A Lattice-Based Framework for Enhancing Statistical Parsers with Information from Unlabeled Corpora

Michaela Atterer
Institute for NLP
University of Stuttgart, Germany
atterer@ims.uni-stuttgart.de

Hinrich Schütze
Institute for NLP
University of Stuttgart, Germany
hinrich@hotmail.com

Abstract

Great strides have been made in building statistical parsers trained on annotated corpora such as the Penn treebank. However, recently performance improvements have leveled off. New information sources need to be considered to make further progress in parsing. In this paper, we propose a new method of using unlabeled corpora for improving syntactic disambiguation. The method is tested on the problem of relative clause attachment with encouraging results.

1 Introduction

Great strides have been made in building statistical parsers trained on annotated corpora such as the Penn treebank (Marcus et al., 1993). However, recently performance improvements have leveled off (Bikel, 2004; Collins and Koo, 2005; Klein and Manning, 2003; Charniak and Johnson, 2005). New information sources need to be considered to make further progress in parsing. One information source that is available in virtually unlimited quantity is unlabeled text. As a large body of work on unsupervised learning from corpora has shown, there is valuable syntactic and semantic information in natural language even if it is unlabeled. We propose to combined supervised and unsupervised learning for syntactic disambiguation as sketched in Figure 1. In the supervised phase, a probabilistic parser is trained on a labeled corpus. In the unsupervised phase, the parser is enriched with information from a large unlabeled corpus.

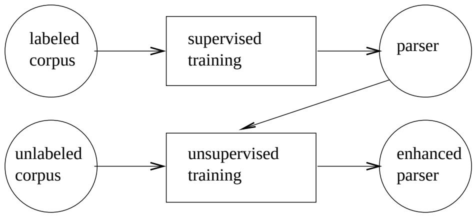

Figure 1: The proposed framework for combining supervised and unsupervised learning.

Exploiting unlabeled resources is of particular importance when training sets are small. Training sets are expensive and thus a major obstacle for broad deployment of statistical NLP methods. Statistical methods have to be adapted to new languages and new domains (e.g., a parser trained on WSJ will not work well on manuals). In many practical settings, training sets available during adapation will be small due to the high cost of training set creation. This motivates us to study the effect of *training set size* on the performance of the method proposed here. Since training sets cannot be assumed to be large in general, it is important to investigate whether methods are still applicable when training sets are smaller than the standard sets used in the research community.

There is a long tradition of using structural analysis of unlabeled corpora for syntactic disambiguation (e.g., (Hindle and Rooth, 1991)). One of the contributions of this paper is a general framework for using unsupervised acquisition of lexical information for structural dis-

Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X),
pages 93–100, New York City, June 2006. ©2006 Association for Computational Linguistics

ambiguation. This framework is based on lexical dependencies because they are mostly local and can therefore be extracted reliably from unlabeled text. At the same time, these extracted dependencies can be easily incorporated into the trained parser. Dependencies are thus well-suited to serve as the common currency for integrating information in combined supervised and unsupervised learning.

2 Structural Disambiguation

Conceptually, we would like to factor the parsing problem into decisions that can be made on purely structural grounds (e.g., recognition of base NPs) and more difficult attachment decisions, in particular those that require world knowledge, e.g. in Example 1

(1) Mr. Baker found [an opening] under [the house] that *led to* a fume-filled coal mine.

Does the opening lead to the coal mine, or does the house? We make the simplifying assumption that "semantic" attachment decisions are independent of each other. This is often the case on a purely syntactic level although it is clearly not true semantically since semantically inconsistent attachments can give rise to incoherent readings.

We formalize an attachment ambiguity as a phrase XP having two or more possible attachment points $i_1, i_2, \ldots$ in a sentence S. Let R be the parse of a sentence S with XP removed. To make an attachment decision, we form triples of the form $< \mathrm{R}, i, \mathrm{XP} >$ where i is a possible attachment node for XP in R. We define a set of generalization functions $G = \{g_j\}$ that map triples into more general triples. Some functions simply delete material, e.g., the subject of the sentence. Others replace nouns with their classes, e.g., "Canada" with "country". Each g_j modifies either R or XP, but not both. The functions can be applied in any order. Functions g_j that would delete the node i are not permitted.

