Parse Disambiguation for a Rich HPSG Grammar

Kristina Toutanova,* Christopher D. Manning,* Stuart M. Shieber,‡ Dan Flickinger,† and Stephan Oepen†

{kristina|manning}@cs.stanford.edu shieber@deas.harvard.edu
{danf|oe}@csli.stanford.edu

*Dept of Computer Science, Stanford University, Stanford, CA 94305-9040, USA

†CSLI, Stanford University, Stanford, CA 94305, USA

‡Dept of EECS, Harvard University, Cambridge, MA 02138, USA

1 Introduction

In this paper, we describe experiments on HPSG parse disambiguation using the Redwoods HPSG treebank (Oepen et al. 2002a,b,c). HPSG is a constraint-based lexicalist ("unification") grammar formalism.[1]

The fine-grained nature of the HPSG representations found in the Redwoods treebank raises novel issues in parse disambiguation relative to more traditional treebanks such as the Penn treebank, which have been the focus of most past work on probabilistic parsing (e.g., Charniak 1997; Collins 1997). The Redwoods treebank is much richer in the representations it makes available. Most similar to Penn treebank parse trees are the phrase structure trees (Figure 1(b)). In this work we have concentrated on the derivation trees (Figure 1(a)), which represent combining rule schemas of the HPSG grammar. The nodes represent, for example, head-complement, head-specifier, and head-adjunct schemas and the derivation trees are consequently significantly different from phrase structure trees. The preterminals of the derivation trees are from a set of about 8,000 lexical labels and are much finer grained than Penn treebank labels, which are about 45 part-of-speech tags, and 27 phrasal node labels.

Another important difference between the (implicit) Penn treebank grammar and the LinGO ERG (English Resource Grammar) is that the latter is maximally binary, with extensive use of unary schemas for implementing morphology and type-changing operations. Much common wisdom that has been acquired for building probabilistic models over Penn treebank parse trees is implicitly conditioned on the fact that the flat representations of the Penn treebank trees mean that most important dependencies are represented jointly in a local tree. Thus lessons learned may not be applicable to our problem (see Collins (1999) for a careful discussion of this issue).

Finally, the Redwoods treebank provides deep semantic representations for sentences, building up an underspecified minimal recursion semantics (MRS) representation (Copestake et al. 1999) together with the syntactic analyses for sentences. This semantic information, unavailable in the Penn treebank, may provide a useful source of additional features, at least partially orthogonal to syntactic information, for aiding parse disambiguation.

On the one hand, the richer information has the potential to provide greater information to ease parse disambiguation; on the other hand, the finer grain raises increased data sparsity issues, especially since the corpus available to us is far smaller than the Penn treebank. It is thus unclear a priori how the unique aspects of the HPSG representations will affect performance on the parse disambiguation task.

[1]For an introduction to HPSG, see the text by Pollard and Sag (1994).

We have explored building probabilistic models for parse disambiguation using this rich HPSG treebank, assessing the effectiveness of different kinds of information. We describe generative and discriminative models using analogous features and compare their performance on the disambiguation task. Among the results that we obtain are:

- Lexical information alone accounts for only half of the parse ambiguity inherent in the corpus, providing an upper bound on parse disambiguation via tagging, which we approach within a few percent.
- Using multiple sources of information, in particular, semantic information, can synergistically improve parse disambiguation performance.
- Conditional models achieve about a 15% error reduction over generative models.
- The models achieve quite high overall parse disambiguation performance, as much as 82.5% exact match parse selection accuracy on ambiguous sentences in the corpus.
- Of the remaining errors, we believe that about 50% are subject to elimination through improved models (as opposed to resulting from errors in the corpus or underlying grammar).

