Guiding a Well-Founded Parser with Corpus Statistics

Amon Seagull and **Lenhart Schubert**
Department of Computer Science
University of Rochester
Rochester, NY 14627
{seagull,schubert}@cs.rochester.edu

Abstract

We present a parsing system built from a handwritten lexicon and grammar, and trained on a selection of the Brown Corpus. On the sentences it can parse, the parser performs as well as purely corpus-based parsers. Its advantage lies in the fact that its syntactic analyses readily support semantic interpretation. Moreover, the system's hand-written foundation allows for a more fully lexicalized probabilistic model, *i.e.* one sensitive to co-occurrence of lexical heads of phrase constituents.

1 Introduction

Statistical approaches to parsing have received a great deal of attention over recent years. The availability of large tagged and syntactically bracketed corpora make the programmatic extraction of lexica and grammars feasible. Researchers have tackled parsing by substituting these automatically derived resources for hand-coded ones. While these approaches have had some success to date (Collins, 1997; Charniak, 1997a), their usability as parsers in systems for natural language understanding is suspect.[1] The reconstruction of Treebank-style bracketings does not serve as an adequate basis for semantic interpretation. The phrase structure rules are too numerous, and the analyses too coarse (especially at the lower levels) to allow association of deterministic semantic rules with phrase structure rules. Charniak himself (1997b) notes that most of the parses constructed by a "wide-coverage" grammar are "pretty senseless".

[1] Collins, Charniak, *etc.* make no claims about their programs being well suited as parsers for language understanding applications. Certainly, this type of parsing has had success to-date in applications such as Information Retrieval.

As an example, consider the flat `NP` structures that are in the Penn Treebank (Marcus et al., 1993). Nouns, determiners, and adjectives are all sisters of each other in the syntactic annotation, *e.g.* `(NP (DT the) (JJ mechanical) (NN engineering) (NN industry))`. A parser which constructs structures such as this fails to solve an ambiguity problem that has generally been considered syntactic: Are we talking about the industry of mechanical engineering, or is the entire engineering industry perceived as mechanical? If our goal is language understanding, including semantic interpretation, the Treebank bracketings must be considered *underspecified*.

We describe here a system which combines hand-coded linguistic resources with corpus-derived probabilistic information to enable (fairly) wide-coverage syntactic parsing. Most importantly, the use of these linguistic resources allows for a better-informed probabilistic model.

2 Setup

Our lexicon is composed from two resources. COMLEX (Grishman et al., 1994) provides the syntactic and morphological information for 39,000 lemmas. WordNet (Fellbaum, 1998) provides the semantic information. In addition, we add to our lexicon approximately 47,000 "multiword" nouns found in WordNet.

SAPIR, the parser we are using, employs a feature-based general grammar of English that has been in development at The Boeing Company over the past fifteen years (Harrison and Maxwell, 1986). The grammar consists of approximately 500 rules. By mapping COMLEX's lexical entries into a format understandable by SAPIR, we have a general purpose, well-founded, English language parser.

With such a parser, we can use Penn's Treebank the way it was probably intended: as a set of bracketing constraints for the syntactic analysis of a sentence. The Linguistic Data Consortium provides a preliminary version (1.075) of the Treebank's bracketing of the Brown Corpus (Kučera and Francis, 1967). Fortunately, SAPIR provides an interface whereby a sentence and a partially specified parse tree can be fed to the parser, so that only the syntactic analyses that conform with the provided bracketing will be pursued.

