the grammar tends to overgenerate, letting through constructions that are not grammatical (Hirschman, Lang et al. 1989, 277).
and “give” which both occur over 1,700 times in the training corpus are (not surprisingly) grouped together, and both share the features NN and NSTGO.\(^{25}\) The few exceptions, where there is some mismatch between the feature set of related verbs, are actually significant; for example “confirmed” is similar to “determined,” and where they both have NSTGO and THAT in common, there is no confirmation for the NTHATS feature (mistakenly?) associated with “confirmed” in the lexicon. Another discrepancy illustrates how the lexicon can be customized to the domain: THAT is certainly a valid option of “explain” for generic English, but it turns out none of its 800+ instances in the training corpus use this construct so it can be safely removed without any loss of coverage. In fact, there are 88 sentences where “that” occurs in the right context of “explain,” but they are all relative clauses or used as determiner: “the present results cannot explain the fact that the low-temperature echo decay time is about CD times shorter than the fluorescence decay time” or “although that appears to be a surprising sinergism [sic] result and might raise some questions about the accuracy of Kwiatkowski’s experimental data, a second-order expansion might explain that unusual behavior.” Eliminating the THAT option will reduce the solution space and almost certainly avoid some false positives for sentences like the ones above. It is not clear how the pattern matching approaches of McKee and Maloney (1992), Brent (1993), and Manning (1993) would differentiate the above cues from valid complements.

A last mismatch of features involves “needed:” there is confirmation in its neighborhood for NSTGO and TVINF, but not for INF (actually EQTOVO). Once again, the latter feature is certainly valid for generic English, but none of the 424 instances of “needed to” in the corpus of physical chemistry are used as infinitival complement. Instead they

\(^{25}\)That is not to say that the \textit{full} lexical entry of both verbs doesn’t exhibit some differences — “give” has more intransitive options, and the prepositions they have an affinity for are dissimilar — but for the features being induced in trial three, both verbs share the two mentioned and lack the other five.
function in adjectival sense, sometimes with silent “BeAux,” as in the following examples: “theoretical calculations are needed to distinguish between these CD possibilities,” and “the resulting data have been utilized to derive the partition function of the molecule [that is] needed to determine the dissociation energy from the mass spectrometric data.” Figure 4.12a shows the false positive that results from this feature. This was not PUNDIT’s first choice, (in fact it tried the “sa_final” option first), but because additional parses were requested to correct the misattachment of “to symmetry,” the Prolog system systematically backtracks to each choice point to try the next alternative. Thus removing the legacy EQTOVO feature results in savings (in terms of pruned search space) that are greatest when the verb occurs towards the end of the sentence, because input is processed left-to-right. Figure 4.12b shows the correct solution, parse number 70. For the unabridged sentence the number of alternatives tried would have been even greater.

The stated purpose of the core lexicon (domain-independence) and the design philosophy (add all allowable options, for broad coverage) imply that fitting the lexicon to the domain language will primarily involve pruning options that don’t apply, but rarely will there be a need for extensions.26 Appropriate verb entries for this lexicon would be modals, auxiliaries, and maybe very frequent verbs like “allow,” “come,” “give” or “use” (cf. table 4.1). It turns out that quite a few entries in the core lexicon should really be delegated to a separate domain-dependent lexicon; for these entries it is more likely that new subcategory frames will be induced.

The main problem with using the core lexicon as a knowledgebase (oracle) is that, while detailed, there are definitions for only a small (164) number of verbs. To enlarge the pool of subcategorized verbs that can be tapped for inducing features, entries were added from the Linguistic String Project “manual” (appendix 3 of Sager 1981). These are

---
26If new features were to be found for verbs that truly belong to the core lexicon, a case could be made to make those permanent additions for use also in other sublanguages, in keeping with the goal of being all-encompassing.
Specifically, we found that relaxing the restriction of the diffusion tensor to axial symmetry provided the additional degree of freedom needed to obtain satisfactory fits to the linewidth data at both CD and at 9.5 ghz, as we discuss in greater detail in the next section.

Figure 4.12. Pruning of Solution Space by Removing Legacy Feature ("EQTOVO")
examples, lists of verbs that illustrate the subcategorization frames as they are presented alphabetically. Although a slight bias towards the medical sublanguage at times surfaces, most of the new entries are generic in nature, or at least applicable to scientific writing (Sager 1981, 337). Some 3,342 verb-feature pairs were added to the oracle, or an average of 6.22 features per verb. Some of these 537 verbs from the manual already were in the core lexicon, but even then the definitions were expanded with additional features. The main difference between the PUNDIT entries versus the manual entries is that the latter, being illustrations, are incomplete: “suggest” is given as an example for the C1SHOULD feature, but is not in the short list of verbs that explain the NSTGO option. Thus it will be common for “new” features to be induced for these “known” verbs.

Table 4.3. Near-Perfect Matches

<table>
<thead>
<tr>
<th>achieved</th>
<th>detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>achieved</td>
<td>employed</td>
</tr>
<tr>
<td>assigned</td>
<td>attributed</td>
</tr>
<tr>
<td>calculating</td>
<td>evaluating</td>
</tr>
<tr>
<td><strong>carried_out</strong></td>
<td><strong>performed</strong></td>
</tr>
<tr>
<td>decreases</td>
<td>increases</td>
</tr>
<tr>
<td>detected</td>
<td>evaluated</td>
</tr>
<tr>
<td><strong>determined</strong></td>
<td>evaluated</td>
</tr>
<tr>
<td>investigated</td>
<td>studied</td>
</tr>
<tr>
<td>occurs</td>
<td>takes_place</td>
</tr>
</tbody>
</table>

A second approach to increasing the size of the pool of verbs with a priori features in the lexicon is to make the neighborhoods from the category space less “tight,” i.e., to lower the cosine-similarity threshold. At 0.990 two verbs would have to appear in the training set with virtually identical syntactic and lexical contexts; as table 4.3 shows, this
occurs extremely rarely, and few of these pairs have a full definition in the lexicon. When the cutoff point is lowered to 0.980, the impact of small lexical differences in the context digests is lessened but the syntactic behavior of such a pair of verbs is still substantially the same. Because the neighborhoods are larger, the chance of some of the neighbors having a priori features increases; at ≥ 0.98, there are 104 pairs, at ≥ 0.97 384 pairs, and at a cosine similarity factor ≥ 0.90, there are 14,504 pairs of related verbs.

The tradeoff with these “looser” neighborhoods is that the quality/relevance of the information becomes more suspect as the cosine similarity measure decreases. In a sense, the measure can be viewed as a confidence weight: each verb has its own neighborhood, formed by computing the cosine similarity weight between it and all other verbs in the category space, and by retaining those whose weight exceeds a certain threshold, say 0.60. If there are no nearby verbs with known features, more remote words can be used for deciding on whether a certain feature should apply to the verb being examined, especially if a substantial majority of these “distant relatives” are in agreement. If the seven features under consideration are treated as boolean values (present/not present), it will most certainly happen in neighborhoods with liberal cutoff points that there will be some disagreement for individual options, so a heuristic must negotiate these “conflicts” and settle for the best abstraction. Such a heuristic should have the following three characteristics:

---

27 Verbs marked in bold are in the core lexicon; italicized verbs have a partial feature set, taken from Sager 1981. Most of the latter however involve esoteric features like VMOD and VINGOFN. Verbs that subcategorize for VMOD can occur in an environment with a sentential subject and an object from a special class of nouns like “trouble,” or “truth.” “for him to leave now would only increase our difficulties” (this special noun class called NSENT3 requires a be-verb to occur with embedded sentences; compare “the problem is that we need money” vs. “*the problem that we need money bothers us”). Verbs marked with VINGOFN can take as object the noun-like sequence determiner - gerund - nounphrase: “they have decreased the sending of supplies.” It is interesting to note that the antonyms “increase” and “decrease” are used by Sager to illustrate different features, i.e., in the manual they each show a disjoint subset of their common object options.
1) verbs that are close to the word being examined should carry more weight in the decision process than verbs that are closer to the perimeter.