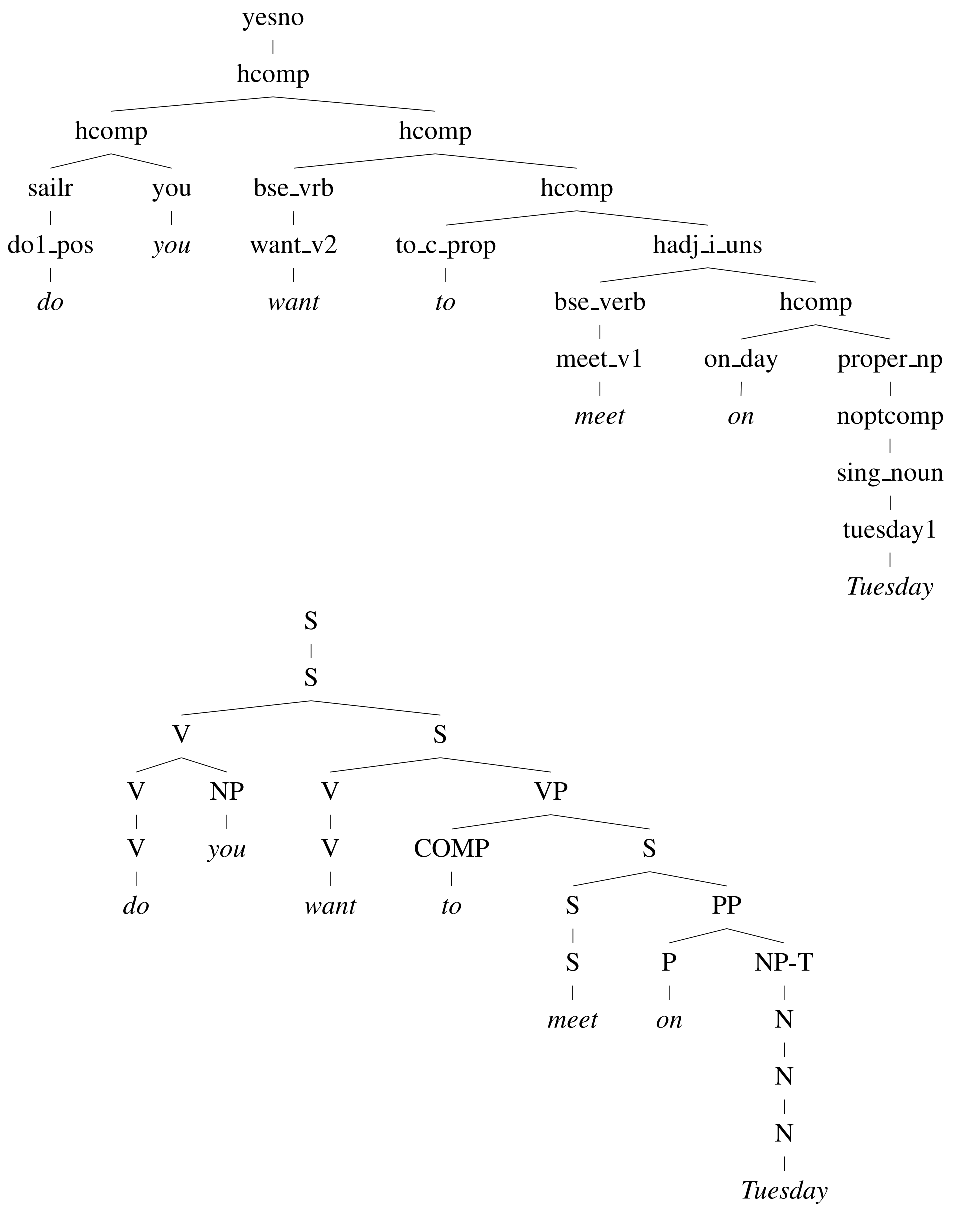
In the sections that follow, we describe the various statistical models we test, provide experimental results on the parse disambiguation task, and provide some preliminary error analysis.

2 Overview of Models

A variety of approaches are possible for building statistical models of parse disambiguation. The Redwoods treebank makes available exhaustive HPSG sign representations for all analyses of sentences. These are large attribute-value matrices which record all aspects of a sentence's syntax and semantics. In our initial experiments we have concentrated on using small subsets of these representations. We have explored training stochastic models using derivation trees and semantic trees (which are approximations to the MRS representation). Figure 1 shows examples of a derivation tree, phrase structure tree and an elementary dependency graph. The learned probabilistic models were used to rank possible parses of unseen test sentences according to the probabilities they assign to them.

