Guiding a Linguistically Well-Founded Parser with Head Patterns

Amon B. Seagull Lenhart K. Schubert

The University of Rochester
Computer Science Department
Rochester, New York 14627

Technical Report 767

December 2001

Abstract

We describe a parser that draws from both extant corpora and linguistic knowledge sources, and thus is suitable as a front end for applications requiring both broad coverage and rich syntactic analysis. We detail many of the difficulties and assumptions involved in combining these data and knowledge sources. We also describe the novel language model that we use for disambiguation and show that it outperforms a comparable model without the same knowledge sources.

We are very grateful to Phil Harrison and The Boeing Company for supplying SAPIR and answering so many questions about it, as well as for helpful suggestions on this project. This work was supported in part by NSF research grants IRI-9623665 and IRI-9711009, DARPA research grant F30602-98-2-0133, and U.S. Air Force/Rome Labs research contract F30602-95-1-0025.

1 Introduction

Statistical approaches to problems in Natural Language Processing have received a great deal of attention in recent years (*cf.* Mooney [1999]), and with good reason. They immediately provide scalability and robustness, two desiderata wanting in NLP systems of the 1970s and 1980s. Syntactic disambiguation is a prime example: in under ten years—since the initial release of the Penn Treebank [Marcus *et al.*, 1993]—researchers have used purely statistical, data-driven methods to quickly reduce error rates on standard metrics by 25%. In fact, by some accounts, these recent advancements in parsing have rendered it "solved" [Schwartz, 1999].

But even taking the literature-best figures of 88% accuracy for labeled constituents [Collins, 1999] as good enough to have "solved" parsing, what problem exactly has been solved? Traditionally, parsing in NLP has had as its goal the construction of a semantic representation, *by way of* syntax. Thus a parser should either output a semantic form, or construct a syntactic analysis from which a semantic form is immediately derivable. Not that parsers for *all* applications need to meet this criterion. Certainly, various NLP applications, *e.g.*, information retrieval, may benefit from "shallow" parsing. But for Natural Language *Understanding*, we claim that a parser should meet these traditional requirements.

Researchers have had much success in recent years in the automatic reconstruction of parse trees of the sort annotated in the Penn Treebank [Magerman, 1995; Collins, 1996; Charniak, 1997a; Ratnaparkhi, 1997]. But do these trees readily admit semantic interpretation? The trees tend to have a very flat structure; base `NP`s are famously left unstructured. Section 00 alone of the Penn Treebank's annotation of the Wall Street Journal corpus has 730 different expansions for `NP`s, ignoring recursive expansions and quotation marks. In their article discussing the treebank, Marcus *et al.* [1993] repeatedly reiterate that the bracketing is *skeletal*, and that sacrifices in syntactic detail were made in order to speed up the annotation process. Parsers that succeed at constructing these skeletal bracketings will probably be most useful as replacements for annotators, thus providing huge amounts of coarsely annotated syntactic data for other NLP projects (*cf.* Caraballo [1999]).

We claim that these parsers cannot function as parsers in the traditional sense; while they meet the desiderata of robustness and scalability, they fail to satisfy the desideratum of usability, at *least* for language understanding applications. Our goal is to combine the data in the treebank with linguistic knowledge to create a syntactically rich corpus, from which we can create a statistical parser that is both scalable and usable.

2 Treebank Parsers and Interpretability

Before discussing our work, it may be necessary to explain why a parser that can learn the trees in the Penn Treebank will not suffice as a parser for an NLU application. Note we are not claiming that a purely statistical method *cannot* be a useful front end; Miller *et al.* [1996] describe a purely data-driven model that learns to translate input sentences to syntactic trees to logical forms to contextually de-referenced representations for querying a database. Bod *et al.* [1996] describe the Data-Oriented Parsing model, which they use to similarly learn to construct semantic representations of sentences by starting with a labeled corpus.

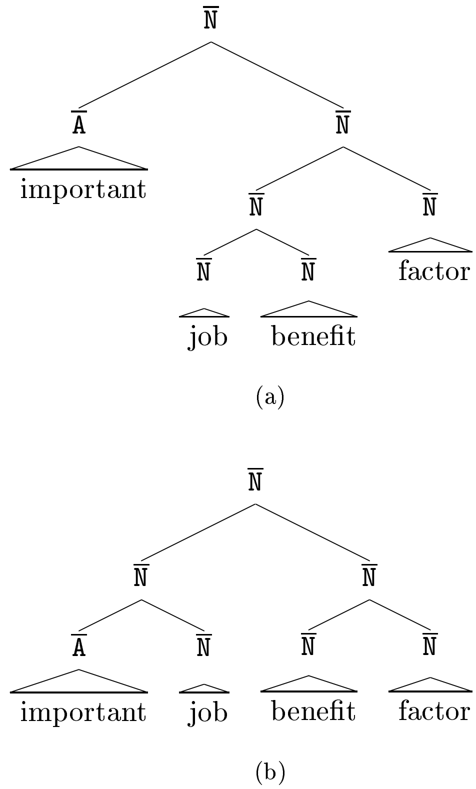

Figure 1: Two of the several parses for 'important job benefit factor'

A parser constructs syntactic representations for the purposes of calculating meanings. As mentioned above, the Penn Treebank does not detail the finer syntactic structure of nominal compounds. Thus a phrase such as 'an important job benefit factor' (from section `e30` of the Brown Corpus) is not disambiguated as to whether this is an important factor concerning job benefits, or a factor concerning benefits of an important job (or other readings). Obviously such a distinction is crucial to meaning; linguists have taken the meaning distinction to follow from the different syntactic structures drawn in Figure 1. Since this level of structure is not detailed in the Penn Treebank, a parser that learns to reproduce treebank trees will not be able to disambiguate that input.

Another difficulty with treating Treebank-style trees as output from a parser is indirectly brought up by Charniak [1997b] when he notes that the phrase 'the dog biscuits' is parsable not only as a nominal compound, but also (surprisingly) by the rule $\mathtt{NP} \longrightarrow \mathtt{NP}\ \mathtt{NP}$ (the rules are lifted directly from the trees in the corpus). Looking at where this labeling appears in the WSJ treebank, he concludes that the rule is intended to cover "measure" noun phrases, as in '5 dollars a share.' But just such an interpretation is nonsensical for 'the dog biscuits,' and Charniak suggests that the only interpretation that would follow from the `NP NP` constituency is an appositive one, where the dog is named 'biscuits.'

The problem here is twofold: one issue is that one syntactic structure seems to felicitously yield two distinct meanings. One might conclude that in fact there are *two distinct* NP $\longrightarrow$ NP NP rules, one licensing appositives, and one for measure noun phrases. Under that hypothesis, a parser cannot simply label the constituent structure and have fully defined a parse; the actual rules used to license the expansions must be identified. In the absence of a mechanism to label rules distinctly from the constituent structures they license, such distinctions in syntax (and thus meaning) cannot be accomplished by the parser.

But there is another problem; 'the dog biscuits' does not seem to be ambiguous between an appositive construction and a measure phrase (which is what the above hypothesis predicts). There is no measure reading of it. Presumably this is because 'the dog biscuits' is a *syntactically* ill-formed measure noun phrase. The first NP needs to have some sort of `numeral` feature, and the second needs to be interpretable as a unit, if they are to form a measure NP. Thus 'the dog biscuits', 'Bill a dog', and 'five dollars the share' are all syntactically ill-formed measure NPs. Not only do we need two NP $\longrightarrow$ NP NP rules to accommodate different semantics, we need two rules because the two structures have different *syntactic* requirements (beyond their labeled constituency).

Given these ambiguities, we propose to accept the Penn Treebank as its creators intended it: as a *skeletal* structure from which more detailed data can be extracted when linguistic knowledge is used to interpret the structures. Charniak [1997b, p. 42] anticipates this when he calls for researchers "to move beyond [treebank parsers], using the tree bank as a first step."

In fact, almost all research using the Penn Treebank to some extent incorporates linguistic assumptions into the work: deciding how to decompose a parse tree into its constituent pieces presumes some linguistic underpinnings (*e.g.*, Charniak [1997a] supposes they are licensed by a context free grammar, and Collins [1996; 1997] instead treats them as sequences of head-daughter attachment decisions). Any lexicalization of the treebank requires some assumption about the headedness of structures; most researchers use some variant of the rules constructed by Magerman [1995]. These rules designate a single word in any phrase to best represent that phrase's meaning. *E.g.*, 'important job benefit factor' is "represented" lexically by the single word 'factor'. These rules essentially impute structure to the treebank that is not explicitly labeled.

Collins [1997; 1999] goes one step further in imputing structure to the treebank. He distinguishes argument from adjunct prepositional phrases, a distinction that was not always made in the Penn Treebank for reasons of efficiency. Thus his parser is able to do a better job at language modeling and thus perform better, and also create more useful output (presumably a parser should indeed distinguish argument from adjunct phrases).

The Verbmobil project [Wahlster, 2000] employs, among a collection of parsers, one which is based on a treebank of data (similar to the Penn Treebank parsers just mentioned). As the Verbmobil project was to build a machine translation system, a richer representation than what was facilitated by that corpus-based parser was required. Schiehlen [2000] describes the approach that was taken to actually build a semantic representation from those relatively impoverished syntactic bracketings; the approach makes heavy use of a lexico-semantic database constructed for the project [Kasper, 2000].

In this paper we describe our framework, where we have fully reinterpreted trees in the Penn Treebank in terms of a linguistically motivated grammar and lexicon. Thus our parser outputs more than trees; it outputs syntactic interpretations. We also show that incorporating additional knowledge, such as an ontology, improves parsing performance.

3 Model: Head Patterns

Our goal is to augment a well-founded (linguistically plausible) lexicon and grammar with well-founded (corpus-based) probabilities. Our probability terms are derived directly from our model, a straightforward lexicalization of probabilistic context free grammars. In our model, rather than generating syntactic categories and then generating syntactic expansions based on those categories, we generate syntactic categories *and lexical heads* and then generate syntactic expansions *with lexical heads*. More formally, the random process by which we take sentences to be generated is as follows:

1. A root category X and head word (or word *sense*; see Section 4) w_X are generated from the set $P \times L$, where P is the set of phrasal categories and L is the lexicon.

2. Based on $\langle X, w_X \rangle$, generate a syntactic expansion $Y_1 \ldots Y_n$, each term of which has a lexical head, $h_1, \ldots, h_n$. The expansion $X \rightarrow Y_1 \ldots Y_n$ is licensed by a particular rule from the set of phrase structure rules R. For each pair $\langle Y_i, h_i \rangle$ where Y_i is not a lexical category, repeat this step setting $\langle X, w_X \rangle = \langle Y_i, h_i \rangle$.

This process generates sentences as sequences of words (in base form) along with syntactic structures; we take sentences to be generated independently of one another. Thus to find the sequence of syntactic structures that is most likely for a sequence of test sentences, we can apply this model and find the structure that is most likely for each sentence, independently.[1]

Given this model, we can take the probability for a parse tree T to be the product of all its expansions:

$$P(T) = \prod_{expansion \in T} P(\vec{Y}, \vec{h}, rulename \mid X, w) \tag{1}$$

where X and w are the syntactic category and head word of the mother node, and Y_i and h_i are the syntactic category and head word of the ith daughter. *rulename* is the identifier for the rule that licenses the expansion. Note that *rulename* usually determines X, and *rulename* and X together always determine $\vec{Y}$.[2] Also note that each tree is assumed to be rooted at a dummy `UTT` node (with a dummy `WORD` head word), which serves as the parent for the "true root" of the tree. (This enables us to write the probability of T uniformly as a product over expansions, rather than as a product over expansions multiplied by an initial probability corresponding to step 1 of the stochastic process.)

[1] Note this assumption seems patently incorrect, but nonetheless is standard in the statistical parsing literature. Furthermore, the nature of our corpus is such that its contiguous sentences were not necessarily contiguous in the original natural language text.

[2] *rulename* does not determine the precise syntax of an expansion when the rule is, for example, a coordination rule ($XP \rightarrow XP$ `coord` XP).