2) both positive and negative evidence (the absence of a feature for a particular verb) should be considered.

3) given the fact that the presence of a feature is the result of a positive decision/action (by a linguist), whereas the absence may be an oversight, there should be a (slight) bias in favor of the former.

Figure 4.13 shows the algorithm (written in perl) that attempts to incorporate these three principles. When a feature is absent in one of the members of the neighborhood, the score for that feature is reduced, and conversely, when evidence is found to support adoption of the subcategorization frame, the score gets boosted. For this purpose the binary toggle absent (0) - present (1) gets mapped onto the range \([-1,+1]\), à la MYCIN (Shortliffe 1976). A final score close to the limits yields a clear-cut decision, but if it ends up around the center area (0) the resolution becomes a judgement call.

To give more importance to verbs near the word under examination, the confidence weight can be skewed so that the impact quickly drops off as the distance increases, instead of being evenly distributed. Figure 4.14 illustrates this graphically (in this implementation, anything below the cutoff point of 0.6 is not considered): for example, a verb with cosine similarity of 0.70 and not marked with a certain feature would reduce the score by 0.70, 0.49, or 0.168 for a skewing factor (power) of 1, 2, and 5 respectively.

Lastly, to accommodate the principle that positive evidence should slightly outweigh negative evidence, the sensitivity threshold can be adjusted by shifting the point at which the weight of evidence is considered sufficient to decide in favor of adopting the feature. For example, if the threshold was set to \(-0.125\) and the skewing factor to \(x^2\), then if the

---

But unlike MYCIN, the final score does not get mapped back into the \([-1,+1]\) range because these are not certainty factors that need to be passed on for further combination. The sole function is to make a decision which is local in scope.
#!/bin/perl
$skew_factor=2; $accept= -0.125;

# for each centroid, do the following:

%current_features=split(/,/, $feature_knowledgebase{$centroid});
foreach $feature ("NULLOBJ","NSTGO","THAT","INF","NTHATS","NN","NP-INF")
{ $score=0; # reset
  %words_in_cluster=split(/,/, $category_space{$centroid});

  foreach $verb (keys %words_in_cluster) # associative array with verb-weight
  {
    %features_of_this_verb=split(/,/, $feature_knowledgebase{$verb});
    if ($features_of_this_verb{$feature}) # 1 -> present: add
      $score+=$words_in_cluster{$verb}**$skew_factor;
    else # 0 -> absent: subtract
      $score-=$words_in_cluster{$verb}**$skew_factor;
  }
  if ($score >= $accept) {$current_features{$feature}=1;}
  else {$current_features{$feature}=0;}
}
$feature_knowledgebase{$centroid}=join(",", %current_features);

Figure 4.13. Scoring Heuristic

Figure 4.14. Skewing Factor to Give More Weight to Close Neighbors
neighborhood consisted of two verbs with weight 0.7 and 0.75, which disagreed on a feature (on and off, respectively), the presence of the feature would win out \((0.7^2 - 0.75^2 = -0.07 > -0.125)\). For a skewing factor of \(x^2\), the furthest the threshold could be pushed back is approximately -0.35, since otherwise a singleton with negative evidence, at weight 0.6, would be interpreted as positive evidence.

Figure 4.15 shows a sample neighborhood of eight known verbs around “required,” which has \textit{a priori} features NSTGO and NP-INF.\(^{29}\) All the verbs that are relatively close to “required” (\(\geq 0.92\): “considered,” “selected,” “allowed,” “found” and “needed”) share both features, with one exception which appears to be an omission. “Needed” and “found” also have an extra feature (INF and NN respectively), but in each case because of lack of corroboration the feature’s low score guarantees that it will not be added to the model. NP-INF is missing from a few of the members, but these are distant enough to lower the score only moderately. Finally, the evidence seems to suggest a new feature for “required:” THAT. This construct does indeed occur in the domain of physical chemistry, as in “furthermore, as in previous studies of hyperquenched dilute solution glasses, the lowering of \(tg\) of solutions \textit{required} that the initial part of the scan be as flat as possible.”

There are two parameters in the heuristic described above that can be adjusted to finetune the model: the skewing factor, and the accept/reject decision point. Optimal settings for these parameters can be determined using a data-driven approach: re-run the algorithm for verbs with \textit{a priori} features, trying different values for skewing (from \(x\) to \(x^5\)) and adoption thresholds (starting at 0 in decrements of 0.025). The “best” settings are those that recover most of the original features. The tool used to quantify performance is the

\(^{29}\)Actually the example is simplified in that there are quite a few more knowns, but they are more distant and do not alter the outcome significantly; the same three features NSTGO, THAT, and NP-INF emerge.
<table>
<thead>
<tr>
<th>required</th>
<th>allowed</th>
<th>confirmed</th>
<th>considered</th>
<th>determined</th>
<th>explained</th>
<th>found</th>
<th>needed</th>
<th>selected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.92</td>
<td>0.81</td>
<td>0.96</td>
<td>0.87</td>
<td>0.76</td>
<td>0.92</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>INF</td>
<td>-0.85</td>
<td>-0.66</td>
<td>-0.92</td>
<td>-0.76</td>
<td>-0.58</td>
<td>-0.85</td>
<td>0.85</td>
<td>-0.88</td>
</tr>
<tr>
<td>NN</td>
<td>-0.85</td>
<td>-0.66</td>
<td>-0.92</td>
<td>-0.76</td>
<td>-0.58</td>
<td>0.85</td>
<td>-0.85</td>
<td>-0.88</td>
</tr>
<tr>
<td>NSTGO</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NTHATS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NULLOBJ</td>
<td>-0.85</td>
<td>0.66</td>
<td>-0.92</td>
<td>-0.76</td>
<td>-0.58</td>
<td>-0.85</td>
<td>-0.85</td>
<td>-0.88</td>
</tr>
<tr>
<td>THAT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP-INF</td>
<td>0.85</td>
<td>-0.66</td>
<td>0.92</td>
<td>-0.76</td>
<td>-0.58</td>
<td>0.85</td>
<td>0.85</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Figure 4.15. Sample Induction for “required” with Skewing Factor $X^2$
recall/precision measure from information retrieval (e.g., Salton 1983, 164). Given table 4.4, the following definitions apply:

\[
\text{Recall} = \frac{a}{a + c} \quad (4.1)
\]

and

\[
\text{Precision} = \frac{a}{a + b} \quad (4.2)
\]

Table 4.4. Contingency Table for Recall/Precision

\[
\begin{array}{|c|c|}
\hline
\text{induced features} & \text{a priori features} \\
\hline
\text{present} & a & b \\
\hline
\text{absent} & c & d \\
\hline
\end{array}
\]

Recall, in other words, is the proportion of features in the original lexicon that the induction application also derives. The measure referred to as precision in the literature is a misnomer in this case, but refers to the proportion of features derived by the induction process that were in the original lexicon. Assuming that features in the original lexicon are highly reliable, a high recall is desirable. A lower “precision” mark indicates that new features were induced, but this is of course the objective, either because the original definitions had “gaps” (the selective entries from the Linguistic String Project appendix) or because the discovered feature is particular to the sublanguage and does not apply to generic English. Choice of threshold value dictates how conservatively the features are adopted from verbs with similar distribution. A lower recall score signifies that some features were dropped. This too can be warranted (e.g., figure 4.12) but it should be done conservatively, because whereas adding an incorrect feature may lead to additional parses that are false positives, missing features may prevent the parser from finding the correct analysis. It should also be remembered that features in the core lexicon are there because of their presence in the body of general English use. While their absence in the sublanguage
sample may be significant, for verbs that have occurred only a limited number of times, such absence may only reflect that the verb has not occurred often enough in the corpus for the feature to have a good chance of manifesting.