We define a subsumption relationship $\subseteq$ on the set of triples produced from $< \mathrm{R}, i, \mathrm{XP} >$: $< T_1, i, Y_1 > \subseteq < T_2, i, Y_2 >$ iff $T_1 \subseteq T_2$ and $Y_1 \subseteq Y_2$, where a phrase structure tree P_1 is subsumed by P_2 iff the nodes of P_1 can be mapped onto P_2 preserving dominance and if nodes are mapped onto identical nodes or more specific nodes (e.g., "country" onto "Canada"). All g_j obey the constraint $g_j(< \mathrm{R}, i, \mathrm{XP} >) \subseteq < \mathrm{R}, i, \mathrm{XP} >$.

Triples are evaluated by an evaluation function ϕ that assesses the support of the lexical relationships in the triple in the unlabeled corpus C: $\phi(< \mathrm{R}, i, \mathrm{XP} >) \in \mathcal{R}$. Generalization is necessary because the particular set of words found in a sentence will rarely occur in C – and even if it does we don't know what the correct parse of the sentence is. The functions g_j produce a series of more and more abstract triples so as to guarantee that C contains enough data for evaluation.

The measure we use here to evaluate triples is pointwise mutual information with respect to an unlabeled corpus C. We define:

$$\begin{aligned} \phi(< \mathrm{T}, i, \mathrm{Y} >) &= MI(< \mathrm{T}, i, \mathrm{Y} >) \\ &= \log_2 \frac{P(< \mathrm{T}, i, \mathrm{Y} >)}{P(\mathrm{T})P(\mathrm{Y})} \end{aligned}$$

$$\text{for } P(< \mathrm{T}, i, \mathrm{Y} >), P(\mathrm{T}), P(\mathrm{Y}) \neq 0$$

$$\phi(< \mathrm{T}, i, \mathrm{Y} >) = 0 \qquad \text{otherwise}$$

where the probabilities are estimated on the unlabeled corpus C. $P(\mathrm{T})$ and $P(\mathrm{Y})$ are the probabilities of dependency structures T and Y occurring in C and $P(< \mathrm{T}, i, \mathrm{Y} >)$ is the probability of the dependency structures of T and Y, with Y attached at node i in T, occurring in C.

The set of triples $Q(< \mathrm{R}, i, \mathrm{XP} >, G)$ derived from $< \mathrm{R}, i, \mathrm{XP} >$ by successive applications of one, two or more generalization functions $g_j \in G$ forms a lattice with respect to $\subseteq$. $< \mathrm{R}, i, \mathrm{XP} >$ is the supremum and $< \emptyset, i, \emptyset >$ the infimum of this lattice. An example of such a lattice is shown in Figure 2 (see below for more detailed discussion). $\phi(< \emptyset, i, \emptyset >)$ is defined as a constant, which depends on the disambiguation task at hand. We take advantage of the lattice structure to compute the *affinity* A between R and XP which expresses to what extent attachment of XP in R at node i is supported by lexical dependencies in C. We propose three different definitions of A:

- The maximum with respect to $<$ on $\mathcal{R}$: $A_< = \max_<(\{\phi(q) | q \in Q\})$

- The sum over the lattice: $A_\Sigma = \sum_{q \in Q} \phi(q)$
- The MI of the maximum with respect to $\subseteq$: $A_\subseteq = \phi(\max_\subseteq(\{q | q \in Q, \phi(q) \neq 0\}))$ (if there are several maximal q, we take the average of their MI values)

Intuitively, we are searching for evidence in C that XP and R fit well together like a key and a lock. Affinity measure $A_<$ selects the best fitting generalization of the triple whereas A_Σ considers the joint evidence of all triples. Maximum and sum can only be computed if the lattice is small. Measure $A_\subseteq$ has the advantage of circumventing the need of computing the entire lattice. We move down from the original triple until we find a "layer" of the lattice where probabilities are not zero. In this paper, we only report results for $A_<$.

