A Statistical Model for Parsing and Word-Sense Disambiguation

Daniel M. Bikel
Dept. of Computer & Information Science, University of Pennsylvania
200 South 33rd Street, Philadelphia, PA 19104-6389, U.S.A.
dbikel@cis.upenn.edu

Abstract

This paper describes a statistical model for syntactic parsing and word-sense disambiguation. We present the motivation for our combined approach to these two previously-separate areas, a detailed account of the model itself and finally some initial results.

1 Introduction

In this paper we describe a generative, statistical model for simultaneously producing syntactic parses and word senses in sentences. We begin by motivating this new approach to these two, previously-separate problems, then, after reviewing previous work in these areas, we describe our model in detail. Finally, we will present the results of our first attempt and the direction of future work.

2 Motivation for the Approach

2.1 Motivation from examples

Consider the following examples:

1. IBM bought Lotus for $200 million.
2. Sony widened its product line with personal computers.
3. The bank issued a check for $100,000.
4. Apple is expecting [$_{NP}$ strong results].
5. IBM expected [$_{SBAR}$ each employee to wear a shirt and tie].

With Example 1, the reading [IBM bought [Lotus for $200 million]] is nearly impossible, for the simple reason that a monetary amount is a likely instrument for buying and not for describing a company. Similarly, there is a resaonably strong preference in Example 2 for [$_{PP}$ with personal computers] to attach to *widened*, on account of the semantic fact that personal computers are products with which a product line could be widened. As pointed out by (Stetina and Nagao, 1997), word sense information can be a proxy for the semantic- and world-knowledge we as humans bring to bear on attachment decisions such as these.

Conversely, both the syntactic and semantic context in Example 3 let us know that *bank* is not a river bank and that *check* is not a restaurant bill. In Examples 4 and 5, knowing that the complement of *expect* is an NP or an SBAR provides information as to whether the sense is "await" or "require". Thus, Examples 3–5 illustrate how the syntactic context of a word can help determine its meaning.

2.2 Motivation from previous work

2.2.1 Parsing

In recent years, the success of statistical parsing techniques can be attributed to several factors, one of the most important of which is the use of bilexical dependencies ((Magerman, 1995), (Ratnaparkhi, 1997), (Collins, 1996), (Collins, 1997), (Charniak, 1997)). Even more crucially, the bilexical dependencies involve head-modifier relations (or simply, "head relations"). The intuition behind the lexicalization of a grammar formalism is to capture lexical items' idiosyncratic parsing preferences. The intuition behind using heads as the members of the bilexical relations is twofold. First, many linguistic theories tell us that the head of a phrase projects the skeleton of that phrase, to be filled in by specifiers, complements and adjuncts; such a notion is captured quite directly by a formalism such as Lexicalized Tree-Adjoining Grammars (Joshi and Schabes, 1997). Second, the head of a phrase usually conveys some large component of the semantics of that phrase.[1] In this way, using head-relation statistics encodes a bit of

[1] Heads originated this way, but it has become necessary to distinguish semantic heads, such as nouns and verbs, from functional heads, such as determiners, INFL's and complementizers. In this paper, we almost always intend "head" to mean "semantic head".

the predicate-argument structure in the syntactic model.

Another motivation for incorporating word senses into a statistical parsing model has been to alleviate the bain of statistical parsing methods, sparse data. Inspired by the PP-attachment work of (Stetina and Nagao, 1997), we use WordNet v1.6 (Miller et al., 1990) as our semantic dictionary, where the hypernym structure provides the basis for semantically-motivated soft clusters.

2.2.2 Word-sense disambiguation

While there has been much work in this area, let us examine the features used in recent statistical approaches. (Yarowsky, 1992) uses wide "bag-of-words" contexts with a naive Bayes classifier. (Yarowsky, 1995) also uses wide context, but incorporates the one-sense-per-discourse and one-sense-per-collocation constraints, using an unsupervised learning technique. The supervised technique in (Yarowsky, 1994) has a more specific notion of context, employing not just words that can appear within a window of $\pm k$, but crucially words that abut and fall in the ± 2 window of the target word. More recently, (Lin, 1997) has shown how syntactic context, and dependency structures in particular, can be successfully employed for word sense disambiguation. Finally, (Stetina and Nagao, 1997) have shown that by employing a fairly simple and somewhat ad-hoc unsupervised method of WSD using a WordNet-based similarity heuristic, they could enhance PP-attachment performance to a significantly higher level than systems that made no use of lexical semantics (88.1% accuracy).