Figure 4.16 plots recall/precision figures for varying threshold values. As the adoption threshold is lowered, “precision” counts predictably drop (a desirable effect), but more so for higher skewing factors. There is also a certain adoption value at which the decline temporarily halts (in fact shows a slight rise), but this point comes earlier with increasing skewing power, until there are no gains (beyond $x^3$). Recall also improves with more liberal (i.e., more negative) adoption values, and also with higher skewing factors.

A spot-check of the new features proposed by the induction process for known verbs appears to suggest that there is a slight overgeneration of the NSTGO feature. Some of these seem intuitively valid, and evidence can be found in the domain corpus. For example, the assignment to “allowed” is not surprising in view of the fact that it stood out as the only close neighbor of “required” that lacked this feature (figure 4.15). It is encouraging to find that this is independently suggested also for “allow, allowing#vbg,” and “allows” given that each morphological variant of the verb has its own cooccurrence vector and hence its own neighborhood.\footnote{A side effect of this is that the conjugated form of a verb is often a member of its own neighborhood, sometimes with high similarity weights, as in “show-showed: 0.98”} In other cases the assignment is more suspect, for example for “exist,” “proceed,” or “remain.” Some overgeneralization is probably unavoidable, but there are two mitigating circumstances:

1) The induction process has to work with a rather small starting base of verbs with \textit{a priori} features, sometimes forcing it to consider more distant neighbors over closer ones because the latter are all unknown words.
Figure 4.16. Data-Driven Optimal Thresholds Assignment

Figure 4.17. Thresholds Assignment Sorted by “cosine”
2) A spurious NSTGO feature will do no worse in parsing than unmodified PUNDIT, since this is the one (and only) feature that the unmodified system assigns by default to any new verb.

The second most commonly proposed feature is NULLOBJ, and a random check did turn up some examples: “so far as we know, only Maessen and Cade have published studies on the electronic structure of the alkali-metal molecules mx2” and “we allowed several days to pass before we did the measurements of raman spectra.”

The majority of the *a priori* features that are not being replicated by the induction process are either THAT or NULLOBJ. Two verbs were selected randomly from the list to check if the removal of the THAT feature was justified; “determined” *does* occur in the training corpus with a clause albeit rather infrequently (“these investigations have determined that vni possesses a ground state, with the unpaired spin having approximately 10% 4s character”). The omission for the second verb (“represent”) seems warranted though, as none of its 915 instances appear with a that clause.\(^{31}\) Actually, the verb “represent” has multiple senses with different syntactic constructs, which are lumped together in the core lexicon; “to express angrily” does subcategorize for THAT, as in “I represented [to him] that it would be dangerous to do what he suggested” but because it isn’t used in this meaning in the physical chemistry domain, as indirectly suggested by the induction process, it should be removed to avoid false positives.

\(^{31}\)As was the case with “explain,” a pattern matcher looking for “evidence” of a clause might easily be tricked by sentences like the ones below, especially by the second one, because of the presence of “thought”: “in this diagram, arrows show the direction of a possible transformation, solid lines indicate exact and consistent transformations, dotted lines denote inconsistent transformations, and dashed ones *represent* transformations that involve the solution of an inverse problem and thus such transformations are correspondingly uncertain and nonunique” and “the subunit considered in this paper is a quasi-octahedral tio62- group, modified as explained below, chosen to *represent* that portion of the structure thought to be most important for the nonlinear response.”
Inducing new Features: an Iterative Process

Figure 4.15 illustrates how a word acquires a new subcategorization frame based on the distribution of feature values for verbs to which it is syntactically similar, in category space. Because “required” itself is in the neighborhood of other verbs, and hence a source contributing to their feature assignment, the addition of THAT causes all these other verbs to be re-evaluated, and may shift the balance just enough to trigger some of them to also adopt this option.32

So far only verbs with pre-existing features have been considered, in an attempt to “fit” their definitions to the sublanguage domain by “reconciling” the subcategorization frames that dominate the neighborhoods of those verbs (incidentally, these neighborhoods themselves have been limited to verbs with a priori object options). A second motivation for constructing the category space, with even greater impact on parsing performance, is to induce lexical/syntactic information for the vast majority of verbs that are new. The two objectives are closely linked, however, in that initially decisions will be based solely on a priori information, but as more new words are receiving feature assignments, they will be added to the pool of information to draw upon, and they may eventually impact some of the verbs in the original lexicon.

The induction then should be seen as an iterative process where changes are propagated, causing re-evaluations and possible modifications, which in turn would set off a ripple effect, until the system converges (no more significant changes occur). The driving force is a priority queue of words “waiting” to be evaluated. The queue is sorted by percentage of already defined verbs in the defined neighborhood of each entry. This is to keep the number of passes that need to be made to a minimum; words with a full deck (all members in the neighborhood have a feature set associated with them) should be processed

---

32Most of the time, though, the change will just reinforce the previous assignment. For example, the addition of NSTGO to the feature set of “allowed” (as discussed earlier) does not affect “required” except to boost the score from 4.64 to 6.35.
first because a very informed decision can be made which will likely stick; verbs with one or more unknowns will definitely need to be revisited later when the missing information becomes available. There will be times when there is no choice but to make a partial decision, but if a verb with several unknowns is scheduled late, chances are that many of these will have become resolved by the time it reaches the front of the queue.

Initially all verbs (predefined and new) are ranked and entered in the queue. When the front of the queue contains a verb which already has a feature assignment (either from the core lexicon or because it has been through a cycle before), it is evaluated, and, if necessary, the knowledge base is updated to reflect any changes in object options. The verb is then removed from the queue. If, however, the verb being evaluated has not previously had any subcategorization frames, all verbs that depend on it are re-ranked (since there is one more known in their own neighborhood) and either re-entered in the queue or promoted to reflect the new priority.

When the queue is empty, all verbs have been evaluated at least once and should have no undefined members in their respective neighborhoods. However, some of their neighbors may have been updated since their own time of evaluation, so another complete pass is made. At this time, if any changes in assignment should occur, only dependent verbs which are very close (cosine similarity > 0.90) are notified and added to the back of the queue for re-evaluation. 33 In the majority of cases this reassessment results in the score of an individual feature being boosted without affecting any assignments (cf. “allowed”). If some change should occur, another cycle is started (for close dependents) until the process eventually converges.

---

33 Verbs with *a priori* object options from the core lexicon make up a separate class and are treated a little differently. Even in the first cycle they only consider close neighbors unless the neighboring verb is from the same class, in which case more distant “cousins” are included in the decision process.
The queue was initialized to 1,147 verbs that have at least one other word in the category space that is syntactically close (cosine similarity > 0.85). During the first cycle, the queue was re-ranked 621 times, triggered by a “new” verb receiving a feature set assignment for the first time. Each time, all the words whose neighborhood contains this previously feature-less verb are scheduled for re-evaluation in order to allow the new information to bear on the decision process. Affected verbs which were already (still) in the queue waiting to be evaluated are shifted towards the front of the queue, which is prioritized to always process words with the smallest percentage of feature-less neighbors first (hence the term re-ranking). Words outside the queue whose neighborhood does not include the verb that triggered the re-ranking are not affected.

In the second pass, during which only “significant” changes are propagated, the queue was re-ranked nine times, at which point the system converged. On average, verbs were re-evaluated 36.62 times; “utilizing,” which has a very broad neighborhood, was calculated 148 times.

It turned out that the queue was not completely empty at the end of the first pass; there is a small group of related verbs that have no ties to any of the verbs with a priori information, or to new verbs that received induced features. Because these verbs form an “island” in the category space, no subcategorization frames can be induced for them unless some of their members are defined in advance.