The actual syntactic disambiguation is performed by comparing the affinities $A(Q(<\mathrm{R}, i_k, \mathrm{XP}>))$ for the possible attachment nodes $i_1, i_2, \ldots$ and selecting the node with the highest affinity.

3 Experimental Setup

When computing the mutual information of an attachment constellation, the required probabilities are estimated based on dependency parses of the unlabeled corpus produced by Minipar (Lin, 1998), a dependency parser that recognizes a wide range of dependencies. We use the Reuters RCV1 corpus (Lewis et al., 2004) as our unlabeled corpus. The first 50 weeks (about 80,000,000 words) were parsed with Minipar and dependencies stored in an inverted index for easy querying. The inverted index is implemented using Lucene (Lucene, 2006). This setup enables searching for the frequency of lexical dependencies. For example, we can query for the number of times that *cat* was the subject of *chase*, and we can estimate the probabilities $P(T)$, $P(Y)$, and $P(< T, i, Y >)$ as relative frequencies by counting the number of times the corresponding dependency structures occur in the corpus. A constellation (T, Y, or $< T, i, Y >$) is first represented as a dependency structure and, for reasons of efficiency, the number of occurrences of this dependency structure is then approximated as the number of sentences that contain all binary dependencies in the structure. We take a trained parser (Minipar[1] or the Collins parser, depending on the experiment), run it on Penn Treebank sentences, search for the type of attachment ambiguity we are interested in and, if it occurs, present two triples of the form $<\mathrm{R}, i, \mathrm{XP}>$ and $<\mathrm{R}, j, \mathrm{XP}>$ to the disambiguation component, where i and j are two possible attachment sites for XP in R. Sections 00–12 of the WSJ were used as the development set, and sections 13–24 as the test set.

4 Application to Relative Clause Attachment

Sentence 1 is a typical example of relative clause (RC) attachment ambiguity.

Both attachments are grammatical, but intuitively *opening* is more likely to occur with the verbs *lead* or *lead to* than *house*. Our hypothesis is that this type of pragmatic knowledge (openings lead to something, houses don't) will be reflected in dependencies extracted from a large corpus. Extracting dependencies is particularly important as RC attachment is a more difficult problem than PP attachment as the following examples show.

(2) Texaco Inc. reported [an 11% increase] in [third-quarter earnings], which it *attributed* partly to the company's massive restructuring [...]

(3) Earlier this year DPC Acquisition made [a $15-a-share offer] for [Dataproducts], which the Dataproducts board said it *rejected* [...]

(4) [...] said Edmar Mednis, [the expert commentator] for [the match], which was *attended* by hundreds of chess fans.

RC attachment interacts with a wider range of grammatical phenomena than PP attachment (e.g., object vs. subject relatives, passive, and agreement). Also, many cases of PP attachment can be resolved structurally. For example, an

[1] Minipar attaches relative clauses low by default, resulting in many incorrect attachment decisions. Since relative clauses are rare, we do not systematically eliminate them when computing "unlabeled" statistics.

on-PP after *rely* almost always attaches to the verb. In contrast, RC attachment is mostly semantic (e.g., *opening* is a more typical subject of *lead to* than *house*). For our experiments, we extracted all sentences from the WSJ corpus that contained a pattern of the form `NP1 Prep NP2 which/that`. (See (Web Appendix, 2006) for documentation on the patterns used.) Our development set contained 282 *which*-clauses (71 with high attachment; 211 with low attachment) and 385 *that*-clauses (156 with high attachment and 229 with low attachment). The test corpus contained 264 *which*-clauses (71 with high attachment and 193 with low attachment) and 391 *that*-clauses (175 with high attachment and 216 with low attachment). For the case of relative clause attachment, we simplify the representation of triples $< \mathrm{R}, i_1, \mathrm{XP} >$,$< \mathrm{R}, i_2, \mathrm{XP} >$ to pairs $< NP_1, \mathrm{XP} >$, $< NP_2, \mathrm{XP} >$,where NP_1 and NP_2 are two potential attachment sites the relative clause can attach to, and XP consists of verb and object (if there is an object) of the relative clause. The maximum lattice for relative clause attachment is depicted in Figure 2. The lattice will be smaller if there is no object, premodifying adjective etc. The supremum of the lattice corresponds to a query that includes the entire NP (including modifying adjectives/nouns)[2], the verb and its object: *"weekly mod report"* && *"report subj show"* && *"decline obj show"*. The generalizing options are:

- strip the NP of the modifying adjective/noun (*weekly report* → *report*)
- use only the head noun of the NP (*Catastrophic Care Act* → *Act*)
- use the head noun in lower case (*Act* → *act*)
- for named entities use a hypernym of the NP (*American Bell Telephone Co.* → *company*)
- strip the object from XP (*company have subsidiary* → *company have*)
- don't use any context at all. In this case the default attachment (to the last NP) is selected.

To compute the values of ϕ, we first parse the sentence with Minipar and extract the relevant verb and grammatical relation. Then we query the database for *subject*, *object*, and *modifier* relations to calculate $P(\mathrm{NP})$, $P(\mathrm{XP})$, and $P(< \mathrm{NP}, \mathrm{XP} >)$. For example, $P(< \mathit{opening}, \mathit{lead\_\ to} >)$ is estimated based on the query *“opening subj lead_ to”*. Including further information about the context (e.g. about the object of the verb in the relative clause) – as opposed to only using noun-verb co-occurrence – proved particularly useful for light verbs like *make* and *have*.

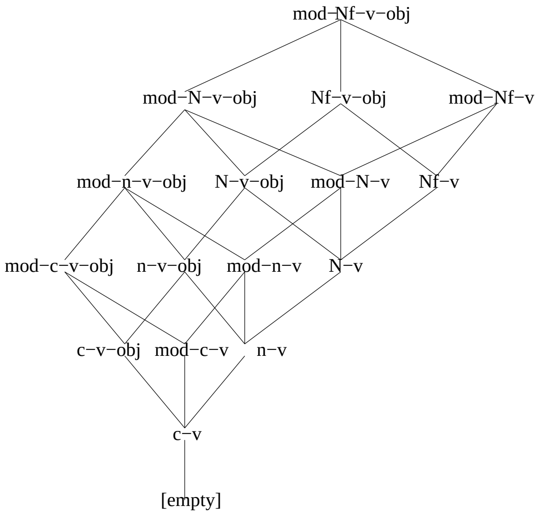

Figure 2: Partially ordered set of pairs of potential attachment site NP and relative clause XP, where mod: premodifying adjective or noun, Nf: head noun with lexical modifiers, N: head noun only, n: head noun in lower case, c: class of NP, v: verb in relative clause, obj: object of verb in the relative clause.

4.1 Named Entities

Named entities often cause sparse data problems. For this reason, we also use queries in lower case and queries where the named entity is replaced by its class. For Example 5 we would have queries *Act subj boost* and *act subj boost.*

(5) Congress still is struggling to dismantle [the unpopular Catastrophic Care Act] of [1988], which boosted benefits for the elderly and taxed them to pay for the new coverage.

[2] From the Minipar output, we use all adjectives which modify the NP via the relation *mod*, and all nouns, which modify the NP via the relation *nn*.

To identify the class of a named entity we use LingPipe (LingPipe, 2006). When LingPipe identifies a named entity as a company or organization, we replace it with *company* in the query. Locations are replaced by *country*. Persons block RC attachment because neither *which* nor *that* clauses attach to person names, resulting in an attachment of the RC to the other NP.

4.2 A Worked Example

Table 1 shows mutual information values for the queries constructed for sentence 6.