3 The Model

3.1 Overview

The parsing model we started with was extracted from BBN's SIFT system (Miller et al., 1998), which we briefly present again here, using examples from Figure 1 to illustrate the model's parameters. The BBN model is closely derived from Model 2 of (Collins, 1997).[2]

The model generates the head of a constituent first, then each of the left- and right-modifiers, generating from the head outward, using a bigram model of node labels. More formally, the lexicalized PCFG that sits behind the parsing model has rules of the form

$$P \rightarrow L_n L_{n-1} \cdots L_1 H R_1 \cdots R_{n-1} R_n \qquad (1)$$

where P, H, L_i and R_i are all lexicalized nonterminals, *i.e.*, of the form $Y\langle w,t,f\rangle$, where Y is a traditional CFG nonterminal and $\langle w,t,f\rangle$ is the word–part-of-speech–word-feature triple that is the head of the phrase denoted by Y.[3] The lexicalized nonterminal H is so named because it is the *head constituent*, where P inherits its head triple from this head constituent. The constituents labeled L_i and R_i are left- and right-modifier constituents, respectively.

3.2 Probability structure of the original model

We use the lower-case of identifiers that appear in (1) to refer to the de-lexicalized nonterminal labels.[4] We now present the top-level generation probabilities, along with examples from Figure 1. Due to space constraints, we omit the smoothing details of BBN's model (see (Miller et al., 1998) for a complete description); we note that all smoothing weights are computed via the technique described in (Bikel et al., 1997).

As with most lexicalized PCFG–derived models, all trees have an implicit +TOP+ node; this is a convenient mechanism to ensure the grammar is *consistent*, that is, that the probabilities of all possible trees sum to 1.

$$P(p \mid +\text{TOP}+)\text{, } e.g.,\ P(S \mid +\text{TOP}+) \qquad (2)$$

$$P(h \mid p),\ e.g.,\ P(VP \mid S) \qquad (3)$$

$$P_L(l_i \mid l_{i-1}, p, h, w_h),\ e.g., \qquad (4)$$
$$P_L(NP \mid ADVP, S, VP, \text{caught})$$

(when generating the NP for NP(boy–NN)) and

$$P_R(r_i \mid r_{i-1}, p, h, w_h),\ e.g., \qquad (5)$$
$$P_R(NP \mid +\text{BEGIN}+, VP, VBD, \text{caught})$$

[2] For those intimately familiar with Model 2 of (Collins, 1997), the primary differences are that the BBN model does not make argument/adjunct distinctions, nor does it use subcat frames, nor does it use the distance metric or have separate punctuation or coordination parameters. Instead, the BBN model simply uses bigram probabilities when generating modifier nonterminals (Model 2 of (Collins, 1999) uses this type of dependency, but only within its BaseNP model).

[3] The word feature is a vector of orthographic and morphological features of the word and is computed deterministically at run-time. The inclusion of the word feature in the BBN model was due to the work described in (Weischedel et al., 1993), where word features helped reduce part of speech ambiguity for unknown words.

[4] That is, p will denote the single nonterminal that forms the *LHS* of a rule (the *parent* constituent), h will denote the head nonterminal of the *RHS* of a rule and l_i and r_i will denote left- and right-modifier nonterminals. Additionally, we will denote head words of head constituents w_h, head words of modifier constituents w_{l_i}, w_{r_i} and similarly for part of speech tags (t_h, t_{l_i}, t_{r_i}) and word features (f_h,f_{l_i},f_{r_i}).

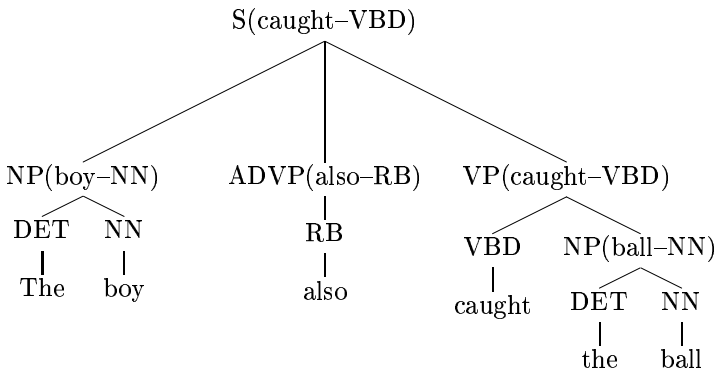

Figure 1: A sample sentence with parse tree.