So it is with this mechanism that we create our corpus. We start with the bracketed (but not part-of-speech-tagged) version of the Treebank, and process the bracketings in several ways, most notably removing quotation marks and "assuaging" gaps. The latter consists primarily of identifying the Treebank's sentential constructs which are considered verb phrases by SAPIR, and making that transformation. For example, a Treebank tree like `(PP in (S (NP *) (VP going (NP home))))` would be mapped to `(PP in (VP going (NP home)))`. Since this bracketing is supplied to SAPIR as a *constraint*, the parser is free to construct the gerundive `NP` containing solely the `VP`. In fact, the bracketing corresponding to the parse found by SAPIR is:

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

(Note that postfix ^ indicates a one-bar level phrase, as per $\overline{\mathrm{X}}$-theory (Jackendoff, 1977).) Approximately 30% of the Treebank bracketings are parseable, after this assuagement, by our parser. These 30% comprise our corpus. Of course, each (parseable) bracketing does not always yield just one parse. SAPIR has some hand-coded costs on syntactic rules which have served as its preference mechanism to-date. When SAPIR finds more than one parse for a given bracketing, we simply choose its most preferred one to use in our corpus. While we certainly do not feel that this is the best way to create our corpus, we would like to note that over 25% of the parseable bracketings yield a unique parse, and over 50% have just one or two possible parses. We should also note that the parseable bracketings are, of course, shorter (on average) than the unparseable ones. The average length of all of the sentences is 17.3 words, while the average for our corpus is 11.2.

3 Language Model

We noted above that we would like a more complete lexicalization than what has been used by recent models in statistical parsing. To this end, we propose a generative model which is a direct extension of a Probabilistic Context Free Grammar (PCFG). In our model, as in a PCFG, the sentence is generated via top-down expansion of its parse tree, beginning with the root node. The crucial difference, however, is that as we expand from a nonterminal to its children, we simultaneously generate both the syntactic category *and the head word* of each child. This expansion is predicated on the category *and head word* of the mother.

We will also make the traditional assumption that all sentences are generated independently of each other. Then, under this assumption and the assumed model, we can write the probability of a particular parse tree T as the product of all its expansions, or

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

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. (Of course, all of these terms should be indexed appropriately for the expansion under consideration, but we leave that off for clarity.) Note that *rulename* usually determines X, and *rulename* and X together always determine $\vec{\mathsf{Y}}$. 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.

We can expand (1) via the chain rule:

$$\mathsf{P}(\mathsf{T}) = \prod_{expansion \in \mathsf{T}} \mathsf{P}(rulename \mid \mathsf{X}, w) \times \mathsf{P}(\vec{\mathsf{h}} \mid \mathsf{X}, w, rulename) \quad (2)$$

Note we have dropped $\vec{\mathsf{Y}}$ from the equations, since as noted above, that sequence is determined by *rulename* and X. This is an appealing rewriting, since the first term of (2), which we will term the syntactic expansion probability, corresponds neatly to the theory of Lexical Preference for those rules whose head constituent is a lexical category. Consider the following sentences, from Ford et al. (1982)

(1) a. Mary wanted the dress on that rack

b. Mary positioned the dress on that rack

LP predicts that the preferred interpretation for the first sentence is the ($_{\text{NP}}$ `the dress on that rack`) structure, while for the second, a reader would prefer the flat `V-NP-PP` structure. This follows from the theory of Lexical Preference, which stipulates that the head word of a phrase selects for its "most preferred" syntactic expansion. This is exactly what is modeled by the first term of Equation (2). "Lexical preference" has been around a long time, and the (corresponding) syntactic expansion probability we use has been used by many researchers in parsing, including many of those mentioned in this article.

The difficulty with this model, and perhaps the reason it has not been pursued to-date, is the intense data sparsity problem encountered in estimating the second term of equation (2), the lexical introduction probability. Much work in statistical parsing limits all probabilities to "binary" lexical statistics, where for any probability of the form $P(X_1, \ldots, X_n \mid Y_1, \ldots, Y_n)$, at most one of the X random variables and one of the Y random variables is lexical. By allowing "n-ary" lexical statistics, we allow an explosion of the probability space.