To calculate the probability of the individual terms of the product in Equation 1, we apply the chain rule and expand into a *syntactic expansion* probability (term 2) and a *lexical introduction* probability (term 3). Note that we can drop the $\vec{Y}$ term as mentioned above.

$$P(\vec{Y}, \vec{h}, rulename \mid X, w) = P(rulename \mid X, w) \quad (2)$$
$$\times P(\vec{h} \mid X, w, rulename) \quad (3)$$

The first of these probabilities corresponds to the psycholinguistic notion of Lexical Preference [Ford *et al.*, 1982] when the expansion is of a verb phrase to its subcategorized constituents. The lexical introduction probability is novel in that it requires us to collect n-ary lexical statistics, where n is as large as the grammar requires (essentially the length of the longest right-hand-side of a rule). While the models mentioned above typically make the assumption that the h_i are generated independently of one another, we wish to preserve this dependency among sisters in the parse tree.

Researchers interested in semantic interpretation have investigated methods of automatically extracting syntactic-semantic *case frames* for verbs (*e.g.*, [Utsuro and Matsumoto, 1997b]). These case frames generally include all arguments of the verb, and thus can account for dependence among these arguments. However the models can be difficult to incorporate into a parser, as the frames have larger reach than a context-free grammar rule will allow. Modeling dependence among sisters in a parse tree will give us some of the informational purchase of a case frame, but restricting ourselves to *syntactic* sisters allows us to employ a syntactic language model in the parser.

Dependence among sisters will most often be a factor in calculating probabilities of certain verb phrases, as many verbs frequently appear with multiple complements. For example, the verb 'place' is often used with an `NP` and `PP` complement. If the `NP` complement is something like 'advertisement', we would expect with high probability the preposition of the `PP` to be 'in', to support such phrases as 'place an ad in the paper'. However if we were to consider only binary lexical patterns, we would evaluate the salience of 'in' as the preposition for 'place' *regardless* of the `NP` argument. Under this assumption, the model may not prefer 'in' to 'on' or any another generic locative preposition.

4 Our Framework

As our goal is to derive probabilities from a corpus to guide a handwritten grammar, we require such a grammar, as well as a corpus annotated with its grammar rules. Further, given our language model, we will need to extract tuples of heads along with those grammar rules to estimate the probabilities in terms (2) and (3), above.

Data sparsity is always a challenge in statistical natural language processing, and we can see from looking at term (3) that it will be an issue here. Estimating the probability of an n-tuple "head pattern" given a syntactic rule yields a huge space of possible events: $|R| \times |L|^n$ instead of $|R| \times |L|^2$, as we would have if we modeled the head words of sisters in a parse tree as independent.[3] To combat sparsity, we collect statistics of both lexical items in syntactic constructs, and *abstractions* of lexical items in syntactic constructs.

To enable the calculation of abstractions of lexical items, we need a mapping from words to their abstractions. WordNet provides such a mapping (and more). WordNet [Fellbaum, 1998] is a psycholinguistic project developed at Princeton University. A comprehensive ontological and lexical resource, version 1.6 organizes $129,505$ words (distinguishing part of speech) into $99,368$ concepts. Each concept is identified by a collection of synonymous words. As many words have multiple meanings, the concept is actually a set of synonymous word *senses*, and is known as a *synset*. WordNet1.6 has $173,941$ word senses. The synsets are related by a number of semantic pointers, such as meronymy (part-whole relationships) and entailment. But it is the hyponymy (subtype-supertype) and hyperonymy (supertype-subtype) relationships that we use here.

To enable the supervised learning of WordNet senses and abstractions in syntactic constructs, we require a corpus that is annotated with both syntactic and word sense tags. The Brown Corpus [Kučera and Francis, 1967], the first large scale computerized text collection project, luckily has been annotated both with syntactic tags (as a part of the Penn Treebank project) and (a third of it) with WordNet sense tags (the SEMCOR project [Landes *et al.*, 1998]). In Section 4.2 we discuss how these two corpora were merged and compiled into a format usable in our framework.

4.1 Grammar and Lexicon

Recall that our goal includes that our parser use a linguistically plausible grammar and lexicon. To that end, we have acquired a grammar from The Boeing Company. Linguists at Boeing have spent 20 years developing a parser, grammar, and lexicon for a feature-rich phrase structure description of Simplified English, a controlled language subset of English that is the intended language of Boeing's maintenance manuals. The parsing system is called SAPIR, and is described by Harrison and Maxwell [1986]. While SAPIR's coverage is quite broad, we have two immediate problems in that its lexicon is tailored toward airplane maintenance manuals, and its grammar is designed to recognize *Simplified* English.

SAPIR's grammar has slightly more than 500 phrase structure rules. Almost 10% of them are "error" rules, written to isolate both common errors (*e.g.*, subject-verb *dis*agreement) and English constructions that are not allowed in Simplified English (*e.g.*, `ing` verb phrases postmodifying nouns). In Boeing's grammar check mode, all "error" rules present in a final parse trigger a warning flag, to alert the writer that the phrase needs correction. For our purposes (using SAPIR purely as a parser, not as a grammar checker), it is sufficient to inspect the "error" rules, and remove only those that model actual syntactic errors. About 2/3 of the "error" rules are thereby removed from the grammar. In addition, we make a number of other minor changes. For example, we will need every rule in the grammar to

[3]Recall that R is our set of phrase structure rules, and L is our lexicon. We use n in the exponent, and not $n+1$, because we insist that every rule in the grammar be a *headed* rule, as described in Section 2. Thus in Equation (1), $w = h_i$ for some i.

have a single[4] right-hand-side daughter identified as the "head constituent" of the structure. We added that specification on rules where it was missing. Also, a handful of rules were handling constructions that were specific to Boeing's manuals, and those spotted were removed. As we endeavored to create a corpus (as described next), we occasionally found particular constructions that were not handled, and we added rules to handle them. After all these changes, and adding the rules mandated by lexical entries as described next, the grammar stands at 549 rules.

Since SAPIR's lexicon is not (intended to be) complete, we looked for another syntactically informed lexicon to incorporate into the parser. COMLEX [Grishman *et al.*, 1994] is a comprehensive English lexicon developed at New York University, available through the Linguistic Data Consortium. COMLEX provides morphological and syntactic information for approximately $39,000$ lemmas. By mapping COMLEX's syntactic features to SAPIR's syntactic features, we are able to create a broad coverage lexicon usable by SAPIR. Note that COMLEX tends to specify richer subcategorization frames than does SAPIR, and thus we added phrase structure rules to the grammar to cover additional lexical verb phrase rules "specified" by COMLEX frames. The creation of a lexicon for SAPIR uses both a small core file of lexical forms taken directly from SAPIR's existing lexicon, and the mapping from COMLEX. The mapping is from version 3.1 of COMLEX, with some additions made locally.

4.2 Corpus Creation

We mentioned above that (a portion of) the Brown Corpus has been both syntactically tagged and word sense tagged. Before merging these two versions of Brown, we preprocessed the corpora to some extent.

We start with the parsed Treebank data, which has phrase level bracketings but no part of speech tags. An example bracketing from section `e21` of the Brown Corpus is in Figure 2. (Treebank also provides a bracketed *and* part of speech-tagged version that we did not use.) Several adjustments are made to the corpus:

- Quotation marks are deleted.
- We remove the final `Q` from the phrase types `SBARQ` and `SQ` and the initial `WH` from the phrase types `WHNP`, `WHPP`, `WHADVP`. (This component of the phrase type will be realized as a feature by SAPIR, so we need only preserve the major category.)
- Any indecision or ambiguity of tag assignment indicated by alternation (`|`), *e.g.*, `PP|ADVP`, is reduced to the first tag.
- Phrases consisting entirely of the token `*pseudo-attach*`, indicating the "true attachment" for constituents that have been extraposed, are deleted. (Figure 2 has an instance of this.) Several types of extraposition are handled by SAPIR through feature passing.

[4] Actually SAPIR (and our model) allows any non-empty tuple of right-hand-side constituents to be the head. In Section 5.3 we discuss how we handle multiple heads.

```
( (S
   (NP Notice
       (PP-1 *pseudo-attach*))
   (AUX should)
   (VP be
       (VP taken
           (PP-1 of
                 (NP (NP (NP  unusual rock formations)
                         ,
                         (NP deposits)
                         ,
                         or
                         (NP shapes
                             (PP of
                                 (NP
                                  (NP the earth)
                                  's crust))))
                     (PP in
                         (NP your region)))))))
  .)
```

Figure 2: Example Treebank bracketing, for the string 'Notice should be taken of unusual rock formations, deposits, or shapes of the earth's crust in your region.'

- Indices attached to phrases, such as PP-1, (usually for purposes of attaching extraposed constituents) are deleted.
- This preliminary version of Treebank's tagging of Brown appears to have duplicated (or triplicated) exclamation points, question marks, and semicolons in the trees that do not appear duplicated in the raw text form of Brown. All repeated instances of any of these punctuation marks are deleted.[5]

There are 20,427 trees in the Treebank-annotated files of Brown that are also sense-tagged by SEMCOR. We remove from the corpus all trees that do not have two top level constituents, where the second is a "final" punctuation mark. This has the effect (hopefully) of screening out headlines (which do not adhere to the same grammatical rules as running text) and sentence fragments. We are left with 19,843 trees.

Besides the minor phrase naming changes noted above, there is a major difference between SAPIR and the syntactic theory underlying the Treebank annotation regarding empty phrases, particularly null subjects. Phrases that are taken by SAPIR simply to be infinitive verb phrases are marked in the Treebank annotation as sentences with empty subjects. We

[5] Of course, these punctuation marks may actually be repeated in the original text, but removing all repeated instances is easier and leads to an overall improvement in the corpus (more Treebank strings end up matching with SEMCOR strings).

```
(PP in (S (NP *)                    (PP in (VP going
       (VP going (NP home))))                  (NP home)))
```

(a) (b)

Figure 3: An original and "repaired" tree after gap-fixing

process these trees to achieve the appropriate syntactic structure. Thus the Treebank tree in Figure 3(a) is mapped to the simpler structure in Figure 3(b).

The mapping is accomplished via a set of pattern-matching routines that operate like regular expressions of s-expression units. Thus one rule takes trees that match

```
 (S $PRE-MODS* (NP *) $ADV1* (AUX TO) $VP*)
```

and maps them to

```
(VP $PRE-MODS*        $ADV1*      TO  $VP*)
```

removing the `S` level bracketing, the empty `NP` subject, and the `AUX` label. This stage of "gap-fixing" also does a few other syntactic manipulations:

- `(PP (PP ..)  (PP ..))` structures have their outer `PP` removed.
- Empty `NP`s that are passivized out of `PP`s are deleted and the `PP` bracketing is removed, leaving the bare preposition.
- Several specific "complex" phrases such as `(PP at (ADVP once))` are flattened into the structure we already know the lexicon requires, *e.g.*, `(ADVP at once)`.

These "gap-fixed" $19,843$ trees are the version of Treebank we will try to merge with SEMCOR.

The SEMCOR datafiles are also preprocessed, to a lesser extent. The most significant change concerns verb complement constructions that are tagged with a multi-word verb sense from WordNet. For example, occurrences of 'worked out' in phrases like 'worked out a plan' or 'worked it out' are tagged in SEMCOR as occurrences of lemma 'work_out' with the sense lexsn=2:30:00:: (a pointer to a synset). Similarly 'take a joke' and its variations, as in 'took the joke poorly', are tagged as occurrences of lemma 'take_a_joke', again with a pointer to a synset. We contract the scope of such tags (without changing their content) so that the tag applies just to the initial verb, not the entire multi-word construction. We then add tags to the remaining words indicating that they do not make independent semantic contributions. Thus, for instance, the following SGML tag for the words 'worked out' in section j14 of SEMCOR is transformed as indicated:

```
<wf cmd=done lemma=work_out wnsn=1 lexsn=2:30:00::>worked out</wf>
```

becomes

```
<wf cmd=done lemma=work_out wnsn=1 lexsn=2:30:00::>worked</wf>
<wf cmd=ignore>out</wf>
```

(SGML is really a family of languages, the most famous of which is HTML; hence this example looks similar to HTML.) In making this kind of transformation, we are adjudicating between the somewhat different notions of “lemmatization” found in WordNet and SAPIR, deciding in favor of SAPIR. SAPIR, as a parser and grammar, tends to view multiple words as the product of multiple syntactic rules. WordNet, in contrast, often views a collection of words as jointly referring to a particular concept in the ontology, and hence tends to treat such collections as single lemmas. But in parsing, we require that word senses be assigned to individual *leaves* in the parse tree, not to multiple leaves, possibly separated by intervening material.