(when generating the NP for NP(ball–NN)).[5]

The probabilities for generating lexical elements (part-of-speech tags, words and word features) are as follows. The part of speech tag of the head of the entire sentence, t_h, is computed conditioning only on the top-most symbol p:

$$P(t_h \mid p). \tag{6}$$

Part of speech tags of modifier constituents, t_{l_i} and t_{r_i}, are predicted conditioning on the modifier constituent l_i or r_i, the tag of the head constituent, t_h, and the word of the head constituent, w_h:

$$P(t_{l_i} \mid l_i, t_h, w_h) \text{ and } P(t_{r_i} \mid r_i, t_h, w_h). \tag{7}$$

The head word of the entire sentence, w_h, is predicted conditioning only on the top-most symbol p and t_h:

$$P(w_h \mid t_h, p). \tag{8}$$

Head words of modifier constituents, w_{l_i} and w_{r_i}, are predicted conditioning on all the context used for predicting parts of speech in (7), as well as the parts of speech themsleves:

$$P(w_{l_i} \mid t_{l_i}, l_i, t_h, w_h) \quad \text{and} \quad P(w_{r_i} \mid t_{r_i}, r_i, t_h, w_h). \tag{9}$$

The word feature of the head of the entire sentence, f_h, is predicted conditioning on the top-most symbol p, its head word, w_h, and its head tag, t_h:

$$P(f_h \mid w_h, t_h, p). \tag{10}$$

Finally, the word features for the head words of modifier constituents, f_{l_i} and f_{r_i}, are predicted conditioning on all the context used to predict modifier head words in (9), as well as the modifier head words themselves:

$$P(f_{l_i} \mid \text{known}(w_{l_i}), t_{l_i}, l_i, t_h, w_h) \quad \text{and} \quad P(f_{r_i} \mid \text{known}(w_{r_i}), t_{r_i}, r_i, t_h, w_h). \tag{11}$$

The function known(x) is a predicate that returns *true* if word x was observed during training, *false* otherwise.

The probability of an entire parse tree is the product of the probabilities of generating all of the elements of that parse tree, where an element is either a constituent label, a part of speech tag, a word or a word feature. All probabilities are computed via maximum-likelihood estimates using frequencies gathered from the training data.

4 Word-sense Extensions to the Lexical Model

The desired output structure of our combined parser/word-sense disambiguator is a standard, Treebank-style parse tree, where the words not only have parts of speech, but also WordNet synsets.[6] Incorporating synsets into the lexical part of the model is fairly straightforward: a synset is yet another element to be generated. The question is when to generate it. The lexical model has decomposed the generation of the $\langle w, t, f \rangle$ triple into three steps, each conditioning on all the history of the previous step. While it is probabilistically identical to predict synsets at any of the four possible points if we continue to condition on all the history at each step, we would like to pick the point that is most well-founded both in terms of the underlying linguistic structure and in terms

[5] The hidden nonterminal +BEGIN+ is used to provide a convenient mechanism for determining the initial probability of the underlying Markov process generating the modifying nonterminals; the hidden nonterminal +END+ is used to provide consistency to the underlying Markov process, *i.e.*, so that the probabilities of all possible nonterminal sequences sum to 1.

[6] A *synset* is WordNet parlance for a set of synonymous words in the WordNet database. Synsets have various relations among them in the database, the most important of which in the context of the present work is the *hypernym relation*, which is corresponds to the IS-A relation in other knowledge hierarchies. The inverse relation is the *hyponym relation*.

of what can be well-estimated. In Section 2.2.1 we mentioned the soft-clustering aspect of synsets; in fact, they have a duality. On the one hand, they serve to add specificity to what might otherwise be an ambiguous lexical item; on the other, they are *sets*, clustering lexical items that have similar meanings. Even further, noun and verb synsets form a *concept taxonomy*, the hypernym relation forming a partial ordering on the lemmas contained in WordNet. The former aspect corresponds roughly to what we as human listeners or readers do: we hear or see a sequence of words in context, and determine incrementally the particular meaning of each of those words. The latter aspect corresponds more closely to a mental model of generation: we have a desire or intention to convey, we choose the appropriate concepts with which to convey it, and we realize that desire or intention with the most felicitous syntactic structure and lexical realizations of those concepts. As this is a generative model, we generate a word's synset after generating the part of speech tag but *before* generating the word itself.