In our initial experiments we built a tagger for the HPSG lexical tags in the treebank, and report results on using the tagger for parse disambiguation. The tags are the lexical labels in the derivation trees. In contrast to the Penn treebank's tagset of some 40 part-of-speech tags, the Redwood corpus uses a much finer-grained set of over 8,000 lexical tags.

Subsequent models included modeling of tree structures. Most probabilistic parsing research is based on branching process models (Harris 1963). The HPSG derivations that the treebank makes available can be viewed as such a branching process, and a stochastic model of the trees can be built as, for instance, a probabilistic context-free grammar (PCFG) model. Abney (1997) notes problems with the soundness of the approach, showing that the distribution of derivations of a unification-based grammar may well not be in the class of PCFG grammars defined using its context-free base. He motivates the use of log-linear models (Agresti 1990) for parse ranking that Johnson and colleagues further developed (Johnson et al. 1999). Building conditional log-linear models is also expected to improve generalization performance because the criterion being optimized is discriminative (Klein and Manning 2002; Ng and Jordan 2002; Vapnik 1998).




```
_4:{
   _4:int_rel[SOA e2:_want2_rel]
  e2:_want2_rel[ARG1 x4:pron_rel, ARG4 _2:hypo_rel]
  _1:def_rel[BV x4:pron_rel]
  _2:hypo_rel[SOA e18:_meet_v_rel]
  e18:_meet_v_rel[ARG1 x4:pron_rel]
  e19:_on_temp_rel[ARG e18:_meet_v_rel, ARG3 x21:dofw_rel]
  x21:dofw_rel[NAMED :tue]
  _3:def_np_rel[BV x21:dofw_rel]
 }
```

Figure 1: Native and derived Redwoods representations for the sentence *Do you want to meet on Tuesday?* — (a) derivation tree using unique rule and lexical item identifiers of the source grammar (top), (b) phrase structure tree labelled with user-defined, parameterizable category abbreviations (center), and (c) elementary dependency graph extracted from the MRS meaning representation (bottom).

In this work we have experimented with both generative and conditional log-linear models over the same feature sets and we report results achieved using both kinds of models. We examine the performance of five models: an HMM tagging model, a simple PCFG, a PCFG with ancestor annotation where the number of ancestors was selected automatically, a model of semantic dependencies, and a hybrid model that combines predictions from several of the above models. For these models we also trained corresponding conditional log-linear models using the same information sources as the generative models.

These models will be described in more detail in the next section. We first describe the generative models and after that their corresponding conditional log-linear models.

3 Generative Models

The tagger we implemented is a standard trigram HMM tagger, defining a joint probability distribution over the preterminal sequences and yields of the derivation trees. Trigram probabilities are smoothed by linear interpolation with lower-order models. Our tagging model does not take advantage of the lexical types (which are about 500 syntactic types) or the type hierarchy in which they are organized and we plan to pursue incorporating this information in future models. The lexical types are not shown in the Figure 1. They are the direct super-types of the lexical labels. For example, the lexical type of the word *meet* in the figure is *v_unerg_le*, and the lexical type of *want* is *v_subj_equi_le*.

The PCFG models define probability distributions over the trees of derivational types corresponding to the HPSG analyses of sentences. A PCFG model has parameters $\theta_{i,j}$ for each rule $A_i \rightarrow \alpha_j$ in the corresponding context free grammar.[2] In our application, the nonterminals in the PCFG A_i are schemas of the HPSG grammar used to build the parses (such as HEAD-COMPL or HEAD-ADJ). We set the parameters to maximize the likelihood of the set of derivation trees for the preferred parses of the sentences in a training set. In further discussion we will refer to this simple PCFG model as PCFG-S.

PCFG models can be made better if the rule applications are conditioned to capture sufficient context. For example, grandparent annotation for PCFGs has been shown to significantly improve parsing accuracy (Charniak and Caroll 1994; Johnson 1998). One feature of the LinGO ERG is that it is binarized and thus it becomes increasingly important to make probabilistic models aware of a wider context. We implemented an extended PCFG that conditions each node's expansion on several of its ancestors in the derivation tree. The number of ancestors to condition on was selected automatically according to a minimum description length (MDL) criterion (Rissanen 1989). We set an upper bound of four ancestors. Our method of ancestor selection is similar to learning context-specific independencies in Bayesian networks (Friedman and Goldszmidt 1996). We will refer to the PCFG model with ancestor information as PCFG-A.