This mapping is aware of “true” multi-word verbs that exist in the lexicon; so instances of, for example, ‘water ski’ are not expanded in this way. Note that in other work [Seagull, 2000a] we discuss how we have more intelligently decomposed some of these “spurious” multi-word verbs.

There are also a small number of other very minor changes, all meant to facilitate the merging of Treebank and SEMCOR's annotation of the Brown Corpus. The changes include i) replacing the tag <punc>'</punc> with the correct sense and form of the word ‘degrees’, since this is how the token is realized in the Treebank; ii) eliminating three consecutive instances of <punc>&</punc> (it is unclear what these mean; they do not correspond to the triplicated Treebank sequences of (END_OF_TEXT_UNIT)); iii) removing double quotes; and iv) replacing the sequence

```
<punc>*</punc>
<punc>*</punc>
<wf cmd=done pos=NN ot=notag>h</wf>
```

with </s> (end of sentence) and <s snum=?> (beginning of sentence), because that sequence mostly (though not always) corresponds to a sentence break in the raw text of the Brown Corpus.

Merging Treebank and SEMCOR

The minor changes to SEMCOR just mentioned (and others that were not, both to SEMCOR and Treebank) are to facilitate the merging of the two corpora. The merging is done wholly on the basis of string matching, which is why the apparently minor changes mentioned above are quite vital. Once we have the preprocessed Treebank and SEMCOR data as described above, we can extract sentences and merge the annotated data. The merging algorithm is greedy and simple, but it is also fast. For each file of data (the Brown Corpus is actually 500 separate text samples, and SEMCOR includes annotation of 186 of them), we iterate over the Treebank trees (recall headlines, *etc.* have been removed) and search for the first sentence in the SEMCOR file that is identical to the sentence of that tree.[6] If the sentence of the Treebank tree is not found, it is thrown away. Otherwise, the position of the corresponding SEMCOR sentence is used as the starting point for the next search, and we move on to the next tree in the Treebank file. The result is that only token-identical sentences are merged, and the order of the merged sentences must be the same as their order in each of the annotated corpora.

There are several significant oversights/simplifications that limit the extent of the merging.

- In general, there are occasional misspellings in the transcriptions, and mispunctuations will similarly occur. But two very regular problems are that there are no forward slashes in the Treebank files (Fractions, for example, are not correctly represented. This is undoubtedly a technical problem due to the preliminary nature of Treebank's annotation of the Brown Corpus.), and that hyphens hyphenating words are frequently missing in SEMCOR's annotation. Also, hyphens as phrase level punctuation, dashes, are recorded as two hyphens in Treebank but just one in SEMCOR.

- Our merging depends on matching sentences,[7] so knowing the boundaries of sentences is obviously important. One major source of misalignment is enumerated sentences, as in 'John decided that we should 1) arrive early and 2) leave early'. It seems this is often realized as one tree in Treebank but separate sentences in SEMCOR; exactly how they are annotated varies, but in general, the "sentences" do not align. When quoted speech includes two syntactic sentences, as in 'John cried out "He's here? Oh, no!"', how Treebank and SEMCOR divide this into sentences varies. Although SEMCOR might provide a nested structure through sequences like `<s snum=71>` ... `<s snum=72>` ... `</s>` ... `</s>`, this does not appear to be used in the annotation. (Treebank, as one might guess, uses nesting quite a bit.)

- Our procedure for removing null subjects, described above as part of the preprocessing of the Treebank data, is not able to eliminate *all* of the null subjects, so the remaining 24 trees that are "irresolvable" are not merged.

Some of these problems have apparent solutions: incorporating both an edit distance measure (for matching) and a realignment procedure could handle many of the mismatches. As well, a dedicated procedure that knows about missing forward slashes in Treebank and missing hyphens in SEMCOR would be workable. However, since our current approach is simple, fast, hopefully low-error (*i.e.*, high precision), and is able to merge $16,640$ of the $19,843$ Treebank trees, we have not worked further on it.

The result of the merge is the original Treebank tree plus a list of pairs of WordNet sense tags and their token positions in the sentence (the allows for sense tags that span multiple

[6] Actually, although we heuristically insert whitespace around words and punctuation in the extracted sentences, the comparison simply looks at the *tokens* and checks that the sequences of tokens are identical (ignoring case). In fact, the representations in Treebank and SEMCOR do not preserve the original whitespace information, so we cannot reconstruct reliably the original text string.

[7] By "sentence", we do not mean to imply that these strings are *syntactically* sentences. Rather, we use the term to refer to the units into which the discourses are segmented.

tokens like that for 'hot dog'). This also preserves annotator uncertainty or ambiguity; sometimes two or more senses are annotated for a single instance of a word. When data is collected from the corpus, these multiple senses split equally the frequency of the word tagged.

Extracting our Corpus

Recall that our goal is to use corpus probabilities to *guide* a handwritten grammar, so we require frequencies of grammar rules in SAPIR's grammar, rather than frequencies of syntactic patterns explicit in the corpus (as are used by the data-driven parsing approaches described in Sections 1 and 2). Fortunately, SAPIR has an interface whereby we can provide the Treebank bracketing, and SAPIR will find a parse (if it exists) respecting the bracketing. Hence we take Treebank's bracketing to be an *underspecification*.

For example, the bracketing in Figure 3(b) is read by SAPIR and the parser is able to construct a parse with a gerundive `NP` containing the `VP[ing]`. The tree structure corresponding to the extracted parse is:

```
(PP (P IN)
    (NP
      (VP (V GOING)
          (NP (Nbar (N HOME))))))
```

Note that the extracted parse is not just this tree, but also the particular rule identifiers that license each expansion. Note also that this process treats the trees as more ambiguous than its creators intended. Treebank might supply a bracketing as `(VP (V ..)  (NP ..) (PP ..))`, but SAPIR is free to construct the larger `NP`, as in `(VP (V ..)  (NP (NP ..) (PP ..)))`. This is necessary to allow the sort of "fleshing out" of trees just described, but we have tried to programmatically discourage the parser from choosing such interpretations, as described later in this section.

Of course, not every Treebank bracketing is extractable with our grammar and lexicon. The current parser finds parses for 6373, or 38.3%, of the 16,640 merged trees. There are three major reasons for not finding a parse.

- The grammar and lexicon do not cover the sentence in question. This may be due either to incomplete grammar coverage, questionable grammaticality, incomplete lexical specifications, or some combination thereof. One of the genres covered in the Brown Corpus is fiction, and some pieces in particular have a good amount of casual dialogue, which does not adhere to standard conventions for written English. As our grammar is essentially a grammar for written English, we cannot parse these sentences.

- Treebank's syntactic analysis cannot be considered an underspecified version of SAPIR's analysis for a particular structure. We have tried to coerce Treebank's syntax to something amenable to SAPIR's grammar, as described above in the section on preprocessing Treebank. We have also modified the constituency of certain relative clause constructions in SAPIR's grammar to better match Treebank's annotation. But certainly there are particular unusual constructions whose syntactic analysis is less certain, and these are likely receiving different treatments by SAPIR and by the Treebank annotators. In fact, the set of major category labels for Treebank and SAPIR is slightly different, so certain Treebank bracketings cannot be parsed simply for a mismatch in category name.[8]

- The bracketing is actually incorrect. The construction of Treebank begins with an automatic parser, and its output is corrected by human annotators. This preliminary version (0.75) of Treebank's annotation of the Brown Corpus has not been checked over as many times as other portions of the Penn Treebank, *e.g.*, the Wall Street Journal corpus. Syntactically impossible annotations will be screened out by our extraction process. For example, the structure for the phrase 'picking herself up' in a sentence in file `k12` is `(VP picking (NP herself (PP up)))`. The mislabel of 'up' as a `PP` instead of `PART` is certainly a mistake. But even more alarming is the mistaken constituency, requiring 'herself up' to be an `NP`. As SAPIR does not allow `PP`s to modify pronouns, this tree will not come through to our corpus.[9]

In an inspection of unparsable sentences at an earlier point in the corpus creation, the two biggest culprits preventing Treebank trees from parsing were ungrammaticality (this includes misspellings and "casual speech", *e.g.*, `holdin'`) and incomplete lexical specifications. Many lexical entries were added to or augmented in COMLEX, but certainly many omissions remain. Another significant problem is that although SAPIR allows adverbial phrases to be freely inserted into structures (*e.g.*, 'John said *slowly* that he would return'), we disable that feature, as our stochastic model would need to be modified to account for this "free insertion".

Given we want to use the Treebank's bracketing annotation to construct a syntactically rich corpus, it is important to know that we are reasonably certain of the parses that SAPIR extracts. We find that more than half of the Treebank trees that we can parse yield at most four parses. If the bracketings in those trees are ignored, the median number of parses per string is 34, so the bracketings are severely constraining the space of possible syntactic interpretations (as one would hope). 37% of the parsable bracketings yield just one or two parses.

The treebank bracketings are indeed severely constraining the set of possible parses for a sentence in the corpus. If we disregard the bracketings, SAPIR is able to find parses for for 78.3% of the strings (what is often called "grammar coverage"), with a median ambiguity of 150 parses per string. This is to be contrasted with the figures mentioned above for parsing while respecting bracketings: 38.3% parse with a median ambiguity of 4 parses per string. Note that over 10% of the strings that did not parse failed because of memory limitations; the parser was set to give up after creating 100,000 constituents.

---

[8] There are instances where one Treebank label maps to two or more SAPIR labels, so we are not able to fix the problem with a simple mapping.

[9] The FIDDITCH parser used as the first stage in annotating the Penn Treebank is used because of its high precision, at the cost of lower recall [Marcus *et al.*, 1993]. We would conclude this error must have been accidentally introduced by an annotator.

label	description	pct parsed	mean sentence length	no. merged trees
a	Press: Reportage	41.6	20.5	543
b	Press: Editorial	40.6	17.9	180
c	Press: Reviews (theatre, books, music, dance)	33.2	20.6	247
d	Religion	24.5	27.4	249
e	Skills and Hobbies	34.6	20.2	1073
f	Popular Lore	34.7	21.5	1453
g	Belles Lettres, Biography, Memoirs, etc.	30.3	22.7	1344
h	Miscellaneous	32.1	23.7	704
j	Learned	33.0	23.9	2655
k	General Fiction	45.1	14.9	3653
l	Mystery and Detective Fiction	42.0	13.4	1511
m	Science Fiction	39.2	12.9	245
n	Adventure and Western Fiction	43.7	12.5	1462
p	Romance and Love Story	43.0	14.7	747
r	Humor	36.8	18.0	574
	all	38.3	18.5	16,640

Table 1: Percent, by genre, of Treebank bracketings that are parsable with our grammar. Note that percent parsable and average sentence length of the merged trees correlate with a sample correlation coefficient of -0.880. Perhaps not surprisingly, the greatest aberration from the correlation is the exceedingly parsable section `a`, "Press Reportage".

As might be expected, the sentences (and bracketings) we can handle tend to be shorter than the average sentences in the corpus. The average number of words (defined by space-delimited sequences of characters) in the 16,640 merged sentences is 17.4 (ranging from 1 to 154); the average for our 6373 extracted sentences is 12.4 (ranging from 1 to 65).

When a bracketing licenses multiple parses for a string, we choose for our corpus the one parse that is preferred by SAPIR. SAPIR has an extensive system of costs on its grammar rules; this is how researchers at The Boeing Company perform disambiguation and pruning. The costs are manually assigned weights, and are generally insensitive to the lexical context. In general, we impose a preference for arguments over adjuncts. (COMLEX will dictate what phrases may be arguments to a particular verb.)

