# A model of syntactic disambiguation based on lexicalized grammars

**Yusuke Miyao**
Department of Computer Science,
University of Tokyo
yusuke@is.s.u-tokyo.ac.jp

**Jun'ichi Tsujii**
Department of Computer Science,
University of Tokyo
CREST, JST
(Japan Science and Technology Corporation)
tsujii@is.s.u-tokyo.ac.jp

## Abstract

This paper presents a new approach to syntactic disambiguation based on lexicalized grammars. While existing disambiguation models decompose the probability of parsing results into that of primitive dependencies of two words, our model selects the most probable parsing result from a set of candidates allowed by a lexicalized grammar. Since parsing results given by the lexicalized grammar cannot be decomposed into independent sub-events, we apply a maximum entropy model for feature forests, which allows probabilistic modeling without the independence assumption. Our approach provides a general method of producing a consistent probabilistic model of parsing results given by lexicalized grammars.

## 1 Introduction

Recent studies on the automatic extraction of lexicalized grammars (Xia, 1999; Chen and Vijay-Shanker, 2000; Hockenmaier and Steedman, 2002a) allow the modeling of syntactic disambiguation based on linguistically motivated grammar theories including LTAG (Chiang, 2000) and CCG (Clark et al., 2002; Hockenmaier and Steedman, 2002b). However, existing models of disambiguation with lexicalized grammars are a mere extension of lexicalized probabilistic context-free grammars (LPCFG) (Collins, 1996; Collins, 1997; Charniak, 1997), which are based on the decomposition of parsing results into the syntactic/semantic dependencies of two words in a sentence under the assumption of independence of the dependencies. While LPCFG models have proved that the incorporation of lexical associations (i.e., dependencies of words) significantly improves the accuracy of parsing, this idea has been naively inherited in the recent studies on disambiguation models of lexicalized grammars.

However, the disambiguation models of lexicalized grammars should be totally different from that of LPCFG, because the grammars define the relation of syntax and semantics, and can restrict the possible structure of parsing results. Parsing results cannot simply be decomposed into primitive dependencies, because the complete structure is determined by solving the syntactic constraints of a complete sentence. For example, when we apply a unification-based grammar, LPCFG-like modeling results in an inconsistent probability model because the model assigns probabilities to parsing results not allowed by the grammar (Abney, 1997). We have only two ways of adhering to LPCFG models: preserve the consistency of probability models by abandoning improvements to the lexicalized grammars using complex constraints (Chiang, 2000), or ignore the inconsistency in probability models (Clark et al., 2002).

This paper provides a new model of syntactic disambiguation in which lexicalized grammars can restrict the possible structures of parsing results. Our modeling aims at providing grounds for i) producing a consistent probabilistic model of lexicalized grammars, as well as ii) evaluating the contributions of syntactic and semantic preferences to syntactic disambiguation. The model is composed of the syntax and semantics probabilities, which represent syntactic and semantic preferences respectively. The syntax probability is responsible for determining the syntactic categories chosen by words in a sentence, and the semantics probability selects the most plausible dependencies of words from candidates allowed by the syntactic categories yielded by the syntax probability. Since the sequence of syntactic categories restricts the possible structure of parsing results, the semantics probability is a conditional probability without decomposition into the primitive dependencies of words. Recently used machine learning methods including maximum entropy models (Berger et al., 1996) and support vector machines (Vapnik, 1995) provide grounds for this type of model-

ing, because it allows various dependent features to be incorporated into the model without the independence assumption.

The above approach, however, has a serious deficiency: a lexicalized grammar assigns exponentially many parsing results because of local ambiguities in a sentence, which is problematic in estimating the parameters of a probability model. To cope with this, we adopted an algorithm of maximum entropy estimation for feature forests (Miyao and Tsujii, 2002; Geman and Johnson, 2002), which allows parameters to be efficiently estimated. The algorithm enables probabilistic modeling of complete structures, such as transition sequences in Markov models and parse trees, without dividing them into independent sub-events. The algorithm avoids exponential explosion by representing a probabilistic event by a packed representation of a feature space. If a complete structure is represented with a feature forest of a tractable size, the parameters can be efficiently estimated by dynamic programming.

A series of studies on parsing with wide-coverage LFG (Johnson et al., 1999; Riezler et al., 2000; Riezler et al., 2002) have had a similar motivation to ours. Their models have also been based on a discriminative model to select a parsing result from all candidates given by the grammar. A significant difference is that we apply maximum entropy estimation for feature forests to avoid the inherent problem with estimation: the exponential explosion of parsing results given by the grammar. They assumed that parsing results would be suppressed to a reasonable number through using heuristic rules, or by carefully implementing a fully restrictive and wide-coverage grammar, which requires a considerable amount of effort to develop. Our contention is that this problem can be solved in a more sophisticated way as is discussed in this paper. Another difference is that our model is separated into syntax and semantics probabilities, which will benefit computational/linguistic investigations into the relation between syntax and semantics, and allow separate improvements to both models.