We also learned PCFG-style models over trees of semantic dependencies extracted from the HPSG signs. These semantic models served as an early experiment in using semantic information for disambiguation. We intend as work progresses to build stochastic models over the elementary dependency graphs extracted from MRS meaning representations shown in Figure 1. The semantic trees mirror the derivation trees. They were obtained in the following manner: Each node in the derivation tree was annotated with its key semantic relation (Copestake et al. 1999). Consequently the annotated tree was flattened so that all dependents of a semantic relation occur at the same level of the tree as its direct descendants. Since the resulting local trees correspond to a huge number of rules, we implemented a more parsimonious model that estimates the probability of each rule by making strong independence

[2] For an introduction to PCFG grammars see, for example, the text by Manning and Schütze (1999).

assumptions and relaxing them by considering more context when possible, using the same algorithm for feature selection based on MDL that we used for PCFG-A. In further discussion we will refer to the model of semantic dependencies as PCFG-Sem.

We explored combining the predictions from the PCFG-A model, the tagger, and PCFG-Sem. The combined model computes the scores of analyses as linear combinations of the log-probabilities assigned to the analyses by the individual models. Since some of the factors participating in the tagger also participate in the PCFG-A model, in the combined model we used only the trigram tag sequence probabilities from the tagger. These are the transition probabilities of the HMM tagging model.

More specifically, for a tree t,

$$Score(t) = \log(P_{PCFG\text{-}A}(t)) + \lambda_1 \log(P_{TRIG}(tags(t))) + \lambda_2 \log(P_{PCFG\text{-}Sem}(t))$$

where $P_{TRIG}(tags(t))$ is the probability of the sequence of preterminals $t_1 \cdots t_n$ in t according to a trigram tag model:

$$P_{TRIG}(t_1 \cdots t_n) = \prod_{i=1}^{n} P(t_i | t_{i-1}, t_{i-2})$$

with appropriate treatment of boundaries. The trigram probabilities are smoothed as for the HMM tagger. The combination weights λ_1 and λ_2 were not fitted extensively. The performance of the model was stable under changes of the value of λ_1 in the range 0.2 to 1, whereas the performance of the combination went down if λ_2 was set to a value above 0.5 . We report results using values $\lambda_1 = 0.8$ and $\lambda_2 = 0.3$.

4 Conditional Log-linear Models

A conditional log-linear model for estimating the probability of an HPSG analysis given a sentence has a set of features $\{f_1, \ldots, f_m\}$ defined over analyses and a set of corresponding weights $\{\lambda_1, \ldots, \lambda_m\}$ for them. In this work we have defined features over derivation trees and semantic trees as described for the branching process models.

For a sentence s with possible analyses $t_1, \ldots, t_k$, the conditional probability for analysis t_i is given by:

$$P(t_i | s) = \frac{\exp \sum_{j=1,\ldots,m} f_j(t_i)\lambda_j}{\sum_{i'=1,\ldots,k} \exp \sum_{j=1,\ldots,m} f_j(t_{i'})\lambda_j} \quad (1)$$

As described by Johnson et al. (1999), we trained the model by maximizing the conditional likelihood of the preferred analyses and using a Gaussian prior for smoothing (Chen and Rosenfeld 1999). We used the conjugate gradient method for optimization.

For the five generative models described in the previous section, we built conditional log-linear models using the same corresponding features. We refer to the log-linear models as CTrigram, CPCFG-S, CPCFG-A, CPCFG-Sem, and CCombined. These models correspond to the generative models Trigram, PCFG-S, PCFG-A, PCFG-Sem, and Combined respectively. For example, the conditional log-linear model CPCFG-S has one feature for each expansion of each nonterminal in the derivation trees $A_i \rightarrow \alpha_j$. The features of the other models were defined analogously to correspond to the respective generative models.

5 Experimental Results

We report parse disambiguation results on the dataset described in Table 1. This corpus is the subset of the current annotated Redwoods corpus for which exactly one analysis was chosen as correct by the annotator. For the other sentences in the treebank, either none of the analyses was chosen as correct, or more than one analysis was acceptable. At this stage of treebank development, the annotation is expected to have some errors, since it was done by a single annotator. In the section on error analysis we discuss the estimated fraction of erroneous annotations further.