Nevertheless, we suggest that human parsing is responsive to the familiarity (or otherwise) of particular head patterns in rules. To combat the data sparsity, we have used WordNet to "back off" to more general semantic classes when statistics on a word are unavailable. To back off semantically, however, we need to be dealing with word *senses*, not just word forms. It is a simple refinement of our model to replace all instances of "head word" with "head sense". Additionally, a semantic concordance of a subset of the Brown Corpus and WordNet senses is available (Landes et al., 1998). Thus, our corpus, collected as described in Section 2, can be augmented to use statistics on word *senses* in syntactic constructs, after alignment of this semantic concordance (of Brown) with treebank's labeled bracketing (of Brown). Moreover, a language model that distinguishes word senses will tend to reflect the semantic as well as syntactic and lexical patterns in language, and thus should be advantageous both in training the model and in using it for parsing.

4 Estimation

We have used WordNet senses so that we might combat the data sparsity we encounter when trying to calculate the probabilities in Equation (2). Specifically, we have employed a "semantic backoff" in estimating the probabilities, where the backing off is done by ascending in the WordNet hierarchy (following its *hypernym* relations). When attempting to calculate the probability of a syntactic expansion — the probability of a category X with head sense w expanding as *rulename* — we search, breadth-first, for the first hypernym of w in WordNet which occurred with X at least t times in our training data, where t is some threshold value. So the probability $P(\textit{rulename} \mid X, w) \approx P(\textit{rulename} \mid X, p(w))$, where $p(w)$ denotes the hypernym we found.

Similarly, for the probability of lexical introduction, we abstract the tuple $\langle X, w, \textit{rulename}\rangle$ to a tuple $\langle X, p'(w), \textit{rulename}\rangle$ which occurred sufficiently often. Once this is found, we search upward, again breadth-first,[2] for some abstraction $\vec{a}(\vec{h})$ of $\vec{h}$ which appeared at least once in the context of the adequate conditioning information. Each $a(h_i)$ is some parent of the word sense h_i. The probability of each original word h_i is then conditioned on the appropriate hypernym of the found abstraction. So we approximate:

$$\begin{aligned} P(\vec{h} \mid X, w, \textit{rulename}) \approx \\ P(\vec{a}(\vec{h}) \mid \langle X, p'(w), \textit{rulename}\rangle) \\ \times \prod_i P(h_i \mid a(h_i)) \end{aligned} \tag{3}$$

Note that $p(w)$ in the first estimation may not equal $p'(w)$ in the second. Also note that backing off completely, to the `TOP` of the ontology, for the word in the conditioning information, is equivalent to dropping it from the conditioning information. Backing off completely in

[2]Breadth-first is a first approximation as the search mechanism; we intend to pursue this issue in future work.

search for the abstraction when calculating the probability of lexical introduction effectively reduces that probability to $\prod_i P(h_i)$.[3]

5 Experimental Results

We sequestered 421 sentences from our corpus of 4892 sentences (with trees and sense-tags), and used the balance for training the probabilities in equation (2). These 4892 are the parseable segment of the 16,374 trees for which we were able to "match up" the Treebank syntactic annotation with the semantic concordance. (Random errors and inconsistencies seem to account for why not all 19,843 trees align. In fact, these 19,843 themselves exclude all trees which appear to be headlines or some other irregular text. We do not, however, exclude any trees on the basis of the type of their root category. The corpus contains sentences as well as verb phrases, noun phrases, *etc.*.)

We then tested the parser varying two binary parameters:

- whether or not the semantic backoff procedure was used — If not, an unobserved conditioning event would immediately have us drop the lexical information. For example, $\langle X, w \rangle$ would immediately be backed off to simply $\langle X \rangle$.
- whether or not we simply estimated the joint probability $P(\vec{h} \mid X, w, \mathit{rulename})$ as $\prod_i P(h_i \mid X, w, \mathit{rulename})$. This we will call the "binary" assumption, as opposed to "n-ary". Effectively, it means that each daughter's head word sense is introduced independently of the others.