Another category of verbs where no lexical information is learned directly are those that occur too infrequently to be included in the category space. There are only eight instances of “considers/vbz” in the training corpus, for example, well below the cutoff point of 1,430 (see chapter III). In those cases the features induced for a related conjugation (“consider/vb,” frequency 1,498) can be “borrowed” since in most circumstances their syntactic behavior is governed by the same rules.
The Test Corpus Revisited

One way to evaluate the outcome of the induction process is to compare it against an on-line dictionary that provides subcategorization information, such as the OALD. McKee and Maloney however object that because such dictionaries are generic in nature and not attuned to a particular sublanguage, they are “of little use:”

In a random sample of 50 verbs from the American Heritage Dictionary, 30 have both transitive and intransitive readings, e.g. murder. In everyday use (as well as in the MUC domain texts), the word is used almost exclusively as a transitive. (1992, 85)

Manning reports precision and recall figures of 90 and 43 percent respectively, when comparing his learned verb features against the OALD, but protests that these are “pessimistic lower bounds” because “the dictionary lists subcategorization frames that do not appear in the training corpus and vice versa.” He adds that the learned corpus “may reflect language use [of the NY Times newswire] more accurately than the dictionary” (1993, 239).

Perhaps a better way to evaluate the quality of the induced features is to compare the parsing performance of an NLP system like PUNDIT on a held-out test corpus before and after the lexicon is updated. Of the 170 sentences in the test corpus, out-of-the-box PUNDIT returned 42 parses, a number which was boosted to 67 when given some assistance in assigning POS to the input words. The new lexicon allows PUNDIT to successfully parse 94 sentences, for a net gain of 27.\textsuperscript{34} It should be pointed out that these 94 sentences contain an average of 2.14 verbs, not counting modals (“could be monitored”) or adjectival usage (“lists the measured constants” or “is constant with increasing

\textsuperscript{34}There are really more than 27 new sentences that receive a successful parse; some of the analyses that were returned with the original lexicon could not be handled by the updated system (see next section). There are also four sentences that might have received a parse, but after the system pondered each of them for 24 hours without producing any result, the plug was pulled.
solvation”). Most of the sentences that received a parse in trial one and two are shorter, less complex ones.

A breakdown of the new successful parses shows few additions that required the NSTGO feature, because that is the one object option PUNDIT hypothesizes for all unknown verbs it identifies. But when the verb ending is irregular (as in “the beam is bent . . .”) PUNDIT’s simple heuristic falters unless a lexicon lookup finds an explicit entry (courtesy of the induction process). Some of the other sentences now successfully parsed involve NULLOBJ (“the loss . . . may occur,” “the spectra change very little . . .”) and INF (“pullman and arm Buster . . . managed to reproduce . . .”). The majority of the cases though depend on the THAT feature (“prove by isotope substitution that . . .,” “it is estimated that . . .,” “suggests, as in the case of 2-butyne, that . . .”). The following successfully parsed sentence is representative of the complexity of the language in the physical chemistry domain, here in part due to the parenthetical participial phrase at the end: “little dependence was observed between the peak of the cluster ion distribution and the magnitude of the current used in the discharge for a given value of the above parameters, indicating that the ions were at a reasonable equilibrium with the temperature of the source body before expansion through the nozzle.”

The reason some of the “successful” parses from trial two were not replicated can be summed up by the fact that the induction process not only adds features but also removes some deemed inappropriate. In the case of previously unknown verbs that were hypothesized to be NSTGO, deriving a different feature set sometimes exposes that parse to be a (subtle) false positive. For example, the following sentence will no longer parse now the main verb is marked only with NULLOBJ: “many investigators contributed in the
effort to determine the $\Delta H^\circ$ for formation of the solvated ammonium ion NH4+(NH3)n.”

Of course the induction process at times removes a feature in error. A case in point is that “contribute” is occasionally used transitively: “we could determine that reaction 3b contributed approximately 10% of the total reaction.” A possible reason the category space / induction process did not recognize this is that such constructs are extremely rare in the training corpus; most of the use involves “contribute from,” “contribute in,” or “contribute to,” and if used transitively, the construction is usually passive (e.g., “the large hydrophobic effects contributed by these groups tend to mask the electrostatic effects”).

The three reasons why some of the input in the testbed does not receive any parse can be summarized as follows:

1) The induction process fails to identify the needed feature, as with “contributed.”

2) PUNDIT requires some additional features, other than the ones being considered here, usually involving prepositions and particles in argument or adjunct position (e.g., “result:” DP2(from)).

3) There are genuine gaps in the grammar, i.e., constructs that PUNDIT cannot handle even with full lexical knowledge. Given the complexity of the input in the domain of physical chemistry, this is not surprising, but it can only be exposed after careful experimentation. For example, although the lexical entry for “observed” was correctly induced as NSTGO, PUNDIT cannot process the following sentence because of the relative clause: “no transitions were observed that could unambiguously be assigned to the solvent molecules surrounding the ammonium ion.” If the input is changed to “no transitions that could . . . were observed” the sentence parses fine. Another example of a sentence outside the coverage of PUNDIT is the following input: “the frequency of a

---

35In trial two, the (only) parse returned incorrectly attached “determine” to “contribute” instead of to “effort.” PUNDIT requires a separate feature DP2(in) to handle constructs like the one above.
given transition for a particular subunit will be shifted from that of the free molecule by an amount determined by . . .” Replacing “that” with a copy of “the frequency” results in a parse, demonstrating that the correct subcategorization information is in place. A final example to illustrate the important point that parsing failure is not always due to the lexical features is the following sentence: “the NH3 solvent molecules in the first solvation shell will be strongly influenced by NH3 molecules in the second solvation shell and beyond.” When the conjunction with the adverb at the end is stripped, the sentence parses correctly.

Another aspect to consider is whether the removal of unnecessary features from the lexicon reduces the solution space by pruning false positives for those occasions when PUNDIT needs to backtrack in search of correct preposition attachments. For example, the induction process removed the INF feature from “determine” (as in “he determined to go at once”). This is really a different sense from the one used in the physical chemistry sublanguage, and without this undesired option, the parse with correct attachments for the sentence below came 43 percent sooner: “the signal-to-noise_ratio for the detection of ClO in the mass_spectrometer was too small to determine the stability of all species relative to the reactants.”

However, in comparison with trial two there were few such savings, because of the small number of verbs with pre-existing features. It should also be noted that for all the verbs in the core lexicon, the only features that were reconsidered were those listed in table 4.2; the other features remain unchanged. Because features were also added, they sometimes offset savings elsewhere by introducing false positives, even if the adoption of the feature was justified; for example NSTGO for “allow” causes “processes” to be attached to the verb when backtracking for alternative prepositional phrase attachments: “the above formula corresponds to the most simple approximation for the charge fluctuations
based on the neglect of crossterms which allow for a mutual coupling of correlation processes.”

Trial 4: Recognizing Finer Distinctions

Trial three was repeated, but this time the virtual classes (printed in italics in table 4.2) were replaced by their individual Linguistic String Project features: THATS - ASSERTION - C1SHOULD, EQTOVO - TOVO, and OBJTOVO - NTOVO.

The ASSERTION feature was induced for only nine verbs, indicating that this is a rare construct for the sublanguage of physical chemistry, which is mostly technical and formal. None of the 170 sentences in the test corpus required this subcategorization frame, but some evidence in the training corpus of 339,842 sentences does exist: “for example, suppose we had the distance scaling in section 5 rather than &ngr;3.” Because all verbs in trial three that were flagged for the THAT class automatically received both THATS and ASSERTION, this overgeneration caused many false positives during backtracking, as in “these measurements revealed ( direct IR absorption ( of ammonium salts ) ) dissolved ( in solution ),” where the underlined part is interpreted as a tensed clause. This is remedied in trial four because only a few verbs are subcategorized for ASSERTION.