(6) The firmness in heating oil was attributed to colder weather in parts of the U.S. and to the latest [weekly report] by [the American Petroleum Institute], which *showed* a decline in inventories of the fuel.

queries for <weekly report, show decline> etc.	MI
"weekly mod report" && "report subj show" && "decline obj show"	0
"weekly mod report" && "report subj show"	8.63
"report subj show" && "decline obj show"	5.38
"report subj show"	7.21
queries for <API, show decline> etc.	MI
"American_Petroleum_Institute subj show" && "decline obj show"	8.44
"Institute subj show" && "decline obj show"	0
"institute subj show" && "decline obj show"	0
"company subj show" && "decline obj show"	1.39
"American_Petroleum_Institute subj show"	8.47
"Institute subj show"	0
"institute subj show"	4.50
"company subj show"	3.17
[empty]	6

Table 1: Queries for computing $P(< NP_1, \mathrm{XP} >)$ (high attachment, above) and $P(< NP_2, \mathrm{XP} >)$ (low attachment, below) for Example 6, (including further tuples after applying g_j) and corresponding mutual information values (MI).

In Table 1, the highest value for the high attachment site *weekly report* is 8.63 and the highest value for the low attachment site is 8.47. We hence choose high attachment for this case. Note that the low attachment site has a value 6 for the empty context. This value reflects the bias for low attachment: the majority of relative clauses are attached low. If all MI-values are zero or otherwise low, this procedure will automatically result in low attachment.

4.3 Decision list

For increased accuracy, the structural disambiguation method is embedded in the following decision list. Step 4 is the lattice-based algorithm described above.

1. If Minipar has already chosen high attachment, choose high attachment (only relevant for named entities in some of the *which* clauses in our data).
2. If there is agreement between the verb and only one of the NPs, attach to this NP.
3. If one of the NPs is in a list of person entities, attach to the other NP.[3]
4. If possible, use structural disambiguation based on the affinities computed on the Reuters corpus.
5. If none of the above strategies was successful (e.g. in the case of parsing errors, where the verb or the relation cannot be retrieved), attach low.

5 Evaluation

that clauses	accuracy
development set, baseline	59.48%
development set, algorithm	64.42%
test set, baseline	55.24%
test set, algorithm	60.87%
which clauses	accuracy
development set, baseline	74.82%
development set, Minipar	78.37%
development set, algorithm	82.27%
test set, baseline	73.12%
test set, Minipar	75.75%
test set, algorithm	78.41%

Table 2: Evaluation results (percentage of correct attachments) for *that* and *which* clauses.

We first evaluated the accuracy of relative clause attachment with Minipar as the base parser. Table 2 shows the evaluation results

[3] This list contains 136 entries and was semiautomatically computed from the Reuters corpus: Antecedents of *who* relative clauses were extracted, and the top 200 were filtered manually.

when the algorithm is run against our development and test sets. We set $\phi(<\emptyset, i, \emptyset>) = 6$.[4] The baseline is always attaching low. Minipar always attaches low except for named entities of the form `NP Prep NP` (e.g. *The State Commission on Judicial Conduct*), which are recognized as a unit, resulting in high attachment for some *which* relative clauses. For *that* clauses, Minipar always attaches low.[5]

For *that* clauses we achieved results about 5 percentage points above the baseline; for *which* clauses about 5 to 7 points above the baseline, and about 3 points above Minipar.

set	not used	accuracy
development		64.42%
development	mod	64.16%
development	mod,f	63.90%
development	mod,f,obj	63.64%
development	mod,f,obj,c	63.38%
test		60.87%
test	mod	60.35%
test	mod,f	60.10%
test	mod,f,obj	60.10%
test	mod,f,obj,c	60.10%

Table 3: Accuracy on *that* clauses when the number of contextual features is decreased. The middle column shows what is left out (mod: the modifier is not used, f: only the head noun is used, obj: only the verb and not its object is used, c: the class/hypernym is not used.)

Tables 3 and 5 show how much of a decrease in accuracy is caused by using less context. For the development set the accuracy drops continuously as we omit an increasing number of elements of the context: pre-modifiers, lexical modifiers, objects, hypernyms. On the test set we can also observe a drop in accuracy. However, it is less consistent: Omitting the object does not decrease performance, and not using classes for named entities does have an effect on the *which* test set, but not on the *that* test set. These results show that using a larger context than just simple noun-verb co-occurrence improves performance and that a number of sources of information need to be combined for consistent improvement.