Tables 1 and 2 display the results for the four different settings, along with the results for a straight PCFG model (as a baseline). Note that t, our threshold parameter from above, was set to 10 for these experiments. Labeled precision and recall (Table 1) are the same as in other reports on statistical parsing: they measure how often a particular syntactic category was correctly calculated to span a particular portion of the input. Recall that our corpus was derived using a hand-crafted grammar. It makes sense, then, to add an additional criterion for correctness: we can check the actual expansions (rulenames) used and see if they were correct. This metric speaks to an issue raised by Charniak (1997b) when he notes that the rule `NP -> NP NP` has (at least) two different interpretations: one for appositive `NP`s and one for "unit" phrases like "5 dollars a share".[4] A hand-written grammar will differentiate these two constructions. Thus Table 2 shows precision and recall figures for this more strict criterion, for the four models in question plus PCFG again as a baseline. Note also that since Table 2 is for syntactic expansions, it does not include lexical level bracketings.

	Sem backoff	No sem backoff
binary	91.2/87.1	90.6/86.4
n-ary	91.3/87.8	90.1/86.2
PCFG	78.9/80.3	

Table 1: Labeled Bracketing Precision/Recall Results

	Sem backoff	No sem backoff
binary	82.5/78.8	81.3/77.6
n-ary	82.7/79.6	80.6/77.3
PCFG	65.5/66.3	

Table 2: Syntactic Expansion Precision/Recall Results

First note that the degree of improvement over baseline of even the most minimal model is approximately what other researchers, using purely corpus-driven techniques, have reported (Charniak, 1997a).

Also note that the "full" model, using both n-ary lexical statistics and semantic backoff, performs (statistically) significantly better than both of the models which do not use semantic backoff. The lone exception is that the precision of the labeled bracketings is not significantly different for the "full" model and the "minimal" model.[5]

[3] We actually stop short of this in our estimations. We search upward for the top-most nodes in WordNet, but we do not continue to the synthetic `TOP` node. Instead, we drop the lexeme from the conditioning information and restart the search.

[4] In fact there should be syntactic differences for these two constructions, since phrases like "the dollars the share" are *syntactically ill-formed* unit noun phrases.

[5] Two-sided tests were used, with $\alpha = 0.05$.

Interestingly, the "minimal" model is not significantly different from either of the two models gotten by adding *one of* n-ary statistics or semantic backoff. The improvement is only significant when both features are added.

Our results for word sense disambiguation (obtained as a by-product of parsing) are shown in Table 3. Clearly, using WordNet to back off semantically enables the parser to do a better job at getting senses right. The sense recall figures for the two models which use semantic backoff are significantly better than for those models which do not. Additionally, the improvement over baseline is significantly better for those models which use semantic backoff (11 percentage points improvement) than for those which do not (4 points better).[6]

	Sem backoff	No sem backoff
binary	40.8/51.6	40.8/45.0
n-ary	41.1/52.1	40.8/45.1

Table 3: Sense Recall: baseline/model. The baseline results are gotten by choosing the most frequent sense for the word, given the part of speech assigned by the parser. (Hence it may be different across different models for the parser.)

6 Related Work

As our framework and corpus are rather different from other work on parsing and sense disambiguation, it is difficult to make quantitative comparisons. Many researchers have achieved sense disambiguation rates above 90% (*e.g.* Gale et al. (1992)), but this work has typically focussed on disambiguating a few polysemous words with "coarse" sense distinctions using a large corpus. Here, we are disambiguating *all* words with *WordNet* senses and *not* very much data. Ng and Lee (1996) report results on disambiguating among WordNet senses for the most frequent 191 nouns and verbs (together, they account for 20% of all nouns and verbs we expect to encounter in a random selection of text). They get an improvement of 6.9 percentage points (54.0 over 47.1 percent) in disambiguating instances of these words in the Brown Corpus. Since the most frequent words are typically the most polysemous, the ambiguity problem is more severe for this subset, but there is also more data: we have about 24,000 instances of 10,000 distinct senses in our corpus, and Ng and Lee (1996) use 192,800 occurrences of their 191 words.