Certainly, some portions of the Brown Corpus are easier to parse than others. Newspaper copy tends to be the most regular (then again, the book review section tends to have many esoteric words and creative constructions), but fiction may have dialogue that adheres to a different grammar, as mentioned above. In Table 1 is listed, by genre, the percent of merged trees that are parsable. Note however that sentence length appears to be a significant factor in predicting parsability.

When parsing a bracketed, sense-annotated sentence, we also ensure that the annotated senses are senses that our parser would propose for that lexical item. Without these checks, we are setting ourselves up for a word sense disambiguation problem where the best possible algorithm cannot achieve 100% accuracy.

There are three consistency checks that an annotated sense must pass, with respect to the parse we have extracted.

- *Consistent root orthographies.* The orthography of the root word form (according to the parser) must match exactly the orthography of the WordNet lemma. Aside from the exception described in the following paragraph, this rarely happens without a mismatch in part of speech (the next bullet). Yet there are some instances. For example, WordNet has an entry for the plural noun 'polls', while COMLEX has it only as the plural form of the lemma 'poll'. Thus any tag that uses a sense of 'polls' will not be retained.

  The lone exception is that we allow root words of verbs from the parser to match against the initial word piece of a multi-word sense tag. Recall that in preprocessing SEMCOR, we shift the tags so that multi-word verbs are taken to be tags on just the initial verb piece. We allow inexact matching here for the same reason. Actually, there are some instances where this helps matching. File `g11` has tagged 'talking of' with sense 1 of 'talk about'; by considering multi-word verb sense tags to be attached only to the initial verb piece, tags like this cause no trouble at all.

- *Consistent parts of speech.* The part of speech of the annotated sense tag must be consistent with the part of speech of the word in the extracted parse. We have used the parsed but not part of speech-tagged version of the treebank, and thus a structure like `(NP the interim jury)` may be parsed either as a `DET-ADJ-N` sequence or a `DET-N-N` sequence. If SAPIR chooses the latter parse (according to the handwritten costs on rules mentioned above) and SEMCOR has tagged 'interim' with an adjectival sense, we will not keep that sense tag. In fact, COMLEX does not specify an adjectival sense of 'interim', only a noun sense. If we were to use the fuller treebank structure, `(NP (DT an) (JJ interim) (NN jury))`, we would not be able to extract a parse for this sentence. In this case, it may not matter; it is not clear that the interpretation we would get from the noun-noun compound 'interim jury' is distinct from that which we would get from an adjective-noun construct, given Merriam-Webster's definition of the adjective 'interim': "done, made, appointed, or occurring for an interim".

  Sometimes a mistaken part of speech will lead to a mismatch in orthography. SAPIR allows verb participles to form adjective and adverbs in appropriate constructs. Thus 'veiled threats' is parsed as an adjectival passive followed by a noun. But these participles often have their own entries in WordNet. According to SAPIR, the root form of this adjectival use of 'veiled' is the verb, 'veil'. But the SEMCOR annotation uses the WordNet tag for sense 1 of the adjective 'veiled'. Thus the part of speech mismatch leads to an orthography mismatch, and the tag fails on both counts.

- *Alignment of SAPIR and SEMCOR lemmas.* The token span of the annotated tag must correspond in the extracted parse to a *leaf node* whose orthography and part of speech match as described above. Mismatches here arise from different *lemmatization* theories, it seems. WordNet tends to have a great many multi-word items, possibly

part of speech	multi-word		single-word	
	WordNet	COMLEX	WordNet	COMLEX
noun	$47,174$	108	$47,300$	$21,827$
verb	2614	7	7705	5658
adjective	645	33	$19,525$	8161
adverb	857	63	3689	3057

Table 2: Number of lexical items in WordNet and COMLEX, broken out into multi-word and single-word. "Multi-word" means any orthography with a space character in it, so the "single-word" columns include hyphenated entries.

because many of them are synonymous with single-word items. COMLEX, however, is a syntactic lexicon, and its purpose (presumably) is to feed into a productive system of rules to arrive at a syntactic description of English. As one measure, Table 2 lists the sheer number of multi-word items in WordNet as opposed to COMLEX. Granted, WordNet is a larger lexicon than COMLEX overall, so the table also shows the larger number of single-word counts in WordNet. Note, however, that the additional number of multi-word entries in WordNet is disproportionately larger than the additional number of single-word entries. As far as our merging goes, this means that there will be many instances of what the parser sees as complex phrases that are tagged in SEMCOR as if they were undecomposable lexical entries, such as 'primary election'. To completely ignore these sense tags would lose a lot of semantic information (which we presume is helpful to parsing). Therefore, we add to our lexicon most of the ~$47,000$ multi-word nouns in WordNet. Nouns are relatively easy to add, as subtle syntactic distinctions are not as pervasive within nouns as they are within adjectives and verbs. We assume all can be either mass or count (In fact, we make this assumption for most nouns in COMLEX anyway, as there does not seem to be a feature that corresponds directly to the mass/count distinction, and it is a distinction easily crossed anyway, *e.g.*, [Pelletier and Schubert, 1984].) and we construct plural forms heuristically. We do not add any multi-word nouns that are truly "phrasal", like 'woman of the house', since the Penn Treebank labels these items with internal structure, as in `(NP woman (PP of (NP the house)))`, and thus we would throw away the sense tag anyway (because the extracted parse will not have a leaf node over the word string 'woman of the house'). Therefore instead of adding all $47,174$ multi-word nouns, we add the $44,915$ that are not phrasal. As mentioned above, multi-word verbs are treated for the most part as senses of the initial word piece. We have, however, extracted 66 "true" multi-word verbs from WordNet such as 'belly dance' and 'water ski', and explicitly added them (and their syntactic features) to the lexicon.

Above we saw how part of speech and orthography can both prevent a sense from being retained. As well, token span and orthography can together do the same. As an example, from file `f20`, the tag for sense 1 of 'choirboy' appears for the input 'choir boy'. This input is only parsable as a noun-noun compound, since 'choir boy' is not a multi-word noun in either COMLEX or WordNet. Hence there is no leaf node to assign the tag 'choirboy' to, and the orthographies are not string-equal anyway. So this tag is removed.

The bulk of the senses match correctly; we screen out a total of 3220 senses and retain 35,616. Of the deleted senses, 215 are for the adverb ‘ n’t ’, which is not a separate syntactic item in SAPIR. Contractions are treated more like inflections of the verb than verb plus adverb constructions. (Moreover, ‘ n’t ’ is monosemous in WordNet, so eliminating it only downplays word sense disambiguation accuracy.) Also, 162 are senses for numbers (written with numerals), and these items are not in SAPIR's lexicon. Rather, a preprocessor replaces them with tokens like `NUMBER1`.

In sum, we start with the ~375,000 word and ~20,000 sentence section of the Brown Corpus that was sense-tagged in SEMCOR, merge together Treebank's syntactic annotation and SEMCOR's word sense annotation, and extract an 85,532 word and 6373 sentence corpus of sentences we can correctly analyze using SAPIR, constrained by the Treebank bracketing and SEMCOR sense tags. From this corpus we extract head patterns of word senses in syntactic constructs, where the phrase structure rules licensing these constructs come from an extant grammar. The probabilities our language model requires are estimated directly from these patterns and their abstractions, and this is the topic of the next section.

5 Estimating Parameters

We have formulated our model in terms of word senses and incorporated WordNet into our framework so that we can use n-ary lexical statistics without being overcome by data sparsity problems. We have explored two basic methods of "backing off" in the WordNet ontology when estimating the syntactic expansion and lexical introduction probabilities.

5.1 Backing Off

In an n-gram language model, estimates are needed for probabilities of words given the $n-1$ preceding words, $P(w_i \mid w_{i-1} \dots w_{i-(n-1)})$. When the data used to estimate these probabilities are unavailable (or unreliable), *lower-order* models are used. For an n-gram estimate, the next-lower-order model uses the $(n-1)$-gram estimate, $P(w_i \mid w_{i-1} \dots w_{i-(n-2)})$. Speech recognition researchers advocate interpolating higher-order models with lower-order models for best results [Jelinek and Mercer, 1980].

Whenever we can, we appeal to WordNet to supply the "lower-order" models. Instead of reducing the number of lexical items in our statistics (as n-gram models do), we reduce the *specificity* of those lexical items, by following hypernym links in the WordNet ontology. We have explored two decision criteria for backing off in the conditioning information; one employs thresholding, and the other uses statistical significance testing.

Thresholding

Note that both of the probability distributions we need to estimate (terms (2) and (3)) are conditional distributions with a lexical item as part of the conditioning information.

Our first cut at estimating these probabilities is the maximum likelihood estimate: simply calculate the appropriate relative frequency. This approach has the failing that many of the relative frequency ratios have very small denominators (often leading to inaccurate probability estimates) or even 0-count denominators (leading to impossible probability values).

To avoid these problems arising from insufficient numbers of observed trials, many researchers, *e.g.*, [Collins, 1997], do not use probabilities that come from relative frequencies whose denominator is below a fixed threshold. Our first approach to dealing with the problem is similar: when the denominator of a relative frequency estimate is below a certain threshold, we appeal to *abstractions* of the lexical item, and progressively search upward in the ontology until we find a denominator with sufficient data in the corpus. For example, suppose we want to estimate the probability in Equation (4).

$$P(\langle \text{buy}, \text{dog}, \text{for} \rangle \mid \texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V NP PP}) \tag{4}$$

To directly estimate this probability, we calculate the relative frequency

$$\frac{c(\texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V}[\text{buy}]\ \texttt{NP}[\text{dog}]\ \texttt{PP}[\text{for}])}{c(\texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V NP PP})}.$$

But if the denominator (how many `V-NP-PP` constructions we saw headed by 'buy') is below a threshold, we use WordNet to abstract to the *hypernym* of 'buy', 'acquire'. If there are not sufficiently many occurrences of 'acquire' heading a `VP8` construction in our corpus, we continue to abstract until we find a hypernym that allows for sufficient statistics. Eventually, the top of the ontology is reached, and we are in effect conditioning on *any* `VP8` construct, regardless of its head word.

In our examples we will often simply refer to words, when in reality the probabilities are in terms of word *senses* and *synsets*. The initial probability estimated is in terms of word senses (sense 1 of 'buy', in the above example), and the initial abstraction is to the synset containing that sense of 'buy'. That synset has just two senses, but dominates a total of 14 senses, all of which contribute to the statistics for that synset. Further abstractions are all in terms of synsets. (The subsequent hypernym is a synset containing sense 1 of 'get' and sense 1 of 'acquire'; it dominates several hundred senses.)

This approach is similarly followed for syntactic expansion probabilities. One exception is that we do not appeal to abstractions when the expansion in question is a subcategorized rule. For example, if we are estimating the probability $P(\texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V NP PP} \mid \texttt{VP}[\text{buy}])$, and there are insufficient occurrences of `VP`s headed by 'buy', we would immediately appeal to the probability of *any* `VP` expanding via `VP8` to a `V-NP-PP` construction.

Sigdiff Testing

Using a fixed threshold to determine the validity of a particular relative frequency estimate has produced good results with an earlier version of our corpus [Seagull and Schubert, 1999]. We have found that results improve with a back-off scheme that is more sensitive than a crude cutoff. The idea is to use a hypothesis-testing mechanism to decide when to back off. Essentially, given a particular conditional probability, we take the null hypothesis (H_0) to be that the lexical item in the conditioning information provides *no* additional information to help us calculate the correct probability. That is, given a conditional probability of the form $P(\vec{Y} \mid \vec{X}, w)$, where w is a lexical item, H_0 is that $P(\vec{Y} \mid \vec{X}, w) = P(\vec{Y} \mid \vec{X})$ (the *a priori* estimate, *i.e.*, *before* the lexical item is added to the conditioning information). The alternative hypothesis is that these two probabilities are different, and we would then estimate $P(\vec{Y} \mid \vec{X}, w)$ by its maximum likelihood estimate. If we cannot reject the null hypothesis, we use WordNet to abstract to the hypernym of w, and repeat the procedure. Thus our procedure finds the most specific hypernym of w that (together with X) makes a significantly distinct prediction about Y (than X does alone).