No verbs were subcategorized for the C1SHOULD feature by the induction process. This is a direct result both of the fact that only four verbs in the core lexicon have this feature (“arrange,” “recommend,” “request,” “specify”), and that all four are very rare in the training corpus (less than 100 times for the first two, less than 50 for the latter two”). None of the sentences in the test set require this feature. C1SHOULD is for all practical purposes a closed class, and hence not a good candidate for induction. Since there are only a handful of infrequent verbs involved, it is probably safer to abide by the core lexicon and leave this feature alone.
The same conclusion likely applies to the TOVO feature; it applies to only twenty verbs in the core lexicon, just six of which (“consider,” “found,” “required,” “taken,” “said,” and “mean”) occur frequently enough in the training corpus to be included in the category space. The first four verbs all appear very close together in the space (cosine similarity between 0.93 and 0.97), but because there are only four of them, there isn’t enough positive evidence to survive the filtering process. Because the feature is needed, as the following sentence suggests, it is more prudent to perhaps overgenerate by yielding to the original lexicon, especially since this feature only affects the Intermediate Syntactic Representation, not the surface structure: “errors in the energies calculated from the sto 3g basis set were found to decrease with increasing number of solvent molecules.” Alternatively, the strategy of trial three could be continued, of not distinguishing between the two infinitival complements; after all, the original Linguistic String Project grammar conflated the two categories.

The above argument can be made one more time for the distinction OBJTOVO - NTOVO. Not only are the same verbs involved, but there is a direct connection between both pairs: the passive object counterpart of NTOVO is TOVO (e.g., “engineers found the errors in . . . to decrease with . . .”) and that of OBJTOVO is EQTOVO.

Conclusion

This chapter has shown how the category space can successfully be used to induce features like transitivity and subcategorization for clauses and infinitival complements. The grammatical information that is automatically captured by adding appropriate subcategorization frames for new verbs to the lexicon equips the parser for processing text

36 Notice this is one example where the core lexicon distinguishes between the tenses of the verb in assigning features; whereas TOVO (through structure sharing) applies to all conjugated forms of “begin,” “go,” and “mean,” it is associated solely with the passive object options of the past tense of the other seventeen verbs. This indirectly supports the practice in the category space of keeping separate neighborhoods for the different conjugations like “consider, considering, considered,” etc.
in a new domain, thus bypassing the traditional porting bottleneck of tedious and error-prone manual entry.

The data-driven process not only expands the lexicon for new words, it also fits the grammar to the new domain by adjusting the feature set of the existing verbs, adding object options where appropriate to increase coverage and removing them to purge unwanted false positives from the solution space. Given the expansive and ever-changing nature of language, this particular benefit of the automated corpus-based induction process is probably beyond what any human linguist can accomplish manually, no matter the extent of his or her expertise, or the time and effort expended.

An evaluation of the parser with the new lexicon shows that the process does boost performance on both accounts, but also points out that certain additional subcategorization features are needed that are difficult to induce, unless the amount of pre-existing grammatical information is increased. And despite being a lexical grammar, it is also clear that some gaps in PUNDIT’s coverage cannot be “fixed” by augmenting the lexical knowledgebase, but require surgery to the grammar rules themselves.
CHAPTER V
CONCLUSIONS AND FUTURE RESEARCH

Future Research

The category space which lies at the heart of the induction process is slightly different from the one reported in Schütze and Pedersen (1993) in that besides a fixed window, variable distance phrasal boundaries are stored as part of the context digests. More important, however, is that for this dissertation words were disambiguated by part of speech so as not to mix up context information of unrelated tokens, a problem Schütze acknowledges plagues his system (1993b, 254). Several examples were given in chapter IV where the induction process removed a feature from a generic lexicon entry because it only applies to one of the senses of the verb and that sense is not used in this domain (e.g., “represent that,” page 116, or “determine to,” page 123). But when multiple senses of a verb, with different feature assignments, occur in the domain, the induction process is “confused;” for example “follows” is used as “comes after” (“the statement in eq 1 follows where subscript m refers to the mixture”) but also as “to happen as necessary result of” (“with such a restriction the empirical proportionality km x2 follows”). If the training corpus were disambiguated not only by POS, but also by sense, the induction process could render more finetuned assignments. Schütze (1992; 1993a) proposes a scheme to accomplish this by using the category space in combination with a clustering program like AutoClass or Buckshot. The problem with such disambiguated data is that the training corpus tends to become very sparse. However, for the domain of physical chemistry at
least, enough additional raw data is available to compensate for this. And the Singular Value Decomposition too helps to generalize the category space.

Another dataset that could use some boosting is the core lexicon that comes with PUNDIT. It has detailed subcategorization information, but only for 164 verbs. The COMLEX project on the other hand, has constructed a broad-coverage computational lexicon with 38,000 headwords. Although the information was extracted from a host of on-line dictionaries and corpora, one of the main building blocks was Sager’s Linguistic String Project work, so this resource could probably be used in this research with very few changes.

Finally, the induction of subcategorization features that involve prepositions and particles should be explored to augment the lexical entries of new verbs. Without PN(to) and NPN(of), PUNDIT cannot parse the following sentences: “one of the subunits attaches to the other,” and “this approach frees us from thinking of these large systems . . .” Such feature assignment of prepositions and particles tends to be domain specific; for example, the sublanguage of physical chemistry exhibits an unusual affinity between “contribute” and “from,” as in “the dispersion energy is contributed from the correlation effects instead of the coulombic and exchange effects.” Although the category space does not seem the right tool to handle this induction, McKee and Maloney (1992) mention success in this area with another corpus-based method, the mutual association ratio. Alternatively, the use of Dunning’s (1993) log likelihood statistic could be investigated.

**Data-Driven Attuning**

The philosophy behind this work is that Natural Language Processing, in order to be germane to applications involving unrestricted real-world text, needs all the help it can get, and that this is true not only for semantic and pragmatic processing, but also for the supposedly easier aspect of syntactic parsing. This assistance includes of course the
traditional knowledgebase of rules and exception classes, but should also have a corpus-based, or data-driven component to give some direction to the “generic” grammar.

Since the early 90s, there has been a surge of interest in corpus-based NLP research; some researchers have tackled the grammar proper, making it a probabilistic system, or doing away with a rule-based system altogether and inducing a customized grammar from scratch using stochastic methods. Others have left the knowledge-based grammar alone, but have tried a data-driven approach to fit the lexicon and the syntactic knowledge it encodes to the domain at hand. This dissertation falls in the latter category.

Despite the shortcomings of knowledge-based systems described above, it seems wrong to throw away all that has been gained, imperfect as it is. Rather, a hybrid system should be developed where the strengths of both paradigms are combined. A good example of that is a probabilistic Context Free Grammar. Even if the choice was made to induce a new grammar from scratch, supervised training, where the training corpus is pre-processed, often yields better results and converges more quickly than unsupervised methods.¹

Both Brent (1993) and Manning (1993), who attempt to induce a lexicon do so by completely discarding all pre-existing knowledge;² both systems are stand-alone, without a parsing engine to test or use the “learned” information.³ Brent in fact takes the “from scratch” to an extreme, and models his system after the way a child learns to understand

¹Such preprocessing (e.g., bracketing) can be done with another “system” and then manually corrected. The Penn Treebank, for example, was created using Hindle’s (1983) Fidditch parser. Stochastic parsers are often ported that way to a new domain because correcting by hand is much easier than entering new tags on virgin text. Brill’s (1992) tagger, which was used for the AIMS project was bootstrapped from a tagger which was developed inhouse and which was trained on a different domain of veterinary medicine.

²Manning did pre-process his system with Kupiec’s (1992) stochastic part-of-speech tagger, but like Brent he does not build on an existing lexicon.