The synset of the head of the entire sentence, s_h is predicted conditioning only on the top-most symbol p and the head tag, t_h:

$$P(s_h \mid t_h, p). \tag{12}$$

We accordingly changed the probability of generating the head word of the entire sentence to be

$$P(w_h \mid s_h, t_h, p). \tag{13}$$

The probability estimates for (12) and (13) are not smoothed.

The probability model for generating synsets of modifier constituents m_i, complete with smoothing components, is as follows:

$$\begin{aligned}\hat{P}(s_{m_i} \mid t_{m_i}, m_i, w_h, s_h) = \quad & (14)\\ \lambda_0 \hat{P}(s_{m_i} \mid t_{m_i}, m_i, w_h, s_h) & \\ + \ \lambda_1 \hat{P}(s_{m_i} \mid t_{m_i}, m_i, s_h) & \\ + \ \lambda_2 \hat{P}(s_{m_i} \mid t_{m_i}, m_i, @^1(s_h)) & \\ + \ \cdots & \\ + \ \lambda_{n+1} \hat{P}(s_{m_i} \mid t_{m_i}, m_i, @^n(s_h)) & \\ + \ \lambda_{n+2} \hat{P}(s_{m_i} \mid t_{m_i}, m_i) & \\ + \ \lambda_{n+3} \hat{P}(s_{m_i} \mid t_{m_i}) &\end{aligned}$$

where $@^i(s_h)$ is the i^{th} hypernym of s_h. The WordNet hypernym relations, however, do not form a tree, but a directed acyclic graph, so whenever there are multiple hypernyms, the uniformly-weighted mean is taken of the probabilities conditioning on each of the hypernyms. That is,

$$\begin{aligned}\hat{P}(s_{m_i} \mid t_{m_i}, m_i, @^j(s_h)) = \quad & (15)\\ \frac{1}{n} \sum_{k=1}^{n} \hat{P}(s_{m_i} \mid t_{m_i}, m_i, @_k^j(s_h)) &\end{aligned}$$

when $@^j(s_h) = \{@_1^j(s_h), \ldots, @_n^j(s_h)\}$.

Note that in the first level of back-off, we no longer condition on the head word, but strictly on its synset, and thereafter on hypernyms of that synset; these models, then, get at the heart of our approach, which is to abstract away from *lexical* head relations, and move to the more general *semantic* relations, here represented by synset relations.

Now that we generate synsets for words using (14), we can also change the word generation model to have synsets in its history:

$$\begin{aligned}\hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i, w_h, s_h) = \quad & (16)\\ \lambda_0 \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i, w_h) & \\ + \ \lambda_1 \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i, s_h) & \\ + \ \lambda_2 \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i, @^1(s_h)) & \\ + \ \cdots & \\ + \ \lambda_{n+1} \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i, @^n(s_h)) & \\ + \ \lambda_{n+2} \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}, m_i) & \\ + \ \lambda_{n+3} \hat{P}(w_{m_i} \mid s_{m_i}, t_{m_i}) & \\ + \ \lambda_{n+4} \hat{P}(w_{m_i} \mid s_{m_i}) &\end{aligned}$$

where once again, $@^i(s_h)$ is the i^{th} hypernym of s_h. For both the word and synset prediction models, by backing off up the hypernym chain, there is an appropriate conflation of similar head relations. For example, if in training the verb phrase [strike the target] had been seen, if the unseen verb phrase [attack the target] appeared during testing, then the training from the semantically-similar training phrase could be used, since this sense of *attack* is the hypernym of this sense of *strike*.