The intuition is that if we cannot reject the null hypothesis, it may well be because there is so little data that we have no confidence in our estimate of the probability (when factoring in the lexical item). As we move "up" in the ontology, we will have more and more data (resulting in a more reliable probability estimate) but that data will look more and more like the data supporting the *a priori* estimate. If there is sufficient evidence that this lexical item (and items that are similar semantically) lends some predictive force to predicting $\vec{Y}$, given we already know $\vec{X}$, we will at some point be able to reject the null hypothesis. Otherwise we will just use the *a priori* estimate.

The hypothesis testing is implemented by constructing probability intervals for the events in question, and comparing intervals to see if they are disjoint. The calculation of intervals uses an iterative procedure, but we use the normal distribution as an approximation when the numbers in question are large enough, as described by Larsen and Marx [1986, p. 295].

5.2 Smoothing and More Backoff

Whether we use a strict cutoff or a hypothesis testing method, these techniques are intended to relieve data sparsity problems manifested in the denominator (the *conditioning* information). But the numerator is is even more susceptible to sparsity problems, since it requires particular heads on the right-hand side: $0/N$ yields a probability of 0, which will lead to parses having probabilities of 0, since we estimate the probability for our tree as a sequence of independent events whose probabilities are multiplied together. Any work in statistical NLP must counter this problem. Typically, *smoothing* is employed. The idea is to take an observed data distribution and *smooth* out the probability mass, so that events that were seen zero times actually have some nonzero probability, and the events that *were* seen have slightly less probability than before smoothing, so we still have a distribution that sums to 1. Chen and Goodman [1998] provide a detailed analysis of various smoothing algorithms for bigram estimates (probabilities of two words occurring in some correspondence, such as following one another or in a particular syntactic relationship).

We want to avoid zero-counts by appealing to WordNet for our "lower order" statistics. However, zero-counts are occasionally unavoidable. We may be trying to estimate the probability of some event that never occurred, in any context, and that we are unable to back off from (*i.e.*, a phrase structure rule that was never used in the parsed corpus). We also may be trying to estimate the probability of a particular word (sense) that never occurred in the training data. Even though we can back off to some extent, we will still need to calculate the probability of introducing that particular word into the phrase structure tree, as described below.

Therefore whenever we calculate a probability, we interpolate the maximum likelihood estimate with a simple uniform distribution.[10] The mixture is with weights $\lambda, 1-\lambda$, and for the experiments discussed in the Section 6 we use $\lambda = 0.9$. Certainly an optimization procedure could be used to find a λ that improves results, but we have not done that.

Backing off, Again

As mentioned above, we can sometimes back off from zero-counts when we encounter them; we can use WordNet to back off again, but this time in the *conditioned* information, as opposed to the *conditioning* information.

More specifically, when estimating a probability like $P(\langle\text{tax}, \text{outcast}\rangle \mid \texttt{VP}[\text{tax}] \xrightarrow{\texttt{VP1}} \texttt{V NP})$, *i.e.*, the probability that 'outcast' appears as the object of the transitive verb 'tax' (ignoring word senses right now), we would like to appeal to the ML estimate (although smoothed as just described). The smoothed ML estimate requires knowing just how many times the words in question appeared in the syntactic construct in question: $c(\texttt{VP}[\text{tax}] \xrightarrow{\texttt{VP1}} \texttt{V}[\text{tax}]\ \texttt{NP}[\text{outcast}])$. If this joint event has never occurred, we could still use our smoothed estimate to get a nonzero probability. But we can use WordNet to do even better. We search upward in the WordNet ontology, looking for some hypernym (or great-hypernym or great-great-hypernym) that *did* occur in this context (transitive verb phrases headed by 'tax'). For example, 'person' is the "great-hypernym" of 'outcast' and perhaps it (or one of the 7000^+ senses dominated by it) appeared as the direct object of 'tax' somewhere in our corpus.[11] Once we have found a parent that appeared in the context, we employ the search for adequate conditioning information as described above. Assuming the sigdiff method, we ask whether knowing the head word of the phrase predicts the non-head children significantly differently than the phrase (with no lexical information) does. In this example, we ask whether the knowledge that the head verb of the transitive verb phrase is 'tax' makes a significantly different prediction about the probability of the object being 'person' than if we didn't know it was a 'tax' verb phrase. We might not have sufficient data on 'tax' (as a transitive verb) but perhaps its great-hypernym, sense 3 of 'charge', does have enough data to *reliably* favor having 'people' as objects more than an arbitrary transitive verb does.

5.3 Multi-Headed Phrases

Our discussion thus far has presumed that each phrase has only a single head, and thus we have only one lexical item in the conditioning information for our two main probabilities. In fact, some phrases have *multiple* heads, most notably coordination constructions. To

[10] For example, we estimate $P(\texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V}[\text{buy}]\ \texttt{NP}[\text{dog}]\ \texttt{PP}[\text{for}])$, as the product of the probabilities $P(\texttt{VP}[\text{buy}] \mid \texttt{VP8})$ and $P(\langle\text{buy}, \text{dog}, \text{for}\rangle \mid \texttt{VP}[\text{buy}] \xrightarrow{\texttt{VP8}} \texttt{V NP PP})$. The uniform for the former is 1/[# rules that expand VPs]. The uniform for the latter is crudely taken to be $1/|L|^2$, where L is the lexicon.

[11] In our calculations of the lexical introduction probability, we always disregard the head child, since the conditioning information includes the head of the parent, and those two are equal with probability 1.

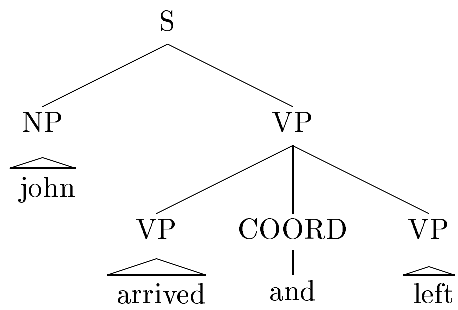

Figure 4: Parse tree for 'John arrived and left'

accommodate this, we allow heads to be nonempty tuples of lexical items. In the sentence 'John arrived and left', we would get a structure something like the tree in Figure 4. The head of the lower `VP`s would be $\langle \text{arrive} \rangle$ and $\langle \text{leave} \rangle$, respectively. The larger `VP` would have head $\langle \text{arrive}, \text{leave} \rangle$.

Essentially, we make the assumption that all coordination is distributive, and "distribute" the coordinates accordingly. The patterns for this sentence, listed as one per phrasal expansion, are listed below. The coordinated `VP` has been "distributed" over its two heads. Each expansion is written by specifying its head along with all nonheads. Since the coordinated `VP` has two heads, and we write an expansion for each, there is in each a "ghost" constituent that is neither the head of record nor a nonhead. The place of that ghost is marked by a $\cdot$.

$$
\begin{aligned}
\texttt{S}[\text{arrive}] &\xrightarrow{\texttt{S1}} \texttt{NP}[\text{john}]\ \texttt{VP}[\text{arrive}] \\
\texttt{S}[\text{leave}] &\xrightarrow{\texttt{S1}} \texttt{NP}[\text{john}]\ \texttt{VP}[\text{leave}] \\
\texttt{NP}[\text{john}] &\xrightarrow{\texttt{NP-PROPER}} \texttt{N}[\text{john}] \\
\texttt{VP}[\text{arrive}] &\xrightarrow{\texttt{COORD1}} \texttt{V}[\text{arrive}]\ \texttt{COORD}[\text{and}]\ \cdot \\
\texttt{VP}[\text{leave}] &\xrightarrow{\texttt{COORD1}} \cdot\ \texttt{COORD}[\text{and}]\ \texttt{V}[\text{leave}] \\
\texttt{VP}[\text{arrive}] &\xrightarrow{\texttt{VP0}} \texttt{V}[\text{arrive}] \\
\texttt{VP}[\text{leave}] &\xrightarrow{\texttt{VP0}} \texttt{V}[\text{leave}]
\end{aligned}
$$

When collecting data, each expansion normally gets a frequency of 1. When we distribute out the coordination, the frequency of 1 is prorated evenly among the "conjunct" expansions. Each of the `S1` and `COORD1` patterns above is given a frequency of only 1/2, so the total count of expansions above is 5, not 6 (matching the actual number of expansions in the parse). When parsing a multi-headed structure, the heads are distributed out in this way, and the individual probabilities are *summed*. With this prorating and summing approach, we end up with an estimate that matches the empirical data.[12]

[12] An alternative approach that we first employed (and was the basis for earlier results [Seagull and

This presumption that we can "multiply out" the heads seems appropriate for distributive coordination, but does not seem quite right for collective coordination. Consider the sentence 'John and his friends assembled in the foyer'. Recall that although we have not been indicating them, all content words are actually tagged with word *senses*: 'assemble' in the above sentence would be tagged with WordNet's sense 2. Sense 1 would be appropriate for usages like 'John assembled a model airplane' and sense 3 should be the tag for the causative 'John assembled his friends in the foyer'. Our distributive assumption would give the following two sentence-level patterns (writing sense n of word w as $w\#n$):

$$\begin{aligned}\texttt{S}[\text{assemble\#2}] &\xrightarrow{\texttt{S1}} \texttt{NP}[\text{john}]\ \texttt{VP}[\text{assemble\#2}]\\ \texttt{S}[\text{assemble\#2}] &\xrightarrow{\texttt{S1}} \texttt{NP}[\text{friend}]\ \texttt{VP}[\text{assemble\#2}]\end{aligned}$$

In a large enough corpus, then, we will collect subject-verb statistics on $\langle$john, assemble#1$\rangle$ *and* $\langle$john, assemble#2$\rangle$ *and* $\langle$john, assemble#3$\rangle$. We might even have seen $\langle$john, assemble#2$\rangle$ more than the others, making us prefer this (infelicitous) collective-subject-desiring sense 2 of 'assemble' in a sentence such as 'John assembled his new entertainment center'.

This distributive assumption is probably not overwhelming our experiments: coordinated nodes account for around 1% of all nodes in our corpus, and around 60% of them are not NP nodes, and so cannot be interpreted collectively. As described in [Seagull, 2000b], our model can be extended to accommodate both types of coordination. It is most advantageous to combine this enhancement with another enhancement discussed there, an extension to handling all syntactic features. (Without this, the model would not distinguish 'The friend assembled in the foyer' from 'The friends assembled in the foyer'.)

5.4 Searching through the Hypernyms

At several points, we have mentioned "searching upward through the WordNet ontology". Since WordNet allows multiple inheritance, this search is not as simple as it might otherwise be. As a first approximation, we simply traversed the hypernyms in breadth-first order; this is the simplification underlying earlier work [Seagull and Schubert, 1999].