Table 2 shows the accuracy of parse selection using the generative models described in section 3. Note that we restrict attention in the test corpus to sentences that are ambiguous according to the grammar, that is, for which the parse selection task is nontrivial. The accuracy results are averaged over a ten-fold cross-validation on the complete data set summarized in Table 1.

Accuracy results denote the percentage of test sentences for which the highest ranked analysis was the correct one. Often the models give the same score to several different parses. In these cases, when a model ranks a set of m parses highest with equal scores and one of those parses is the preferred parse in the treebank, we compute the top one accuracy on this sentence as $\frac{1}{m}$. For comparison, a baseline showing the expected performance of choosing parses randomly according to a uniform distribution is included as the first row.

The results in Table 2 indicate that high disambiguation accuracy can be achieved using simple statistical models. The HMM tagger does not perform well on the task by itself in comparison with other models that have more information about the parse. For comparison, we present the performance of a hypothetical clairvoyant tagger that knows the true tag sequence and scores highest the parses that have the correct preterminal sequence. The performance of the perfect tagger shows that, informally speaking, roughly half of the information necessary to disambiguate parses is available in the lexical tags. Using ancestor information in the PCFG models improved parse ranking accuracy significantly over a simple model PCFG-S. The PCFG-Sem model has respectable accuracy but does not by itself work as well as PCFG-A. The performance of model combination shows that the information they explore is complimentary. The tagger adds left-context information to the PCFG-A model (in a crude way) and the PCFG-Sem model provides semantic information.

Table 3 shows the accuracy of parse selection using the conditional log-linear models. We see that higher accuracy is achieved by the discriminative models. The difference between the generative and conditional log-linear models is largest for the PCFG-S model and its corresponding CPCFG-S model. The difference between the generative and conditional log-linear models for the trigram tagger is small and this result is in agreement with similar results in the literature comparing HHM and conditional random fields models for part of speech tagging (Klein and Manning 2002). Overall the gain from using conditional log-linear models for the final combined model is 14% error reduction from the generative model.

The parse disambiguation accuracy achieved by these models is quite high. However, in evaluating this level of performance we need to take into account the lower ambiguity rate of our corpus and the smaller sentence length. To assess the influence of ambiguity rate on the parse disambiguation accuracy of our model, we computed average accuracy of the best model CCombined as a function of the number of possible analyses per sentence. Table 4 shows the breakdown of accuracy for several sentence categories.

The first row shows the number of sentences with ambiguity greater than or equal to two analyses, which are all sentences for which the disambiguation task is non-trivial. Therefore the random baseline and accuracy result are the same as in Table 3 for the CCombined model. Successive rows show

Table 1: Annotated corpus used in experiments: The columns are, from left to right, the total number of sentences, average length, and lexical and structural ambiguity

sentences	length	lex ambiguity	struct ambiguity
5312	7.0	4.1	8.3

Table 2: Performance of generative models for the parse selection task (exact match accuracy on ambiguous sentences).

Method		**Accuracy**
Random		25.81
Tagger	trigram	47.74
	perfect	54.59
PCFG	PCFG-S	66.26
	PCFG-A	76.67
	PCFG-Sem	69.05
	Combined	79.84

Table 3: Performance of conditional log-linear models for the parse selection task (accuracy).

Method		**Accuracy**
Random		25.81
CTagger	trigram	48.70
	perfect	54.59
CPCFG	CPCFG-S	79.30
	CPCFG-A	81.80
	CPCFG-Sem	74.30
	CCombined	82.65

Table 4: Parse ranking accuracy of CCombined by number of possible parses.

Analyses	Sentences	Random	CCombined
≥ 2	3824	25.81%	82.65%
≥ 5	1789	9.66%	71.32%
≥ 10	1027	5.33%	65.24%
≥ 20	525	3.03%	59.62%

random baseline and accuracy of our best model for the subset of sentences with ambiguity greater than or equal to the bound shown in the first column. We can see that the accuracy results degrade with increased ambiguity.