Carroll et al. (1998) report results on a parser, similarly based on linguistically well-founded resources, using corpus-derived subcategorization probabilities (the first term in Equation (2)). They report a significant increase in parsing accuracy, measured using a system of *grammatical relations.* Their corpus is annotated with grammatical relations like `subj` and `ccomp`, and the parser can then output these relations as a component of a parse. Carroll et al. (1998) argue that these relations enable a more accurate metric for parsing than labeled bracketing and recall. Our evaluation of phrase structure rules used in a parse is a crude attempt at this higher-level evaluation.

As mentioned above, much recent work on lexicalizing parsers has focused on binary lexical relations, specifically head-head relations of mother and daughter constituents *e.g.* (Carroll and Rooth, 1998; Collins, 1996). Some have used word classes to combat the sparsity problem (Charniak, 1997a). Link grammars allow for a probabilistic model with ternary head-head-head relations (Lafferty et al., 1992). The link grammar website reports that, on a test of their parser on 100 sentences (average length 25 words) of Wall Street Journal text, over 82% of the labeled constituents were correctly calculated.[7]

Some limited work has been done using n-ary lexical statistics. Hogenhout and Matsumoto (1996) describe a lexicalization of context free grammars very similar to ours, but without presenting a generative model. The probabilities used, as a result, ignore valuable conditioning information, such as the head word of constituent helping to predict its syntactic expansion. Nevertheless, they are able to achieve approximately

[6] The improvement gotten for moving from binary to n-ary relations, when using WordNet, is not significant. This is most likely due to the small percentage of expansions which are likely to be helped by n-ary statistics — less than 1%. In fact, there were only *seven* instances, over the 421-sentence test set, where an n-ary rule was correctly selected by the parser *and* the head of that phrase was also correctly selected. Given such small numbers, we would not expect to see a significant improvement, when using n-ary statistics, for word sense disambiguation.

[7] `http://bobo.link.cs.cmu.edu/link/improvements.html`

95% labeled bracketing precision and recall on their corpus. Note that they use a small finite number of word classes, rather than lexical items, in their statistics.

Utsuro and Matsumoto (1997) present a very interesting mechanism for learning semantic case frames for Japanese verbs: each case frame is a tuple of independent component frames (each of which may have an n-tuple of slots). Moreover, they use an ontology rather than simply word classes when finding the case frames. In this way, the work is essentially a generalization of the work of Resnik (1993). They report results on disambiguating whether a nominal argument in a complex Japanese sentence belongs to the subordinate clause verb or the matrix clause verb. Their evaluation covers three Japanese verbs, and achieves accuracy of 96% on this disambiguation task.

Chang et al. (1992) describe a model for machine translation which can accommodate n-ary lexical statistics. They report no improvement in parsing accuracy for $n > 2$. Their results most likely suffer from sparse data (they had only about 1000 sentences), although they did use semantic classes rather than lexical items. They report that their total sentence accuracy (percent of test sentences whose calculated bracketing is completely correct) is approximately 58%.

7 Future Work

There are many directions to take this work. One advantage of our well-founded framework is that it allows more linguistic information, *e.g.* features like tense and agreement, to be used in the language model.[8] For example, a verb phrase in the imperfect may often be modified by an adjunctive, durative `PP:for`. We would like to use the techniques of corpus-based parsing to extract these statistical patterns automatically. The model easily extends to incorporate a host of syntactic features (Seagull and Schubert, 1998).

[8] Note that these particular features are in theory available to a purely corpus-based parser, as part-of-speech tags in the Penn Treebank are marked for tense and agreement. But that information is not available to the phrase-level constituent unless a notion of heads and feature passing is added to the mechanism. It seems that foot features, unless explicitly realized at the phrase level (*e.g.* `WHPP`) would be even more difficult to percolate without an *a priori* notion of features and grammar.