5.1 Integration into a statistical parser

After having shown the success of our method in a stand-alone evaluation, we now turn to evaluating it when integrated into a statistical parser, the Collins parser as reimplemented by (Bikel, 2004). We apply structural disambiguation (SD) to all *that*- and *which*-sentences of Sec. 13–24 with a relative clause attached to either the first or second NP in a pattern of the form "NP PREP NP RC". Sentences without an "NP PREP NP RC" structure in the gold standard are omitted (i.e., we don't attempt to correct spurious RC attachment ambiguity). Since we want to develop methods that can leverage small training sets, we perform the evaluation for 5 different training set sizes: 50%, 25%, 5%, 1%, and 0.1% of the Penn treebank, each a subset of Sec. 00–12 (Table 4). Note that the number of eligible relative clause constellations in the test set varies depending on the training set.

For *which* sentences, SD consistently improves parsing accuracy. For *that* sentences accuracy is improved for small training sets (0.1% and 1%). Differences that are significant according to the χ^2-test are indicated in the table. This demonstrates that our approach is successful especially in cases where the amount of training data available is limited.

Train data	# **which** sent.	Coll. only	Coll.+SD
50%	251	71.7%	78.5%
25%	250	70.0%	78.8%*
5%	238	68.9%	79.8%*
1%	245	67.8%	78.9%*
0.1%	194	60.8%	75.8%*
Train data	# **that** sent.	Coll. only	Coll.+SD
50%	366	72.7%	62.3%
25%	367	70.3%	61.9%
5%	356	67.4%	61.2%
1%	354	58.8%	60.2%
0.1%	314	47.5%	61.2%*

Table 4: Performance of the Collins parser (percent correct attachments) with and without structural disambiguation (SD). The combined method is superior for *which* and for small training sets. Significant improvements are marked with $*$.

[4] We experimented with a number of values on our development set. Accuracy of the algorithm is only slightly affected for values between 4 and 7.

[5] Note that this property leads to a higher "Minipar" baseline for *which* clauses.

6 Related Work

There have been few attempts to incorporate information from unlabeled corpora directly into the parser (Charniak, 1997; Johnson and Riezler, 2000), but they were either unsuccessful or tested on small data sets only. We know of no other work that combines attachment disambiguation based on unlabeled corpora with state-of-the-art statistical parsers.

Our lattice formalization can be viewed as a back-off model that combines estimates from several "backoffs" (in a typical back-off model, there is a single more general model to back off to). (Collins and Brooks, 1995) present a similar approach for prepositional phrases. One variant of their model computes the estimate in question as the average of three "backoffs." In contrast to prepositional phrases, many other attachment decisions, including relative clause attachments, are largely semantic. Given the verb *rely*, verb attachment of a PP headed by *on* is very likely. There are no similar strong regularities for semantic attachments: they require measuring the semantic "fit" of the two elements being syntactically attached to each other. This is why we use MI in this paper to disambiguate attachment. To our knowledge, MI has not been used in a back-off model before.

The lattice can also be viewed as a set of overlapping features, similar to the feature space of many discriminative algorithms. However, in contrast to discriminative learning, our approach is unsupervised.

There is a large body of literature on PP attachment, e.g. (Hindle and Rooth, 1991; Volk, 2001; Calvo et al., 2005) that shares the overall goals of this paper: using information from unlabeled corpora for syntactic disambiguation. (Volk, 2001) counts the number of occurrences of word n-grams on the web to select the correct attachment of PPs. We believe that grammatical dependencies are a more promising research direction since they are more robust compared to raw text if data are sparse. (Toutanova et al., 2004)'s approach is similar to ours in that morphological variants and word classes are considered, but their method differs in that they use both labelled corpora and unlabelled corpora for calculating attachment decisions. Work in the tradition of (Hindle and Rooth, 1991) is most similar to the approach proposed here. The authors parse an unannotated corpus and use dependency statistics for disambiguation of PP attachment. Our interest is in developing a framework that can disambiguate syntactic ambiguities in general, at least as far as they correspond to attachment ambiguities, as opposed to solving a particular syntactic ambiguity problem.

set	not used	accuracy
development		82.27%
development	mod	81.21%
development	mod,f	81.21%
development	mod,f,obj	80.49%
development	mod,f,obj,c	78.72%
test		78.41%
test	mod	78.41%
test	mod,f	78.03%
test	mod,f,obj	78.41%
test	mod,f,obj,c	78.03%

Table 5: Accuracy on *which* clauses, when the number of contextual features is decreased. (cf. Table 3 for further explanation.)