³McKee and Maloney (1992) report on preliminary work on inducing a lexicon using a pattern matching method similar to Brent’s and Manning’s; they do plan to integrate this information into a complete parsing system.
language. The algorithm of both authors basically involves a pattern matcher that scans the input for a verb, and once an anchor is found, its right context is searched for cues for subcategorization frames. Brent’s cues are very primitive, but because he only picks up frames when the indicators are unambiguous, his results are very reliable, albeit sparse (unless a very large training corpus is used). Manning’s triggers on the other hand are more sophisticated, but because they are less dependable he must rely on statistical filtering to reduce the “noise.” Although Manning’s work in inducing features certainly accomplishes the goal of customizing the lexicon to a particular domain, the porting process is still very much a manual enterprise in that he must write a mini-parser, a finite state machine that includes an NP recognizer “and various other rules to recognize certain cases that appear frequently” (1993, 237).

The dilemma of any pattern matching approach is in essence a bootstrapping problem; if the goal is to induce syntactic information (in the form of lexical features), then paradoxically some heavy syntactic processing power is needed to “parse” the training data to mine for evidence that a particular verb subcategorizes for an object option, while avoiding false triggers (imposter patterns). Manning has built into his finite state device a panic mode to skip over ambiguous elements (cf. Brent’s Learner), but the trick is to recognize when things get hairy; that is where a lot of programming effort takes place, and this finetuning is never over (and must be repeated for every port to a new domain) as Manning himself admits: “there are many . . . things that the parser does wrong or does not notice . . . . One could continue to refine the parser” (1993, 238).

---

4For example, in the physical chemistry domain, “determine” does not subcategorize for INF: despite “evidence” like “to examine the role, if any, that charge transfer plays in these systems, it is necessary to determine to within a few tenths of an electronvolt the ionization potentials of the SUBST oxides involved.” More examples of such difficult false triggers can be found on page 104 (“explain that”) and 105 (“needed to”).
Induction Without Pattern Matching

The category space described in this dissertation uses a very different approach to induce subcategorization frames; instead of starting from scratch, the method described in this document exploits the existing rich lexicon and assigns features to new words based on their paradigmatic relatedness to known words. Thus instead of having to “hunt” for evidence, this approach is able to exploit the expertise of seasoned linguists who constructed the initial lexicon, which was intentionally designed to be broad-coverage. Such a strategy not only avoids having to distinguish good cues from irrelevant triggers, but is capable of inducing some features like ASSERTION for which there is no marker that would indicate its presence.

The category space is the arbiter of paradigmatic relatedness, and since it is bootstrapped from a training corpus that is representative for the domain sublanguage, the resulting lexical entries will be customized for that domain. Porting the lexicon to a new domain is as simple as bootstrapping another category space.

The existing verbs in the lexicon themselves undergo a similar process whereby they are fitted to the domain: some of their “generic” features which are not appropriate are dropped, whereas “gaps” in object options are filled. The net result is that the grammar becomes attuned to the sublanguage: parses become possible because the enabling features are present, while the search space is pruned of many false positives because unnecessary features are omitted.

The system was tested by comparing the number of successful parses of a held-out test corpus before and after customizing the lexicon. A direct comparison with Brent’s and Manning’s system is not possible because their lexicons are not tied to a parsing system.
Although he expresses reservations about their indicator value, Manning does give some recall/precision figures, which are lower than the ones reported here.5

Knowledge-Based Backbone

The grammar knowledgebase is the product of a never-ending attempt by linguists to impose order on something that refuses to be pinned down because it is a living thing, a moving target. To a great extent, of course, these linguists are able to point to regularities, because language is first of all a practical thing, a means to communicate, and there must be a common base for such transfer to take place. But all rules have exceptions, and often it turns out these exceptions are not isolated or random, so the rule is finetuned, or a secondary directive is issued. The problem is that what is “grammatical” depends on whom you ask, or on the unwritten rules of a certain domain. When the core grammar is augmented to accommodate all these idiosyncracies, the danger is not that an ungrammatical sentence might slip through, but that perfectly legitimate input receives an incorrect analysis that is sanctioned by some peripheral grammar rule that doesn’t apply to the domain under investigation. The semantic component which gets this false positive may reject it and request a second reading, and the correct parse will most probably come down the pipeline eventually if the grammar is truly broad-coverage, but a semantic module is not always well equipped to detect such errors and may have a difficult time enough trying to resolve

5Manning emphasizes precision over recall whereas the induction algorithm in this dissertation favors recall. However, by using different values for the skewing threshold (3) and adoption point (-0.03), precision can be given preference too, which would yield precision=0.92 and recall=0.83, still better than his returns. Grimshaw et al. who are (manually) constructing a wide-coverage computational lexicon sides with this dissertation in choosing recall over precision: “in some cases this approach will provide a preposition list that is a little rich for a given verb but we have decided to err on the side of slight overgeneration rather than to risk missing any prepositions that actually occur” (1994, 4).

“Precision” is not a good yardstick for evaluating the performance of the induction process, because it measures the outcome against a “flawed” lexicon; the induced features, because of the data-driven nature of the process, are more “precise” when measured against the “real world” of the sublanguage domain than the hand-built entries that are the product mostly of introspection and anecdotal evidence.
attachment problems, anaphoric references, etc., even when presented with the “right” parse.\(^6\)

In systems that use a lexical grammar, i.e., where part of the grammatical “knowledge” is stored outside the non-terminals of the grammar proper, using subcategorization frames associated with terminals (words in the lexicon), the peril likewise is that this resource becomes bloated over time with options exercised only in certain settings or when the word is used in a marginal sense.

Clearly something must be done to separate the wheat from the chaff; the problem is twofold: getting the grammar and lexicon to a certain level of competence was a laborious and time-consuming process, and undoing this (i.e., eliminating unwanted options) is almost as difficult and painful as the constant augmenting in the first place. And secondly, what constitutes wheat and chaff is different for each domain, so this “dieting” must be repeated for every port.

Corpus-based techniques can help automate this filtering, i.e., the source text should be viewed not only as an “obstacle” to be tamed (parsed), but as a resource that is best authority on what is grammatical for the domain.

\(^6\)Usually what happens is that the first parse returned is reasonable and non-controversial; i.e., the rulebase is somewhat prioritized to try “orthodox” rules first. But when an analysis is rejected on semantic grounds, the system systematically backtracks to the latest choice point, and may find some “acceptable” sequence that involves more esoteric rules.
APPENDIX A

LSP FEATURES
Appendix A briefly describes the subset of the Linguistic String Project features that are mentioned in this dissertation. The information is culled from Sager 1981 and Linebarger 1988. For each subcategorization feature, a frame is given that describes the environment in which the word occurs, followed by an example. A brief explanation of the feature is then given, often by contrasting the feature with others or by listing an example of ungrammatical strings. When a feature is followed by “[ ]” this implies that the lexicon must specify for each verb or noun the particular adjectives, prepositions, etc. that apply.

**ASOBJBE**

SUBJ tV “as” N

“Enzymes function as catalysts.”

Object option N of the tensed verb is paraphrasable as “in the capacity of.”

Should be distinguished from the adjunct sequence “as” + NSTGO, paraphrasable as “while”: “he lived abroad as a teenager.”

**ASSERTION**

SUBJ tV [“that”] S

“He believes John was a witness.”

“that” is optional; subset of THATS.

**ASTG [ ]**

SUBJ tV ADJ

“That story rings true.”

The particular adjectives that can occur in adjective position must be specified in the lexicon for each verb.

Should be distinguished from the class OBJBE which has no such restriction: “she remains true - she remains red - she remains sick.”

**C1SHOULD**

SUBJ tV “that” SN

“I demand that he come.”

The verb in the embedded sentence SN is not tensed; can also occur with “should.”

**DP1 [ ]**

N tV DP

“They carried on.”

The DP (particle) cannot be analyzed as a adverbial adjunct: “*They carried.”

The particular DPs must be specified in the lexicon for each verb.