Note that even if WordNet did not have multiple inheritance, we still need to deal with a search that is not as simple as moving "up" in a tree. When we abstract over n-tuples of lexical items, there are multiple n-tuples of word senses (WordNet synsets) that serve as parents but are incomparable to each other. Consider the sentence 'John gave the boy a ball', for which we need to estimate the lexical introduction probability for the verb phrase expansion, $P(\langle \text{boy}, \text{ball}\rangle \mid \texttt{VP}[\text{give}] \xrightarrow{\texttt{VP6}} \texttt{V NP NP})$. If we haven't seen any instances of the pattern $\texttt{VP}[\text{give}] \xrightarrow{\texttt{VP6}} \texttt{VP}[\text{give}]\ \texttt{NP}[\text{boy}]\ \texttt{NP}[\text{ball}]$, we will need to search for some pair of

Schubert, 1999]) was to give each expansion a frequency of 1, and then *average* the probabilities of the component conjuncts in parsing. This approach actually overestimates the true probability of a coordinated expansion. Imagine that in this framework we have seen x non-coordinated NP nodes and n coordinated NP nodes, and we want to estimate $P(\texttt{NP} \xrightarrow{\texttt{COORD1}} \texttt{NP COORD NP} \mid \texttt{NP})$. Since coordinated expansions are counted twice, the ML probability estimate is $\frac{2n}{x+2n}$, whereas the "true" (empirical) probability is $\frac{n}{x+n}$. The ratio of these estimates (and thus of our overestimation of the probability) approaches 2 when $x \gg n$.

hypernyms of 'boy' and 'ball' that *did* occur as the objects in a ditransitive verb phrase headed by 'give'. As always, all lexical items are actually WordNet word *senses*, where possible. Assume the probability we are trying to estimate is for the appropriate senses of the words: sense 1 of 'boy', "a youthful male person", and maybe sense 6 of 'ball' "a spherical object used as a plaything". Not having seen $\langle \text{boy}, \text{ball} \rangle$ in this context, the simplifying assumption used in earlier work would have us first look at $\langle \text{male person}, \text{ball} \rangle$ and $\langle \text{boy}, \text{toy} \rangle$, *i.e.*, all the abstractions with distance 1 from the original words. Then we would look at all the abstractions with distance 2: $\langle \text{male person}, \text{toy} \rangle$, $\langle \text{person}, \text{ball} \rangle$, and $\langle \text{boy}, \text{artifact} \rangle$. The first abstraction to have appeared in the appropriate context (in this case, a ditransitive verb phrase headed by 'give') would be used in the subsequent search for adequate conditioning information. The order in which the abstractions of a given distance are searched would be arbitrary.[13]

This arbitrariness is certainly undesirable, since it leaves us without assurance that we are making optimal use of the head pattern statistics. One solution might be to evaluate the *size* of these jumps in the WordNet ontology. Our method treats all parent-child links as having the same length, but certainly this is not true. Sense 1 of 'physical object' has among its immediate children 'natural object', 'artifact', and sense 2 of 'ice', "the frozen part of a body of water". If we could recognize that 'ice' to 'physical object' is a larger jump than 'boy' to 'male person', we might be able to impose a complete ordering on the distance of abstractions from the original words, and thus always search the more specific patterns before the more general ones. Determining the size of these jumps is not a terribly well-defined or easy problem, however. Recently, Caraballo [1999] has described a method for automatically constructing a hierarchy of nouns from coarsely parsed text. Methods such as these, which depend on quantitative measures, would likely lend themselves to attaching quantitative measures to the jumps in WordNet's ontology. Resnik [1995] describes a method for computing similarity between items in a taxonomy (he uses WordNet) that utilizes information theoretic metrics; this approach is certainly relevant. Both researchers, however, assume [largely] monosemous words; to modify the approach for our purposes would likely require large amount of sense-tagged text.

An alternative is to not commit to the relative sizes of links in WordNet, but to attempt to adjudicate among the incomparable abstractions. Kyburg, *e.g.*, [1991], has long argued for a simple, conservative approach to adjudicating sources of evidence that do not permit a "specificity" ordering between them. Assume n incomparable sources of evidence yield n distinct probability intervals, $[p_i, q_i]$ for an event. Kyburg's program conservatively collapses those intervals into $[\min_i p_i, \max_i q_i]$. We use exactly this procedure to accommodate multiple inheritance in WordNet.

As an example, imagine we are estimating $P(\mathtt{NP} \overset{\mathtt{NP-PROPER}}{\longrightarrow} \overline{\mathtt{N}} \mid \mathtt{NP}[\text{alcohol}])$. Our prior interval is for the event of an `NP-PROPER` expansion given an arbitrary `NP` node. `NP-PROPER` was the chosen expansion for 2229 out of 35,382 `NP` nodes, yielding the interval $[0.0605, 0.0656]$ at 95% confidence.[14] Our first significance test is against the probability of an `NP` node

[13] The order is deterministic, but it is arbitrary in the sense that we had not intentionally coded the algorithm to present them in a particular order.

[14] This was the confidence level used for all significance tests, both in the course of parsing and in the evaluation of results described in Section 6.

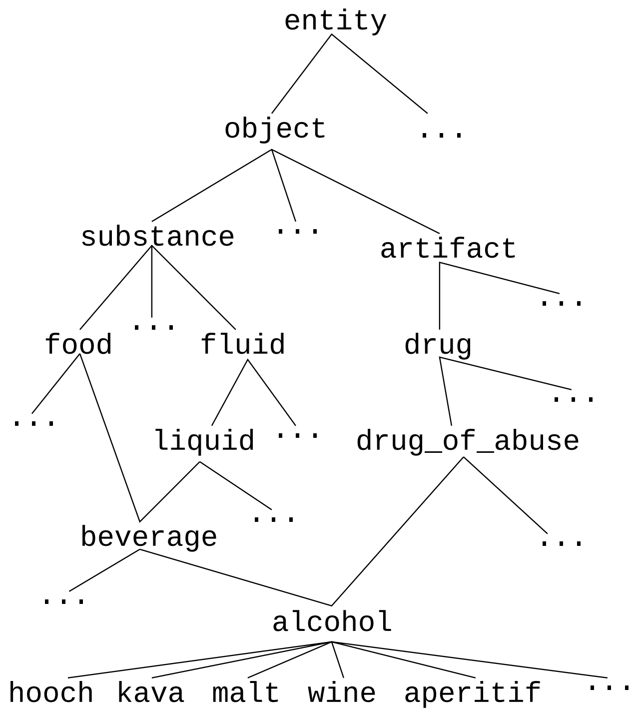

Figure 5: A fragment of the WordNet hierarchy. Each node is labeled with the first word of the WordNet synset it designates (sense numbers are not indicated).

headed by 'alcohol' (sense 1) expanding via `NP-PROPER`. We saw only 1 `NP` node headed by that sense of 'alcohol', and it did not expand via `NP-PROPER`, resulting in the interval $[0, 0.95]$, which overlaps the prior. Thus we abstract to the synset containing this sense of alcohol, which includes counts for the other members of the synset (particular senses of 'alcoholic beverage', 'drink', 'intoxicant', and 'inebriant') plus the 283 other word senses in the 209 synsets that fall under 'alcohol' in WordNet. This class has seen 26.5 `NP` nodes, and still none expanded with `NP-PROPER`, resulting in the interval $[0, 0.109]$, still overlapping. Next, we abstract to the hypernyms of sense 1 of 'alcohol'; see Figure 5 for the fragment of WordNet synsets in question. There are two hypernyms: sense 1 of 'beverage' and sense 1 of 'drug of abuse'. `NP-PROPER` was never used for any of the `NP` nodes headed by either of those two classes, which occurred 44.5 and 33.5 times, respectively, resulting in intervals $[0, 0.0658]$ and $[0, 0.0843]$. Since these reference classes are incomparable in terms of specificity, we collapse them conservatively to $[0, 0.0843]$ and still fail to reject the null hypothesis. We continue abstracting via the hypernyms, now to the classes 'food', 'liquid', and 'drug'. Here we have seen some instances of `NP-PROPER`, and the intervals we get for these three classes are, respectively, $[0.00406, 0.0562]$, $[0, 0.0320]$, and $[0.0101, 0.135]$. Again, the classes are incomparable, and the combined interval $[0, 0.135]$ overlaps the prior, so we continue to look to abstractions. 'substance', 'fluid', and 'artifact' are the direct hypernyms we next see, but we directly discount 'substance' since it is less specific than 'fluid' (see Figure 5).

Note that while the classic reference class problem is concerned with finding an *interval*-valued probability estimate, we are satisfied here with *point* probability estimates. Philosophers identifying an appropriate reference class must balance the desire for relevance against the desire for a precise estimate (*i.e.*, a narrow probability interval). Since we are content with point estimates, we can effectively ignore the precision problem. Our only desideratum is that the estimate be precise enough to warrant setting aside the uninformed estimate based on H_0. Of those classes that satisfy that desideratum, we use the most specific as the reference class for calculating the point probability estimate—if 'fluid' $\cup$ 'artifact' gives an estimate that allows us to reject H_0, then that estimate is preferred to a strictly more general class, one that contains 'substance'. This is why we discount 'substance' in this stage of the example.

Our path towards the root now stands at the nodes 'fluid' and 'artifact'. None of the 100 'fluid' NP nodes used NP-PROPER, and only 10 of the $2581\frac{51}{1260}$ 'artifact' NP nodes[15] did, giving us intervals of $[0, 0.0295]$ $[0.00211, 0.00712]$, which resolve to the former, which is indeed disjoint from the prior interval $[0.0605, 0.0656]$. Our point estimate, then, comes from the smoothed ML estimate for $P(\texttt{NP} \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}} \mid \texttt{NP}[\text{fluid}] \cup \texttt{NP}[\text{artifact}])$. Since we are in a multiple inheritance hierarchy (that is the point of this search), 'fluid' and 'artifact' do not constitute mutually exclusive events; we cannot simply add together the component probabilities. Given n reference classes $\{R_1, \ldots, R_n\}$, we can use the inclusion-exclusion formula (*e.g.*, Breslauer and Dubhashi [1995]) to write

$$P(Y \mid \bigcup_{i=1}^{n} R_i) = \frac{\sum_{I \subseteq \{1..n\}, I \neq \emptyset} (-1)^{|I|-1} |\bigcap_{i \in I} R_i \cap Y|}{\sum_{I \subseteq \{1..n\}, I \neq \emptyset} (-1)^{|I|-1} |\bigcap_{i \in I} R_i|}.$$

Having settled on 'fluid' and 'artifact' as the reference classes for this estimate, we use this formula to calculate

$$P(\texttt{NP} \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}} \mid \texttt{NP}[\text{fluid}] \cup \texttt{NP}[\text{artifact}])$$

as

$$\frac{c(\texttt{NP}[\text{fluid}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}}) + c(\texttt{NP}[\text{artifact}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}}) - c(\texttt{NP}[\text{fluid} \cap \text{artifact}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}})}{c(\texttt{NP}[\text{fluid}]) + c(\texttt{NP}[\text{artifact}]) - c(\texttt{NP}[\text{artifact} \cap \text{fluid}])}.$$

Assuming 'alcohol' is the lone common hyponym of 'fluid' and 'artifact',[16] we rewrite this as

$$\frac{c(\texttt{NP}[\text{fluid}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}}) + c(\texttt{NP}[\text{artifact}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}}) - c(\texttt{NP}[\text{alcohol}] \overset{\texttt{NP-PROPER}}{\longrightarrow} \overline{\texttt{N}})}{c(\texttt{NP}[\text{fluid}]) + c(\texttt{NP}[\text{artifact}]) - c(\texttt{NP}[\text{alcohol}])},$$

which is the expression we would expect on intuitive grounds. For this example, we have $\frac{0+10-0}{100+2581\frac{51}{1260}-26\frac{1}{2}} = 0.00377$ (before the simple smoothing described in Section 5.2).

[15] Recall from Section 5.3 that heads in multi-headed phrases share the count of that expansion, thus we often arrive at fractional counts.

[16] In fact, we make this simplifying assumption in the implementation, for all such cases of shared percentage.

6 Experimental Results

We have evaluated our model on the corpus described in Section 4.2. Recall that the SEMCOR project annotated 186 files of the Brown Corpus; it is at this level of granularity that we perform cross-validation. The parser was tested on a randomly selected set of 40 files. The test set has 1144 sentences and $14,777$ words.[17]

One note on proper nouns: SEMCOR tags all proper nouns not in WordNet with one of three WordNet tags, 'person', 'group', or 'location' (each in its first WordNet sense). We extract a record, for each instance of a proper noun tag, of the head noun of the phrase tagged (according to SAPIR) and the proper noun tag used. Thus an instance of the words 'John Smith' tagged with sense 1 of 'person' would result in a record that 'Smith' is a type of person. These proper noun records are hooked in to our internal representation of the WordNet ontology, so we can appeal to abstractions of patterns like $\langle \mathrm{smith}, \mathrm{runs} \rangle$. Note that we assume knowledge of "proper noun types" even for words in the test set; essentially, we assume that a process slightly more enriched than named entity identification has preprocessed the input to the parser.

We aim to show that incorporating WordNet into the parser improves syntactic disambiguation. Therefore, we have tested two models. One, which uses WordNet to back off semantically, operates as described in Section 5. When WordNet is not available for backing off, we need a different strategy. We rank the possible independence assumptions as follows.

1. Lexical items that are components of a joint event are in fact independent. Thus we would approximate the probability of a tuple of heads $P(\vec{h} \mid X)$ as the product of the probability of each head, $\prod_i P(h_i \mid X)$.