5 Training

Ideally, there would be a gold standard corpus that contained both syntactic and word sense information. Alas, such a combined corpus does not exist, and developing one by hand or even by bootstrapping would be quite expensive, as evinced by the labor required to produce the Penn Treebank and other, similar parsing resources that were produced by correcting automatic parses. Instead, we have a gold standard syntactic corpus, in the form of the Penn Treebank (Marcus et al., 1993), and we have a completely disjoint word sense corpus, SemCor (Miller et al., 1994),[7] with a reported inter-annotator agreement of 78.6% overall, and as low as 70% for words with polysemy of 8 or above (Fellbaum et al., 1998). This makes training a

[7] SemCor is a 455k-word section of the Brown Corpus where each noun, verb, adjetive and adverb has been human-annotated with a WordNet synset.

non-trivial issue. We deal with it by adopting a technique very loosely inspired by the Co-Training framework (Blum and Mitchell, 1998). In that paper, the application of the framework was to the classification of HTML documents, where a document was seen to have two "views", one defined by the bag of words in all hyperlinks to the document, the other the bag of words in the document itself. In our case, we consider a word to have two "views", the first being the lexicosyntactic relationship with its modifiers given in the Penn Treebank, and the second being the synsets of its modifiers given in SemCor. We would like to co-train the synset- and word-generation probability models to be consistent with the contexts provided in both Treebank and SemCor. To this end, we employ the following expedient—albeit non-rigorous—method:

1. Train the syntactic parsing model on the Penn Treebank.

2. Parse the unannotated portion of the Brown corpus that comprises SemCor. This step is an automatic means to add syntactic information to SemCor.

3. Merge the synset information of SemCor into the parse trees from Step 2. This step yields a corpus with human-annotated word senses and machine-generated parse trees.[8]

4. Train the combined parsing/word sense disambiguation model on this merged corpus.

5. Using the combined model, parse all of the training portion of Treebank, constraining all output parses to have the correct bracketing and part of speech tags. This step adds word senses to all the content words in the Penn Treebank, yielding a corpus with human-annotated parse trees and machine-generated word senses.

6. Test syntactic parsing performance on a held-out Treebank development test set (Section 22).

7. Using the combined model, parse all of SemCor, constraining all word senses to be those annotated.

8. Repeat from Step 5, stopping at Step 6 when performance reaches a local maximum, or when time runs out.[9]

The idea is that at each iteration, Step 5 constrains the syntactic parameters of the combined model using human-annotated training, and Step 7 constrains the word-sense parameters of the combined model using human-annotated training, with the hope of converging on a combined model that is "optimal" for both syntax and word-sense. The reader may notice that Step 6 has no analogous test in the word sense disambiguation domain. This is not an oversight; rather, syntax, in addition to being more well understood than word sense, has well-established testing metrics and a more consistent gold standard (Treebank) than does word sense (SemCor).

6 Decoding

Even though the model is a top-down, generative one, parsing proceeds bottom-up. The model is searched via a modified version of CKY,[10] where candidate parse trees that cover the same span of words are ranked against each other. Two forms of pruning are employed: a beam is applied to each cell in the chart, pruning away all parses whose ranking score is not within a factor of $e^{-5} \approx 0.0067$ of the top-ranked parse, and only the top-ranked 25 subtrees are maintained, and the rest are pruned away.

7 Initial Results

We only had time to perform two iterations of the ad-hoc but computationally-expensive "co-training" procedure; this yielded no significant change in parsing performance of the model (which was: labeled R83.28, P84.06 on the 1574 sentences of length ≤ 40 in our development test set, Section 22 of the Penn Treebank). In addition

[8] Here we were forced to make two decisions. First, SemCor allows multiple synsets to be assigned to a particular word; in these cases, we simply discard all but the first assigned synset. Second, WordNet has collocations, whereas Treebank does not. To deal with this disparity, we re-analyze annotated collocations as a sequence of separate words that have all been assigned the same synset as was assigned the collocation as a whole. This is not as unreasonable as it may sound; for example, `vice_president` is a lemma in WordNet and appears in SemCor, so the merged corpus has instances where the word *presdient* has the synset `vice_president_1`, but only when preceded by the word *vice*. The drawback is that this introduces additional polysemy for lemmas that happen to comprise WordNet collocations.

[9] While this procedure is expedient from an engineering standpoint, it is rather compute-intensive.