Based on our experiments, we can make the following observations:

- Overall it is surprising that the PCFG-S/A and CPCFG-S/A models over derivation trees work so well given the nature of node labels which are schema names and do not in general contain information about the phrasal types of the constituents.
- The current semantic models PCFG-Sem and CPCFG-Sem do not give us large performance gains. Perhaps this is due to data sparsity at the current size of the corpus, or the limitations of the semantic representation as semantic dependency trees rather than MRS structures.
- The conditional model CPCFG-Sem does not do much better than the joint PCFG-Sem model. This might be justified by the fact that although the CPCFG-Sem model will have a lower asymptotic error rate, it may not be approached due to the sparsity of the training data at the level of semantic relations (Ng and Jordan 2002).
- The conditional model CPCFG-A works so well that the combination with semantic and lexical label sequence features is much less advantageous than for the generative models. Also the overfitting effect of adding a large number of lexical features is stronger for the conditional model thus making it harder to improve generalization performance and making careful feature selection increasingly important.

6 Error Analysis

It is possible, by inspection of the errors made by the system, to see what the hard disambiguation decisions are that the combined syntactic-semantic models cannot at present get right. We analyzed some of the errors made by the best log-linear model defined over derivation trees and semantic dependency trees. We selected for analysis all of the 68 sentences that the model CCombined got wrong on one of the training-test splits in the 10-fold cross-validation on the whole corpus. The error analysis suggests the following breakdown:

- About 30% of errors are due to errors in annotation.
- About 10% of errors are due to grammar limitations. These are cases where the grammar did not provide a plausible analysis or it licensed analyses which should not have been possible choices.
- About 10% of the cases have more than one plausible analysis and discourse context is needed to resolve the ambiguity.
- About 50% of the errors are real errors and we could hope to get them right.

The inconsistency in annotation hurts the performance of the model both (i) when in the training data some sentences were annotated incorrectly and the model tried to fit its parameters to explain them and (ii) when in the test data the model chose the correct analysis but it was scored as incorrect because of incorrect annotation. (It is not straightforward to detect inconsistencies in the training data by inspecting test data errors. Therefore the percentages we have reported are not exact.)

The log-linear model seems to be more susceptible to errors in the training set annotation than the PCFG models, because it can easily adjust its parameters to fit the noise, especially when given a large number of features. This might partly explain why the log-linear model does not profit greatly from the addition of a large number of features.

On inspection of the real errors in the test set, which fall in the last error category listed above, we noted two most frequently occurring types of errors — PP attachment and lexical label selection.

The PP attachment errors were the single most common error type. These errors seem to be addressable by better use of semantic or lexical information as other researchers have proposed (e.g., Collins and Brooks 1995; Hindle and Rooth 1991). Most of the time low attachment is correct as has been observed for other treebanks and the model does seem to prefer low attachment fairly strongly. But we do not at present have special features to model low or high attachment and in future models we plan to add this information.

An example of an error of this sort where the correct attachment is high is for the sentence *"I do not like to go anywhere on Sundays"*, where the model chose to attach the PP *on Sundays* to *anywhere* rather than to *go*. For this case the low attachment to *anywhere* should be strongly dispreferred if there was sufficient lexical information.

Another interesting case of a PP attachment error is for the sentence *"I will put you in my schedule for March sixteenth at one o'clock"*. The correct attachment for the PP *at one o'clock* is low, as a modifier of *sixteenth*, but the model chose to attach it high to *put* in the meaning that the putting in the schedule event would happen at one o'clock and not the meeting. Again here semantic collocation information would be useful as for example knowing that people do not usually talk about entering information in their schedules at a particular time.

The second largest type of errors are cases where the lexical label of a word was not chosen correctly. An example of this is for the sentence *"Yeah, that is about all"*. The model selected the meaning of *about* as a preposition, whereas the preferred analysis of *about* in this case should be as a degree specifier. In addition to being very common as a degree specifier in our corpus domain, *about* is also very common in the collocation *about all*. So again lexical information should be useful. Another similar case is the sentence *"But we are getting real close to the holidays"*. The model did not select the correct meaning of *real* here as an adverb but chose the meaning of *real* as an adjective which could be a possible meaning in this sentence in fairy-tales but quite improbable in the domain of appointment scheduling.