2. The lexical head of a syntactic expansion does not influence events already dependent on the rule used in that expansion. Thus for a lexical item w as head of an expansion using rule r, we approximate $P(X \mid w, r, Y)$ as $P(X \mid r, Y)$

3. An event (*e.g.*, a word) occurs independently of the syntactic construct it appears in. Thus, for a rule r, $P(X \mid r) \approx P(X)$.

Independence assumptions about conditioning information are made when there is *insufficient* data (the number of observed events is below a certain threshold). Independence assumptions that are about conditioned information (*i.e.*, the first assumption above) are made when there is *no* observed data.

One of the standard evaluation metrics in statistical parsing has been precision and recall of labeled bracketings, a metric proposed by Black *et al.* [1991]. While we will supply labeled bracketing results for our method, this is arguably not the most relevant method of evaluation. Since we are parsing in the context of an explicit, handwritten grammar, we are able to check not only the category label and position of a phrase (as in a labeled bracketing comparison), but also whether the phrase structure rule licensing the expansion of the phrase into its constituents is correct. This is, in fact, *necessary* to check for correctness when the

[17] As before, the words are counted as space-separated sequences of characters, so punctuation, possessives, contractions, *etc.* are not counted separately.

	Precision	Recall
Full Model	85.1%	82.5%
No Backoff	84.4%	81.6%

Table 3: Expansion precision and recall results, with and without "semantic backoff"

	Precision	Recall
Full Model	88.9%	86.2%
No Backoff	88.6%	85.6%

Table 4: Labeled bracketing precision and recall results, with and without "semantic backoff"

grammar has multiple rules with the same major category structure. Recall the discussion in Section 2 about `NP` $\rightarrow$ `NP NP` licensing both appositive constructions and "measure" noun phrases. A parser that correctly labels a string as `(NP (NP ..) (NP ..))` has not arrived at a syntactic interpretation, given the ambiguity that remains in the labeling. As our overarching goal is to build a statistical parser that facilitates semantic interpretation, and semantic interpretation is generally specific to rules, we need to find the correct rule to know that we will be able to construct the correct semantic interpretation.

Moreover, evaluating on a per-expansion basis seems like a more intuitive evaluation of a traditional, grammar-based parser than simply checking category labels. A grammar-based parser makes decisions about expanding phrases or combining them (the parser may operate top-down or bottom-up) *via phrase structure rules.* Phrases cannot be constructed irrespective of the grammar, in contrast to the non-grammar-based approaches mentioned in Section 7. Thus in evaluating our parser, we will evaluate its decisions. Collins [1999, p. 191 *ff*] similarly evaluates his parser on its dependency precision and recall, since this is how the parser constructs its analyses.

In Table 3 we show the precision and recall of the expansions found by the parser, running with and without "semantic backoff". Each expansion is a pair of position indices, a mother label, a sequence of daughter labels, and a rule identifier. Note that suboptimal pruning causes the parser to fail without finding a parse in about 1% of the test sentences, accounting for a good portion of the drop from precision to recall. We use SAPIR's pruning mechanism, but the the comparison and ranking of surviving constituents is based on our corpus probabilities.

Note also that these differences between the full model and the model that does not use WordNet for backing off are statistically significant, using two-sided tests at $\alpha = 0.05$.[18] In Table 4 we show the figures for labeled constituent precision and recall, the standard metric in reporting parsing performance. Differences there are not statistically significant. Note that all these measures are significantly better than the disambiguation that is achieved using SAPIR's existing cost structure; see Table 9 at the end of this section.

[18] We use 95% confidence level for all statistical significance tests reported here.

Recall that our language models are over word *senses* and syntactic structures; our parser thus outputs both a syntactic form and a set of word sense choices. We evaluate the Word Sense Disambiguation of the parser by calculating the percent of senses in the corpus that were guessed correctly by the parser (recall). We do not evaluate precision because many word senses are deleted in the extraction of the corpus, as described in Section 4.2, and thus the parser will often guess senses (correctly or incorrectly) for words that do not have sense tags in our version of the corpus. In Table 5 are the figures for sense recall for the two versions of the model, one that uses WordNet to back off, and one that does not. Both models perform significantly better than baseline, and the model that uses WordNet does significantly better than the other.

	Recall	Baseline
Full Model	55.8%	43.9%
No Backoff	49.4%	44.0%

Table 5: Word sense recall results, with and without semantic backoff. Baseline figures simply use the most frequent sense in our corpus, given the word and part of speech. Since the parser may choose a different part of speech under different models, the baseline may be different.

Allowing arbitrary n-tuples of lexical statistics makes our language model more subject to sparsity problems than a model that caps n at 2. Typically "treebank parsers" make use of only binary lexical statistics, relating heads to their modifiers (or more generally, dependents). This independence assumption is in fact one we make anyway when there is no data on the joint event of n lexical items. But there are very few grammar rules (by type and token) that will actually take advantage of this dependence among sisters. Noting that coordinated phrases have multiple heads, as described in Section 5.3, and discounting explicit constituents in rules (such as particular forms of punctuation), we find that 18.6% (by type) of the rules might benefit from n-ary statistics. In our corpus, however, we find that the count by *token* is just over 1%. The small percentage is due to the fact that the likely beneficiaries of daughter co-dependendence, subcategorized `VP` expansions, account for under 10% of the expansions (by token) in our corpus. Thus the 1% figure indicates that approximately 10% of our base `VP` expansions could benefit from n-ary statistics. To see whether n-ary statistics make a difference, we ran a model that always assumes that the daughters are generated independently. The figures for expansion and labeled bracketing precision and recall are in Table 6. The expansion and labeled bracketing data should be compared with those in Tables 3 and 4 respectively.

The expansion precision is seen to be lower than that of the full model by .2%, and the recall by .1%. These differences are not statistically significant, and this result is not surprising in view of the rarity of occurrences of expansions that could benefit from non-binary statistics (noted above). However, it is worth pointing out that the precision of 84.9% obtained with binary statistics is only .5% above the precision of 84.4% obtained with the "no back-off" version of the full model, and as such is not significantly better; whereas the precision of 85.1% for the full model with back-off does exceed that of the "no back-off" version by enough to be significant at the 95% confidence level. In other words,

Expansions		Labeled Bracketings	
Precision	Recall	Precision	Recall
84.9%	82.4%	88.8%	86.1%

Table 6: Results when all lexical statistics are binary: daughters are assumed to have been generated independently of one another

with binary statistics alone the value of backing off would not have become apparent, at least not under this metric with the modest-sized training and test corpora we are using.

While using WordNet for semantic backoff clearly improves both syntactic and word sense disambiguation, it may be the case that collecting data in terms of word *senses* has unnecessarily splintered the lexico-syntactic patterns we are trying to collect. To test this hypothesis, we have deleted sense information from our corpus, and trained the model on word lemmas. The language model is the same as described in Section 3, and parameters are estimated the same way as described in Section 5, using thresholding to back off. The results are in Table 7. The three metrics (other than labeled bracketing recall) are all significantly worse than the full model (results in Tables 3 and 4). Interestingly, another

Expansions		Labeled Bracketings	
Precision	Recall	Precision	Recall
84.0%	81.6%	88.1%	85.5%

Table 7: Results when all lexical statistics are in terms of words, not word senses, so no semantic backoff is employed.

statistic that we have measured, top-down recall [with heads], is also significantly worse under this non-sense approach. This metric measures the percent of time, when the parser has a correct constituent at a certain position, the constituent was expanded via the correct phrase structure rule. We also measure how often the phrase was expanded correctly when we also had the head word (sense) of the phrase correct. All models perform (statistically) significantly better with heads than without, but there are no differences among models on the same metric. The exception is in comparing the model that does not use word senses to the models that use senses and also use WordNet to back off from them. Those two models (the full model and the independent daughters model) both have 97.5% top-down recall with heads. This figure is significantly better than the 96.8% achieved by the model that uses no word senses. (However, none of the three are significantly distinct from the 97.1% recovered by the model which uses word senses but does not use WordNet to back off from them.)

We have also calculated metrics evaluating the models with respect to the number of crossing brackets in each sentence's analysis.[19] The percent of sentences with no crossings is

[19]Two brackets a and b are said to *cross* if one of a's span or b's span does not either wholly contain the other or wholly exclude the other. The *crossing brackets* metric measures the number of brackets in the calculated parse that cross a bracket in the correct parse.

not significantly different across the different models, and hovers around 58%. The percent of sentences with two or fewer crossings similarly is consistently around 79%.

An accuracy metric very relevant for enterprises in natural language understanding is the percent of sentences that are parsed [exactly] correctly. That is, for what percent of sentences in the test corpus was every constituent and every expansion calculated correctly. Our full model correctly parses 47.3% of the sentences in the test set; the model that does not use semantic backoff and n-ary statistics correctly parses 43.6% of the sentences. This difference is significant at the 90% level, but not at the 95% level we use elsewhere in these results.

As described in Section 5.1, our hypothesis testing mechanism for deciding when to back off to a particular WordNet concept outperforms the simple thresholding method used in earlier work. In Table 8 are the figures for the thresholding method. In comparing this

Expansions		Labeled Bracketings		WSD
Precision	Recall	Precision	Recall	Recall
84.0%	81.5%	88.3%	85.7%	45.8%

Table 8: Parser performance when simple thresholding is used for deciding how far to back off

simple thresholding approach to the hypothesis-testing approach, we see that all metrics are lower. Expansion precision and recall, labeled bracketing precision, and word sense recall are all lower by a statistically significant amount. Even the precision of the part of speech tagging is worse (our model guesses parts of speech right at a 97.9% clip, while this simple thresholding method guessed 97.5% right). Incidentally, we chose to set the threshold at 20 rather arbitrarily: we had used $t = 10$ in earlier work, and an inspection of results on that test set revealed 10 was too low for reliable decision making. In fact, this was the motivation for moving to a more statistically-grounded approach, via hypothesis testing. At any rate, the thresholding method at $t = 5, 10, 20, 30$ is always worse than the sigdiff method.

We noted in Section 4.2 that when the Treebank's bracketing licenses several parses, according to SAPIR, we use SAPIR's hand-coded rule costs to choose a single parse to put in the corpus. If the structures in the Treebank are *really* underspecified, our corpus will merely be a reflection, through our model, of SAPIR's existing costs. While we noted that the median number of parses SAPIR finds, per Treebank bracketing, is only 4, it is worth running SAPIR itself on the corpus; if SAPIR does very well, either the bracketings do not give us enough information, or the simple rule costs are sufficient for syntactic disambiguation! This is not the case; in Table 9 we display syntactic accuracy for SAPIR's costs. All metrics are significantly worse than our simplest model. As SAPIR's costs are designed for the sentences in airplane maintenance manuals, we really would not expect great disambiguation performance on a heterogenous corpus like the Brown Corpus. As well, most of the rules that we added to SAPIR were not costed, so in effect there is no disambiguation mechanism in place for the new rules.

Experiments we have conducted demonstrate a slight but steady improvement in syntac-

Expansions		Labeled Bracketings	
Precision	Recall	Precision	Recall
73.7%	69.4%	78.6%	74.0%

Table 9: Results of SAPIR running with its existing cost structure

tic accuracy as the size of the corpus increases. A ten-fold increase in corpus size (yielding a corpus size the same order of magnitude as Treebank's Wall Street Journal corpus) looks like it would yield syntactic accuracy measures around 90%. Word sense disambiguation, on the other hand, clearly improves with corpus size but appears likely to asymptote somewhere well below 90%. This supports the hypothesis held by many researchers that successful word sense disambiguation requires a variety of knowledge sources.

7 Related Work

As mentioned in Sections 1 and 2, many parsers have been built to do a good job at replicating the structures in the Penn Treebank. Collins [1997; 1999] goes a step further in modeling and replicating syntactic information that is not explicit in the treebank. His work gives the best published results on replicating Treebank bracketings, with labeled bracketing accuracy close to 89%. This is not much different than what we have reported here, though the experiments are vastly different: our corpus is made up of a much *smaller* set (less data to learn from) of *shorter* sentences (easier data to parse) from a *variety* of genres (harder data to parse). Further, the syntactic structures in our grammar are more detailed, and may be more difficult to learn (though expansions such as $\mathtt{NP} \rightarrow \overline{\mathtt{N}}$ are not present in the Treebank but are presumably fairly easy for us to guess). Thus it is difficult to say whether one framework is strictly easier to learn in than the other.