[10] The Cocke-Kasami-Younger parsing algorithm is a dynamic programming approach to parsing CFG's in Chomsky Normal Form that is guaranteed to deliver all possible parses in $O(n^3)$ time, where n is the number of terminals.

to having so few iterations of the training procedure, we believe that bugs may still remain in our code to account for lack of parsing improvement. Also the default beam width of the BBN model may have been too narrow for the combined model. The model does however produce what appear to be plausible synset assignments to words. Trained on the first 99% of the articles that comprise SemCor and tested on the last 198 sentences, it has a recall of 76.9% and a precision of 63.2% for exact synset matches overall, for all four WordNet parts of speech (nouns, verbs, adjectives and adverbs).[11] Since generalized word sense disambiguation is a new problem, this should be considered a baseline, and we are currently investigating proper evaluation metrics for WordNet-sense assignment, such as the information-theoretic metric proposed in (Lin, 1997).

After the subsmission of this paper, we have obtained newer results using a different training/test set and a different training procedure; please see `http://www.cis.upenn.edu/~dbikel/wn-parsing-aug-2000.ps` for a full description of these newer results.

8 Future Work

This paper represents a first attempt at a combined parsing/word sense disambiguation model. Although it has been very useful to work with the BBN model, we are currently implementing and hope to augment a more state-of-the-art model, *viz.*, Model 2 of (Collins, 1997). We would also like to explore the use of a more radical model, where nonterminals *only* have synsets as their heads, and words are generated strictly at the leaves. Additionally, we would like to incorporate more semantically-motivated dependencies in the model, such as trilexical or even tetralexical dependencies among heads. For example, whereas PP attachment methods such as (Stetina and Nagao, 1997) and (Collins and Brooks, 1995) use quadruples of head words, the model presented here never predicts a head conditioning on more than simply the parent head—a bad independence assumption for the purpose of PP attachment. Finally, we would like to investigate the incorporation of unsupervised methods for WSD, such as the heuristically-based boostrapping method of (Stetina and Nagao, 1997) and the theoretically purer bootstrapping method of (Yarowsky, 1995). Bolstered by the success of (Stetina and Nagao, 1997) and (Lin, 1997), we believe there is great promise the incorporation of word-sense into a probabilistic parsing model.

[11] The low recall appears to be due to the large number of unknown words in the test data not getting synsets (including proper names), as well as function words (such as "of") that appear in collocations (see footnote 8). This precision number is also unfortunately not comparable to those from SensEval, as it does include some monosemous words.

9 Acknowledgements

I would like to greatly acknowledge the gracious researchers at BBN who allowed me to use and abuse their parser: Scott Miller, Lance Ramshaw, Heidi Fox, Sean Boisen and Ralph Weischedel. I would also like to thank my advisor Mitch Marcus for his invaluable technical advice and support.

References

Daniel M. Bikel, Richard Schwartz, Ralph Weischedel, and Scott Miller. 1997. Nymble: A high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201, , Washington, D.C.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training (extended version). In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Menlo Park. AAAI Press/MIT Press.

M. Collins and J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Third Workshop on Very Large Corpora*, pages 27–38.

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 Assocation for Computational Linguistics and 8th Conference of the European Chapter of the Assocation of Computational Linguistics*, pages 16–23.

Michael John Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania.

Christiane Fellbaum, Jaochim Grabowski, and Shari Landes. 1998. Performance and confidence in a semantic annotation task. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 9. MIT Press, Cambridge, Massachusetts.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In A. Salomma and G. Rosenberg, editors, *Handbook of Formal Languages and Automata*, volume 3, pages 69–124. Springer-Verlag, Heidelberg.

Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain.

D. Magerman. 1995. Statistical decision tree models for parsing. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, Massachusetts. Morgan Kaufmann Publishers.

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

George A. Miller, Richard T. Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine J. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop.*

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 1998. SIFT – Statistically-derived Information From Text. In *Seventh Message Understanding Conference (MUC-7)*, Washington, D.C.

Adwait Ratnaparkhi. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Brown University, Providence, Rhode Island.

Jiri Stetina and Makoto Nagao. 1997. Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Fifth Workshop on Very Large Corpora*, pages 66–80, Beijing.

R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. 1993. Coping with ambiguity and unknown words through probabilistic methods. *Computational Linguistics*, 19(2):359–382.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Fourteenth International Conference on Computational Linguistics (COLING)*, pages 454–460.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Assocation for Computational Linguistics*, pages 88–95.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.