Another amusing lexical error was for the sentence *"You said you were getting in Tuesday night"*. The model selected the rare meaning of *in* as an abbreviation for *Indiana*.[3] This is not semantically plausible in this sentence and domain as people should not normally get states.

In summary we think that more lexical information will help resolve attachment and lexical ambiguities. We can expect that increasing the corpus size will be helpful to obtain better word-specific statistics for our current models. Automatic clustering or exploring existing lexical hierarchies could also improve our modeling of semantic preferences. Since our current experiments suggest that there are not very big gains from the semantic dependencies model, further research is necessary to resolve this conflict of intuitions for what features should be helpful and what turns out to work in practice.

[3] Note that these sentences are a transcription of spoken dialogues so capitalization information is not reliably available in the data.

7 Conclusions and Future Work

This paper presented our work on HPSG parse disambiguation using statistical models. We demonstrated the usefulness of building models over derivation trees of HPSG analyses and how they can be supplemented with semantic and lexical label sequence information with significant accuracy improvement. Simultaneously we presented paired comparisons of generative and conditional models over the same features, showing the value of the conditional models.

In the near future we intend to further utilize the depth of syntactic and semantic information available in the Redwoods treebank to build more complex probabilistic models capable of using this information for prediction. In particular, as a first step, we plan to train stochastic models over the the MRS elementary dependency graphs. The non-tree structure of these graphs raises interesting research questions on how to combine information from multiple ancestors. Later we plan to explore using even closer approximations to the full MRS semantic representations. We also plan to use syntactic information at much greater depth than the derivation trees provide directly and to look into extracting useful features from HPSG signs.

8 Acknowledgments

We would like to thank Emily Bender, Ivan Sag, Thorsten Brants, Timothy Baldwin, and the other participants in the Redwoods project for fruitful discussion.

References

Abney, Steven P. 1997. Stochastic attribute-value grammars. *Computational Linguistics* 23:597 – 618.

Agresti, Alan. 1990. *Categorical Data Analysis*. John Wiley & Sons.

Charniak, Eugene. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 598 – 603, Providence, RI.

Charniak, Eugene, and G. Caroll. 1994. Context-sensitive statistics for improved grammatical language models. In *Proceedings of the Twelth National Conference on Artificial Intelligence*, pp. 742 – 747, Seattle, WA.

Chen, S., and R. Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon.

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

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

Collins, Michael John. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Meeting of the Association for Computational Linguistics and the 7th Conference of the European Chapter of the ACL*, pp. 16 – 23, Madrid, Spain.

Copestake, Ann, Daniel P. Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics. An introduction. in preparation.

Friedman, Nir, and Moises Goldszmidt. 1996. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*.

Harris, T. E. 1963. *The Theory of Branching Processes*. Berlin, Germany: Springer.

Hindle, Donald, and Mats Rooth. 1991. Structural ambiguity and lexical relations. In *Meeting of the Association for Computational Linguistics*, pp. 229–236.

Johnson, Mark. 1998. PCFG models of linguistic tree representations. *Computational Linguistics* 24: 613 – 632.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic 'unification-based' grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics*, pp. 535 – 541, College Park, MD.

Klein, Dan, and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *EMNLP 2002*.

Manning, Christopher D., and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Ng, Andrew, and Michael Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes. In *NIPS 14*.

Oepen, Stephan, Ezra Callahan, Dan Flickinger, and Christopher D. Manning. 2002a. LinGO Redwoods. A rich and dynamic treebank for HPSG. In *Beyond PARSEVAL. Workshop of the Third LREC Conference*, Las Palmas, Spain.

Oepen, Stephan, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. 2002b. LinGO Redwoods. A rich and dynamic treebank for HPSG. In *Treebanks and Linguistic Theories*, Sozopol, Bulgaria.

Oepen, Stephan, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002c. The LinGo Redwoods treebank: Motivation and preliminary applications. In *COLING 19*.

Pollard, Carl, and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Rissanen, Jorma. 1989. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *World Scientific Series in Computer Science*. New Jersey: World Scientific Publishing.

Vapnik, V. N. 1998. *Statistical Learning Theory*. John Wiley and Sons.