Recently, other researchers have combined corpus-derived statistical data with hand-written grammars; Carroll *et al.* [1998] describe their integration of verb subcategorization probabilities, derived from 10 million words of the British National Corpus, with a stochastic parser built from a handwritten grammar trained on an LR parsing model. The other parameters of their model are trained from a subset of the Brown Corpus that is the same order of magnitude as ours. Labeled bracketing accuracy is reported at around 83%, but the focus of the paper is a more informative metric for evaluating parsing: evaluating *grammatical relations*. While such an evaluation is difficult because it requires additional hand-annotation of the corpus, it is just this sort of "richer" syntactic evaluation that we are getting at by measuring accuracy of phrase structure rules in addition to the labeled bracketing.

While the "treebank parsers" mentioned in Section 2 make use of "binary" lexical statistics (where the event space is proportional to the square of the size of the lexicon) there are other parsers which do make use of n-ary statistics, as we do here. Chang *et al.* [1992] present a model, for the purposes of Machine Translation, which is capable of accommodating n-ary lexical statistics. Chang *et al.* report that there is no increase in parsing accuracy when they use lexical statistics in tuples larger than pairs, but that may be due to their small training corpus (about 1000 sentences) and limited ontology. With a richer ontology such as WordNet, we would expect to gain more from a larger corpus.

Utsuro and Matsumoto [1997b; 1997a] compile n-ary tuples of case frames for verbs. A case is either nominative, accusative, or a preposition. Each noun is one of 45,000 leaves in a Japanese hierarchical thesaurus. (This is essentially a generalization of Resnik's work [1993].) They then partition that set into independent partial subcategorization frames. In testing, a given collocation of $\langle \text{case}, \text{noun} \rangle$ pairs for a verb will be predicted based on the set of independent partial subcategorization frames for that verb. As their goal is to extract case frames for verbs, it is difficult to formulate a more general syntactic preference metric that can be used in parsing. Their evaluation focuses on several verbs and how well their model predicts the specific arguments found for those verbs in a test corpus.

Most research in Word Sense Disambiguation has focussed on a few senses of a few words. Above we reported disambiguation results for *all* content words using a *very* fine-grained dictionary. Ng and Lee [1996] report WSD results for the 200 most common nouns and verbs in English, using WordNet as the dictionary and the Brown Corpus as the data. Their disambiguator uses a myriad of features, including parts of speech of the local context, keywords in the larger context, local collocations, and the governing verb if the word in question appears to be a direct object. Their program recalls about 54% of the word senes, seven percentage points over baseline. Of course, direct comparisons to our work are not possible given the very different setup, but the figure indicates the level at which current word sense disambiguators perform when WordNet is used as the dictionary. (Contrast with the over 95% accuracy reported by Yarowsky [1995].) Our finding that word sense aids in syntactic disambiguation supports the results described by Stetina and Nagao [1997], where verb-noun-prep-noun quadruples were disambiguated in a first phase by an unsupervised algorithm, and these sense-annotated quadruples were then used to build PP-attachment classifiers. The latter outperformed classifiers that did not have knowledge of word senses and conceptual distance.

8 Conclusions and Future Work

We have shown that a linguistically well-founded, broad-coverage parser can be effectively trained on Treebank corpora, achieving 85.1% precision and 82.1% recall on expansions. (For labeled bracketings, the figures are higher.) The figures for the untrained parser are 73.7% precision and 69.4% recall, on expansions. A novel feature of the training technique is the use of head pattern statistics, *i.e.*, statistics for combinations of head word senses, and their WordNet abstractions, on the right-hand side of rules. These are motivated by a language model that regards head word senses as being generated simultaneously with phrasal constituents.

More specifically, we have shown that formulating a language model in terms of word *senses* and appealing to an ontology for backoff statistics together yield superior parsing performance compared to a parser that uses statistics of word forms. While the performance improvement is small, indications are that a larger corpus (along the lines of what has been used by others) would yield substantially better results.

Our language model is quite simple, and we have formulated some enhancements to it, including a distinction of collective vs. distributive coordination, a distance measure, gap unification, and free insertion of adverbials [Seagull, 2000b]. Implementation of this enhanced model requires some additional annotation (*e.g.*, marking collective or distributive coordination in the corpus), so this is planned as a next step. The pruning in the parser is suboptimal; approximately 1% of the sentences in our test set did not parse within the pruning parameters we used, so we may look to a different pruning procedure.

Moreover, our parser can function as a useful component of a natural language understanding system. The grammar yields interpretable syntactic structures. In fact, linguists at The Boeing Company are developing semantic translation rules for the grammar. Translating our parsed sentences into a semantic representation will allow use of deeper patterns than just combinations of head word senses, and possibly superior disambiguation. A possible approach to the use of such deeper patterns is to formulate a language model that produces semantic structures *as well as* syntactic structures and word strings.

References

[Black *et al.*, 1991] E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, B. Santorini, and T. Strzalkowski, "A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars," In *Proceedings of the Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, California, February 1991. DARPA.

[Bod *et al.*, 1996] Rens Bod, Remko Bonnema, and Remko Scha, "A Data-Oriented Approach to Semantic Interpretation," In *Proceedings of the Workshop on Corpus-Oriented Semantic Analysis, ECAI-96*, Budapest, 1996.

[Breslauer and Dubhashi, 1995] Dany Breslauer and Devdatt P. Dubhashi, "Combinatorics for Computer Scientists," available from `http://www.brics.dk/BRICS/LS/95/4/BRICS-LS-95-4/BRICS-LS-95-4.html`, August 1995.

[Caraballo, 1999] Sharon A. Caraballo, "Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text," In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126, College Park, Maryland, June 1999.

[Carroll *et al.*, 1998] John Carroll, Guido Minnen, and Ted Briscoe, "Can Subcategorisation Probabilities Help a Statistical Parser?," In *Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*, pages 1–9, Montreal, Canada, August 1998.

[Chang *et al.*, 1992] Jing-Shin Chang, Yih-Fen Luo, and Keh-Yih Su, "GPSM: A Generalized Probabilistic Semantic Model for Ambiguity Resolution," In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 177–184, 1992.

[Charniak, 1997a] Eugene Charniak, "Statistical Parsing with a Context-Free Grammar and Word Statistics," In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.

[Charniak, 1997b] Eugene Charniak, "Statistical Techniques for Natural Language Parsing," *AI Magazine*, Winter 1997.

[Chen and Goodman, 1998] Stanley Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," Technical Report TR-10-98, Harvard University, August 1998.

[Collins, 1996] Michael Collins, "A New Statistical Parser Based on Bigram Lexical Dependencies," In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, 1996.

[Collins, 1997] Michael Collins, "Three Generative, Lexicalised Models for Statistical Parsing," In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, 1997.

[Collins, 1999] Michael Collins, *Head-Driven Statistical Models for Natural Language Parsing*, PhD thesis, University of Pennsylvania, 1999.

[Fellbaum, 1998] Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[Ford *et al.*, 1982] Marilyn Ford, Joan Bresnan, and Ronald M. Kaplan, "A Competence-Based Theory of Syntactic Closure," In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations.* MIT Press, 1982.

[Grishman *et al.*, 1994] Ralph Grishman, Catherine Macleod, and Adam Meyers, "COMLEX Syntax: Building a Computational Lexicon," In *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, 1994.

[Harrison and Maxwell, 1986] Philip Harrison and Michael Maxwell, "A new implementation for GPSG," In *Proc. of the Can. Soc. for Computational Studies of Intelligence (CSCSI-86)*, pages 78–83, Quebec, 1986.

[Jelinek and Mercer, 1980] Frederick Jelinek and Robert L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, May 1980. North-Holland.

[Kasper, 2000] Walter Kasper, *Verbmobil: Foundations of Speech-to-Speech Translation*, chapter Multilingual Semantic Databases, pages 348–356, Springer, 2000.

[Kučera and Francis, 1967] Henry Kučera and W. Nelson Francis, *Computational Analysis of Present-day American English*, Brown University Press, Providence, R.I., 1967.

[Kyburg, 1991] Henry E. Kyburg, Jr., "Evidential Probability," In John Mylopoulos and Ray Reiter, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence*, volume 2, pages 1196–1202, Sydney, Australia, August 1991. Morgan Kaufmann.

[Landes *et al.*, 1998] Shari Landes, Claudia Leacock, and Randee I. Tengi, "Building Semantic Condordances," In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 8, pages 199–216. MIT Press, 1998.

[Larsen and Marx, 1986] Richard J. Larsen and Morris L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, second edition, 1986.

[Magerman, 1995] David Magerman, "Statistical Decision Tree Models for Parsing," In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, 1995.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, 19:313–330, 1993.

[Miller *et al.*, 1996] Scott Miller, Robert Bobrow, Robert Ingria, and Richard Schwartz, "A Fully Statistical Approach to Natural Language Interfaces," In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 55–61, 1996.

[Mooney, 1999] Raymond J. Mooney, "Learning for Semantic Interpretation: Scaling Up Without Dumbing Down," In *Proceedings of the ICML Workshop on Language Learning in Logic*, pages 7–14, Bled, Slovenia, June 1999.

[Ng and Lee, 1996] Hwee Tou Ng and Hian Beng Lee, "Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach," In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Santa Cruz, California, June 1996.

[Pelletier and Schubert, 1984] F. J. Pelletier and L. K. Schubert, "Two Theories for Computing the Logical Form of Mass Expressions," In *Proceedings of the 10th International Conference on Computational Linguistics*, pages 108–111, Stanford, Calif., July 1984.

[Ratnaparkhi, 1997] Adwait Ratnaparkhi, "A Linear Observed Time Statistical Parser Based on Maximum Entropy Models," In *Proceedings of the Second Conference On Empirical Methods in Natural Language Processing (EMNLP-2)*, Brown University, August 1997.

[Resnik, 1995] Philip Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada, 1995.

[Resnik, 1993] Philip Stuart Resnik, *Selection and Information: A Class-Based Approach to Lexical Relationships*, PhD thesis, University of Pennsylvania, 1993.

[Schiehlen, 2000] Michael Schiehlen, *Verbmobil: Foundations of Speech-to-Speech Translation*, chapter Semantic Construction, pages 200–215, Springer, 2000.

[Schwartz, 1999] Richard Schwartz, "Why Doesn't Natural Language Come Naturally?," Invited Talk, Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, June 1999.

[Seagull, 2000a] Amon Seagull, "A Compaction of WordNet Senses for Evaluation of Word Sense Disambiguators," Technical Report 726, University of Rochester, April 2000.

[Seagull, 2000b] Amon Seagull, *Well-Foundedness and Reliability in Statistical Natural Language Parsing*, PhD thesis, University of Rochester, 2000.

[Seagull and Schubert, 1999] Amon Seagull and Lenhart Schubert, "Guiding a Well-Founded Parser with Corpus Statistics," In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 179–186, College Park, Maryland, June 1999.

[Stetina and Nagao, 1997] Jiri Stetina and Makoto Nagao, "Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary," In Jou Shou and Kenneth Church, editors, *Fifth Workshop on Very Large Corpora*, pages 66–80, Hong Kong, August 1997. ACL/SIGDAT.

[Utsuro and Matsumoto, 1997a] Takehito Utsuro and Yuji Matsumoto, "Learning Probabilistic Subcategorization Preference and its Application to Syntactic Disambiguation," Technical Report NAIST-IS-TR97006, Graduate School of Information Science, Nara Institute of Science and Technology, Japan, February 1997.

[Utsuro and Matsumoto, 1997b] Takehito Utsuro and Yuji Matsumoto, "Learning Probabilistic Subcategorization Preference by Identifying Case Dependencies and Optimal Noun Class Generalization Level," In *Proceedings of the Fifth Applied Natural Language Processing Conference*, pages 364–371, Washington, D.C., April 1997.

[Wahlster, 2000] Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, 2000.

[Yarowsky, 1995] David Yarowsky, "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods," In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, June 1995.