Previous work on relative clause attachment has taken a machine learning approach where an attachment decision is represented as a feature vector which is then fed into a classifier trained on a labeled training set. In contrast, our main emphasis is on exploiting information from unlabeled corpora. (Siddharthan, 2002a; Siddharthan, 2002b) uses WordNet classes for constructing some of the features characterizing attachments. For *which* clauses (Siddharthan, 2002b) achieves an accuracy of 76.5% on his test set.[6] RC attachment is also addressed by (Yeh and Vilain, 1998), who experiment with a transformation-based error-driven learning approach, which aims to disambiguate various cases of PP attachment ambiguities and subordinate clauses at the same time. They report an overall accuracy of 75.4%, but do not give numbers for relative clause attachment.

[6] We attempted to recreate Siddharthan's training and test sets, but were not able to based on the description in the paper and email communication with the author.

7 Conclusion

We make three contributions in this paper. First, we propose a lattice-based framework for combining supervised and unsupervised methods for syntactic disambiguation. Parses from a treebank-trained parser are refined by using additional information from a large unannotated corpus, represented as dependencies extracted by a dependency parser. The lattice integrates information obtained from variable context sizes. This approach makes it possible to base attachment decisions on the most specific context available in the unlabeled corpus.

Secondly, we evaluate attachment disambiguation by comparing to the performance of a state-of-the-art parser. Most previous work on attachment ambiguity has not been evaluated against this stringent baseline. We also argue that it is important to compare results across different training set sizes since in practical applications we can expect training sets to be smaller than is typical in academia.

Finally, we address the problem of relative clause attachment, a problem that has received much less attention than PP attachment. We argue that RC attachment is a good test case for enhancing statistical parsers with information from unlabeled corpora because it is more complex than PP attachment due to a wider range of grammatical phenomena involved and because few instances of RC attachment ambiguity can be resolved structurally. We also provide a baseline for future evaluations of work on RC attachment disambiguation.

References

Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Hiram Calvo, Alexander Gelbukh, and Adam Kilgarriff. 2005. Distributional thesaurus vs. wordnet: A comparison of backoff techniques for unsupervised PP attachment. In *CICLing 6*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *ACL 43*.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *AAAI/IAAI*, pages 598–603.

Michael Collins and James Brooks. 1995. Prepositional attachment through a backed-off model. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, Somerset, New Jersey. Association for Computational Linguistics.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70, March.

Donald Hindle and Mats Rooth. 1991. Structural ambiguity and lexical relations. In *Proc. of ACL 29*, pages 229–236, Morristown NJ. Association of Computational Linguistics.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *NAACL*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

LingPipe. 2006. http://www.alias-i.com/lingpipe/.

Lucene. 2006. http://lucene.apache.org.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large natural language corpus of English: the penn treebank. *Computational Linguistics*, 19:313–330.

Advaith Siddharthan. 2002a. Resolving attachment and clause boundary ambiguities for simplifying relative clause constructs. In *Student Research Workshop, ACL*.

Advaith Siddharthan. 2002b. Resolving relative clause attachment ambiguities using machine learning techniques and wordnet hierarchies. In *Proceedings of the 4th Discourse Anaphora and Anaphora Resolution Colloquium (DAARC 2002)*.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *Proceedings of ICML*.

Martin Volk. 2001. Exploiting the WWW as a corpus to resolve pp attachment ambiguities. In *Proceedings of Corpus Linguistics 2001*.

Web Appendix. 2006. http://www.ims.uni-stuttgart.de/˜schuetze/conll06/apdx.html.

Alexander S. Yeh and Marc B. Vilain. 1998. Some properties of preposition and subordinate conjunction attachments. In *Coling 17*.