**DP2 [ ]**

SUBJ tV DP N

“He looked up the number.”

N (which cannot be a pronoun) can permute with DP (particle): “he looked the number up.”

The particular DPs must be specified in the lexicon for each verb.

Should be distinguished from a prepositional phrase (PN) which does not permute: “*he looked the shaft up.”
EQTOVO

SUBJ tV “to” V OBJ

“The field engineer wants to repair the disk drive.”

The subject (SUBJ) of the matrix verb tV is also subject of the infinitive V.

NCOUNT1

“A[n]” N tV

“A book fell.”

Countable nouns; “*book fell.”

Nouns not marked with NCOUNT1 are either mass or abstract nouns: “*a blood flows.”

NCOUNT2 [ ]

PREP N

“He came by car.”

Subset of NCOUNT1 which can occur as object of a specified preposition without an article.

The particular PREP must be specified in the lexicon for each noun.

NHUMAN

N N

“She bought the boy a book.”

or N “who[m]”

“The man who ate the cheese left.”

N can occur as indirect object or as host of a relative clause headed by “who[m].”

NOTNOBJ [ ]

e.g. NOTNOBJ:NHUMAN

“*The government abolished the boy.”

Verbs restricted in terms of noun objects with which they can occur in scientific writing.

The particular NOTNOBJ attributes must be specified in the lexicon for each verb.

NOTNSUBJ
e.g. NOTNSUBJ:NHUMAN

“*The gentleman occurred.”

Verbs restricted in terms of noun subjects with which they can occur in scientific writing.

The particular NOTNSUBJ attributes must be specified in the lexicon for each verb.

NPN [ ] - PNN [ ]

SUBJ tV NOUN PREP NOUN

“They attributed the painting to Rubens” or

“They liberated the city from the enemy.”

Either the complete NPN object is required “*they attributed the painting”
or there is a strong selectional dependency between the verb and PREP: “*they liberated the city to the enemy.”

Should be distinguished from the sequence N + adjunct (PREP N): “they liberated the city on Sunday.”

The particular PREP must be specified in the lexicon for each noun.

PNN is simply a permutation of NPN: “he gave to her the book he just bought.” The lexicon should always specify both features together.

NSENT1 [ ]

N SN tV OBJ

“The fact that he left surprised me”

or N be SN

“The fact is that no one wants to come.”

The noun can occur with an embedded sentence SN as a right adjunct, or with a predicate + SN, but not both: “*the fact that he left is that no one wants to come.”
Should be distinguished from relative clauses (“the book that he wrote...”) , which becomes clear when the verb of the embedded sentence is intransitive: “*the book that the atom exists is clear.”

The type of SN string must be specified in the lexicon for each noun: ATHAT (“the fact that...”) AFORTO (“his attempts to leave...”) ASHOULD (“the demand that salaries be raised...”) or AWH (“the question whether to vote...”).

NSENT3
N be SN “The trouble is that we have no money.”
Subset of NSENT1 which can only occur in the second environment: “*the trouble that we have no money bothers us.”

NSNWH
SUBJ tV N SNWH “I asked him why he did it.”
N must be NHUMAN

NSTGO
SUBJ tV N “He amassed a fortune.”
Transitive verbs, including those that can drop the object (“he reads [a book]”) and verbs marked for NPN where PN is droppable (“he fastened the chain [to the door]”).

NTHATS
SUBJ tV N [“that”] SN “I told you that he came.”
N must be NHUMAN

NTIME1
ADJ N S or S ADJ N “Last week, we met in Chicago.” or “We’ll finish the work next time.”
N ADV S or S N ADV “An hour hence, the place will be deserted.” or “I saw him two days ago.”

The noun cannot occur alone as a sentence adjunct, but must itself have a left adjunct that is an adjectival timetag or a right adjunct that is an adverbial timetag.

NTOVO
SUBJ tV N [“to”] V “They allowed him to go.”
Should be distinguished from the use “in order to:” “he drinks milk to keep thin.”

NULLOBJ
SUBJ tV “The book has disappeared.”
Intransitive verbs, including those where the object or PN is droppable (“he reads”).

OBJBE [ ]
SUBJ tV NSTG “He seemed a happy man.”
SUBJ tV ASTG “He looks satisfied.”
SUBJ tV DSTG “He feels down.”
SUBJ tV PREP N “This appears of great significance.”
The particular attribute (NSTG, ASTG, DSTG, PREP N) must be specified in the lexicon for each verb.
Verbs which occur with a limited set of adjectives (“ring true”) should be marked as ASTG rather than OBJBE.
Verbs which occur with a wide range of adverbs (“he came here”) should be not marked as OBJBE.

OBJTOVO
SUBJ tV N “to” V OBJ
“The factory told the field engineer to repair the pump.”
N is both object of the matrix verb tV and subject of the infinitive V.

PN [ ]
SUBJ tV PREP N
“It consists of protein.”
The verb exhibits a strong selectional dependency with the preposition, and generally does not occur with a zeroed object.
Verbs marked for NPN where the leftmost N can be dropped should also be included:
“he gives [money] to charity.”
The particular prepositions must be specified in the lexicon for each verb.

SNWH
SUBJ tV WH ASSERTION “I asked whether I should go.”
WH also includes “if” and “how.”
A preposition can occur at the beginning or end of the SNWH string: “I don’t know from whom he obtained the information.” This should be distinguished from PN which is dependent on the container verb and whose preposition cannot occur at the end: “*I wonder whether to go about.”

SOBJBE [ ]
SUBJ tV N OBJBE
“They consider him their savior.”
OBJBE is the predicate of N.
The particular attribute must be specified in the lexicon for each verb: NSTG, ASTG (“we thought him interesting”), DSTG (“we prefer a meeting today”), or PREP N (“they find it of slight interest”).

SVEN
SUBJ tV N Ven
“He wishes the papers destroyed.”
Should be distinguished from N + adjunct (“I delivered the papers duplicated”), as the Ven cannot be dropped: “*he wishes the papers.”
Should be distinguished from SOBJBE (“I consider the house well-designed”); the latter is more adjectival in nature and can take an adverbial left adjunct like “very.”

SVO
SUBJ tV N V [OBJ]
“I made him sign a statement.”
Should be distinguished from C1SHOULD (“I suggest he sign.”)

THATS
SUBJ tV “that” S
“He added that they were happy.”
The verb of the embedded sentence S is tensed.
Also includes verbs where “that” is optional.

TOVO
SUBJ tV “to” V
“John tried to go.”
Aspectual verbs without overt subject in the contained sentence.
VINGOFN
  SUBJ tV N’s [ADJ] Ving “of” OBJ “They imitated his singing of the song.”
  SUBJ tV DET [ADJ] Ving “of” OBJ “They imitated the singing of the song.”

VMOD
  SN tV NSENT3 [PREP N] “That John was here presents a problem for us.”
  SN tV N NSENT3 “For him to give up now would give us trouble.”

VSENT4 [ ]
  “it” tV SN “It seems that he left.”
  Cannot occur in the environment SN tV; “*that he left seems.”
  The particular SN (ASSERTION, THATS, TOVO) must be specified in the lexicon for each verb.
APPENDIX B
LSP TERM GLOSSARY
Appendix B gives a brief overview of string grammar terms, adapted from Hirschman and Dowding (1990) and Ball et al. (1992). The string grammar was originally formulated by Harris (1962), and implemented by Sager and co-workers (1981). The string grammar differentiates between two types of constructs:

- **endocentric or \(1\times r\) constructions**, which consist of a head flanked by left and right modifiers; the behavior of such constructs is governed by the head: a nounphrase, for example, is noun-like in behavior.

- **exocentric constructions or strings**, whose behavior is different from their constituents. A prepositional phrase, for example, is neither preposition- or noun-like. Strings consist of mandatory and optional adjuncts.

Parse trees always include a slot for each element, even if its realization is null; this makes for bushy but very regular parse skeletons.