Currently the parser uses a pruning scheme that filters as it creates the parse bottom-up. The filtering is done based on the probability of the individual nodes, irrespective of the global context. The pruning procedure needs refinement, as our full model was not able to arrive at a parse for eight of the 421 sentences in the test set.

We would certainly like to expand our corpus by increasing the coverage of our grammar. Also, adding a constituent size/distance effect, as described by Schubert (1986) and as used by some researchers in parsing (*e.g.* Lesmo and Torasso (1985) and Collins (1997)) would almost certainly improve parsing.

Most likely, WordNet senses are more fine-grained than we need for syntactic disambiguation. We may investigate methods of automatically collapsing senses which are similar. Also, we may use more data on word sense frequencies, outside of the data we get from our "parseable bracketings". We used WordNet for these experiments both because WordNet provides an ontology, and because there was an extant corpus which was annotated with both syntactic and word sense information. Using a corpus that is tagged with "coarser" senses will almost certainly yield better results, on both sense disambiguation and parsing.

8 Conclusion

This work suggests that despite their low frequency, n-ary lexical statistics can be combined with an ontology, such as WordNet, to be used to aid parsing and word sense disambiguation. More interestingly, results from our small corpus indicate that WordNet (or some ontology) *is* necessary for n-ary statistics to be useful. In addition, these results can be obtained within the framework of a well-founded grammar and lexicon. All of this together yields a broad-coverage parser that lends itself to applications requiring natural language understanding. In the future, we hope to improve our model and expand our corpus, and thus to improve our parsing accuracy further.

8.1 Acknowledgments

The grammar and parser we are using are generously supplied by The Boeing Company. Many thanks to Phil Harrison at Boeing for answering all our questions about SAPIR, and for helping to make it available in the first place. This material is based upon work supported by NSF grants IRI-9503312, IRI-9623665, and IRI-9711009.

References

Glenn Carroll and Mats Rooth. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, Granada, Spain. ACL SIDGAT.

John Carroll, Guido Minnen, and Ted Briscoe. 1998. 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.

Jing-Shin Chang, Yih-Fen Luo, and Keh-Yih Su. 1992. 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.

Eugene Charniak. 1997a. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence.*

Eugene Charniak. 1997b. Statistical techniques for natural language parsing. *AI Magazine*, Winter.

Michael John Collins. 1996. 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.

Michael John Collins. 1997. 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.

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

Marilyn Ford, Joan Bresnan, and Ronald M. Kaplan. 1982. A competence-based theory of syntactic closure. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations.* MIT Press.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439, December.

Ralph Grishman, Catherine Macleod, and Adam Meyers. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto.

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

Wide R. Hogenhout and Yuki Matsumoto, 1996. *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, chapter Training Stochastic Grammars on Semantical Categories, pages 160–172. Springer, NY.

Ray S. Jackendoff. 1977. $\overline{\text{X}}$ *Syntax: A Study of Phrase Structure.* The MIT Press, Cambridge, MA.

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

John Lafferty, Daniel Sleator, and Davy Temperley. 1992. Grammatical trigams: A probabilistic model of link grammar. In *AAAI Fall Symposium on Probablistic Approaches to Natural Language.*

Shari Landes, Claudia Leacock, and Randee I. Tengi. 1998. Building semantic condordances. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 8, pages 199–216. MIT Press.

Leonardo Lesmo and Pietro Torasso. 1985. Weighted interaction of syntax and semantics in natural language analysis. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 772–778.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Macinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Hwee Tou Ng and Hian Beng Lee. 1996. 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.

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

Lenhart K. Schubert. 1986. Are there preference trade-offs in attachment decisions? In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 601–605.

Amon B. Seagull and Lenhart K. Schubert. 1998. Smarter corpus-based syntactic disambiguation. Technical Report 693, University of Rochester, November.

Takehito Utsuro and Yuji Matsumoto. 1997. 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.