Following is an example of a string rule:

```
assertion ::= sa, subject, sa, ltvr, sa, object, sa.
```

Below are some definitions of endocentric constructs.

\(1\times r\) a left-adjunct + \(\times\) + right-adjunct construction, where \(\times\) can be:

- **n** noun
- **a** adjective
- **v** verb
- **ven** past participle
- **tv** tensed verb
- **ving** present participle
- **q** quantity word
- **pro** pronoun

**nvar** head noun in \(ln\)

\(ln\) left-adjunct in \(ln\)

\(rn\) right-adjunct in \(ln\)

**npos** noun position in prenominal \(ln\)

**apos** adjective position in prenominal \(ln\)

**qpos** quantity position in prenominal \(ln\)
<table>
<thead>
<tr>
<th>tpos</th>
<th>determiner; prenominal position</th>
</tr>
</thead>
<tbody>
<tr>
<td>namestg</td>
<td>nominal, such as date - time- money expression</td>
</tr>
<tr>
<td>nstg</td>
<td>noun string</td>
</tr>
<tr>
<td>sa</td>
<td>sentence adjunct</td>
</tr>
<tr>
<td>sa_init</td>
<td>sentence adjunct in initial position</td>
</tr>
<tr>
<td>sa_final</td>
<td>sentence adjunct in final position</td>
</tr>
<tr>
<td>dstg</td>
<td>adverb string (d is adverb)</td>
</tr>
<tr>
<td>vingo</td>
<td>present participle + object</td>
</tr>
<tr>
<td>venpass</td>
<td>past participle + passive</td>
</tr>
<tr>
<td>passobj</td>
<td>passive object</td>
</tr>
<tr>
<td>nullwh</td>
<td>wh gap (null element corresponding to moved wh-expression)</td>
</tr>
<tr>
<td>nn</td>
<td>double dative object</td>
</tr>
<tr>
<td>conj_wd</td>
<td>conjunction word</td>
</tr>
<tr>
<td>spword</td>
<td>special (conjunction) word</td>
</tr>
<tr>
<td>vo</td>
<td>untensed verb + its object</td>
</tr>
<tr>
<td>svo</td>
<td>untensed clause</td>
</tr>
</tbody>
</table>
APPENDIX C
AIMS TAGS
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AUX</td>
<td>Verb, auxiliary</td>
</tr>
<tr>
<td>2</td>
<td>CC</td>
<td>Coordinating conjunction</td>
</tr>
<tr>
<td>3</td>
<td>CD</td>
<td>Cardinal number</td>
</tr>
<tr>
<td>4</td>
<td>DT</td>
<td>Determiner</td>
</tr>
<tr>
<td>5</td>
<td>EX</td>
<td>Existential &quot;there&quot;</td>
</tr>
<tr>
<td>6</td>
<td>FW</td>
<td>Foreign word</td>
</tr>
<tr>
<td>7</td>
<td>GER</td>
<td>Verb, gerund</td>
</tr>
<tr>
<td>8</td>
<td>IN</td>
<td>Preposition</td>
</tr>
<tr>
<td>9</td>
<td>JJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>10</td>
<td>JJR</td>
<td>Adjective, comparative</td>
</tr>
<tr>
<td>11</td>
<td>JJS</td>
<td>Adjective, superlative</td>
</tr>
<tr>
<td>12</td>
<td>LS</td>
<td>List item marker</td>
</tr>
<tr>
<td>13</td>
<td>MD</td>
<td>Modal</td>
</tr>
<tr>
<td>14</td>
<td>NN</td>
<td>Noun, singular or mass</td>
</tr>
<tr>
<td>15</td>
<td>NNS</td>
<td>Noun, plural</td>
</tr>
<tr>
<td>16</td>
<td>NP</td>
<td>Proper noun, singular</td>
</tr>
<tr>
<td>17</td>
<td>NPS</td>
<td>Proper noun, plural</td>
</tr>
<tr>
<td>18</td>
<td>PDT</td>
<td>Predeterminer</td>
</tr>
<tr>
<td>19</td>
<td>POS</td>
<td>Possessive ending</td>
</tr>
<tr>
<td>20</td>
<td>PP</td>
<td>Personal pronoun</td>
</tr>
<tr>
<td>21</td>
<td>PP$</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>22</td>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>23</td>
<td>RBR</td>
<td>Adverb, comparative</td>
</tr>
<tr>
<td>24</td>
<td>RBS</td>
<td>Adverb, superlative</td>
</tr>
<tr>
<td>25</td>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>26</td>
<td>SC</td>
<td>Subordinating conjunction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>27</td>
<td>SYM</td>
<td>Symbol</td>
</tr>
<tr>
<td>28</td>
<td>TO</td>
<td>&quot;to&quot;</td>
</tr>
<tr>
<td>29</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>30</td>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>31</td>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>32</td>
<td>VBG</td>
<td>Verb, present participle</td>
</tr>
<tr>
<td>33</td>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>34</td>
<td>VBP</td>
<td>Verb, non-3rd person singular present</td>
</tr>
<tr>
<td>35</td>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>36</td>
<td>WDT</td>
<td>Wh-determiner</td>
</tr>
<tr>
<td>37</td>
<td>WP</td>
<td>Wh-pronoun</td>
</tr>
<tr>
<td>38</td>
<td>WP$</td>
<td>Possessive wh-pronoun</td>
</tr>
<tr>
<td>39</td>
<td>WRB</td>
<td>Wh-adverb</td>
</tr>
</tbody>
</table>
REFERENCES


Hatzivassilouglou, Vasileios. 1994. Do we need linguistics when we have statistics? A comparative analysis of the contributions of linguistic cues to a statistical word grouping system. In The balancing act: Combining symbolic and statistical approaches to language: Proceedings of the ACL workshop held in Las Cruces, N.M., July, 1994, by the Association for Computational Linguistics. 43-52.


Kochman, F., and J. Kupin. 1991. Calculating the probability of a partial parse of a
sentence. In Proceedings of the speech and natural language workshop held in Pacific

Kucera, H., and W. Francis. 1967. Computational analysis of present-day American

Speech and Language 6: 225-42.

Lang, François, and Lynette Hirschman. 1988. Improved portability and parsing through
interactive acquisition of semantic information. In Proceedings of the second confer-
ce on applied natural language processing held in Austin, TX, February 1988, by the
Association for Computational Linguistics.

Sublanguage: Studies of language in restricted semantic domains, ed. R. Kittredge and
J. Lehrberger, 81-106. New York, N.Y.: de Gruyter.

Liberman, M. 1991. Guidelines for the linguistic data consortium, draft proposal for
DARPA.


Macleod Catherine, Ralph Grishman, and Adam Meyers. 1994. Creating a common
syntactic dictionary of English. In Proceedings of the international workshop on
sharable natural language resources held in Nara, 1994.


diss., Stanford University.

Magerman, David, and M. Marcus. 1990. Parsing a natural language using mutual
information statistics. In Proceedings of the ninth national conference on artificial
intelligence held in Boston, MA, July, 1990, by the American Association for Artificial
Intelligence. The AAAI Press, 984-89.

Magerman, David, and M. Marcus. 1991. Pearl : A probabilistic chart parser. In Proceed-
ings of the fifth EACL Conference held in Berlin, Germany, April 1991, by the
Association for Computational Linguistics. 15-20.

Manning, Christopher. 1993. Automatic acquisition of a large subcategorization dictionary
from corpora. In Proceedings of the thirty-first annual meeting of the ACL held in
Columbus, OH, June, 1993, by the Association for Computational Linguistics. 235-42.


Mosteller, Frederick, and David Wallace. 1964. *Inference and disputed authorship: The federalist*. Addison-Wesley.


held in Las Cruces, N.M., June, 1994, by the Association for Computational Linguistics. 181-87.