Overall, the approach taken in this paper is different from existing models in the following respects.

- Since it does not require the assumption of independence, the probability model is consistent with lexicalized grammars with complex constraints including unification-based grammar formalism. Our model can assign consistent probabilities to parsing results of lexicalized grammars, while the traditional models assign probabilities to parsing results not allowed by the grammar.

- Since the syntax and semantics probabilities are separate, we can improve them individually. For example, the syntax model can be improved by smoothing using the syntactic classes of words, while the semantics model should be able to be improved by using semantic classes. In addition, the model can be a starting point that allows the theory of syntax and semantics to be evaluated through consulting an extensive corpus.

We evaluated the validity of our model through experiments on a disambiguation task of parsing the Penn Treebank (Marcus et al., 1994) with an automatically acquired LTAG grammar. To assess the contribution of the syntax and semantics probabilities to the accuracy of parsing and to evaluate the validity of applying maximum entropy estimation for feature forests, we compared three models trained with the same training set and the same set of features. Following the experimental results, we concluded that i) a parser with the syntax probability only achieved high accuracy with the lexicalized grammar, ii) the incorporation of preferences for lexical association through the semantics probability resulted in significant improvements, and iii) our model recorded an accuracy that was quite close to the traditional model, which indicated the validity of applying maximum entropy estimation for feature forests.

In what follows, we first describe the existing models for syntactic disambiguation, and discuss problems with them in Section 2. We then define the general form for parsing results of lexicalized grammars, and introduce our model in Section 3. We prove the validity of our approach through a series of experiments in Section 4.

## 2 Traditional models for syntactic disambiguation

This section reviews the existing models for syntactic disambiguation from the viewpoint of representing parsing results of lexicalized grammars. In particular, we discuss how the models incorporate syntactic/semantic preferences for syntactic disambiguation. The existing studies are based on the decomposition of parsing results into primitive lexical dependencies where syntactic/semantic preferences are combined. This traditional scheme of syntactic disambiguation can be problematic with lexicalized grammars. Throughout the discussion, we refer to the example sentence *"What does your student want to write?"*, whose parse tree is in Figure 1.

### 2.1 Lexicalized parse trees

The first successful work on syntactic disambiguation was based on lexicalized probabilistic context-free grammar (LPCFG) (Collins, 1997; Charniak, 1997). Although LPCFG is not exactly classified into lexicalized grammar formalism, we should mention these studies since they demonstrated that lexical dependencies were essential to improving the accuracy of parsing.

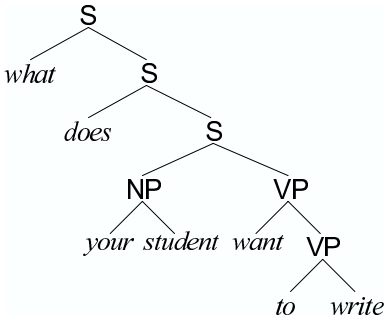


Figure 1: A parse tree for "*What does your student want to write?*"

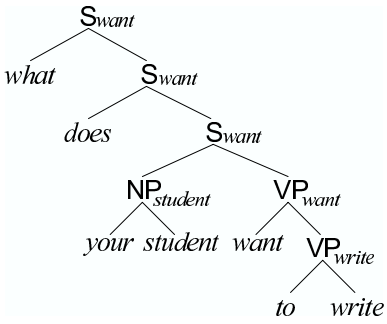


Figure 2: A lexicalized parse tree

A lexicalized parse tree is an extension of a parse tree that is achieved by augmenting each non-terminal with its lexical head. There is an example of a lexicalized parse tree in Figure 2, which is a lexicalized version of the one in Figure 1. A lexicalized parse tree is represented by a set of branchings in the tree[1]: $T = \{\langle w_{h_i}, w_{n_i}, r_i \rangle\}$, where $w_{h_i}$ is a head word, $w_{n_i}$ the head word of a non-head, and $r_i$ a grammar rule corresponding to each branching. LPCFG models yield a probability of the complete parse tree $T = \{\langle w_{h_i}, w_{n_i}, r_i \rangle\}$ by the product of probabilities of branchings in it.

$$p(T) = \prod_i p(w_{h_i}, w_{n_i}, r_i | \eta),$$

where $\eta$ is a condition of the probability, which is usually the nonterminal symbol of the mother node. Since each branching is augmented with the lexical heads of nonterminals in the rule, the model can capture lexical dependencies, which increase the accuracy. This is because lexical dependencies approximately represent the semantic preference of a sentence. As is well known, a syntactic structure is not accurately disambiguated only with syntactic preferences, and the incorporation of approximate semantic preferences was the key to improving the accuracy of syntactic disambiguation.

We should note that this model has the following three disadvantages.

1. The model fails to represent some linguistic dependencies, including long-distance dependencies and argument/modifier distinctions. Since an existing study incorporates these relations ad hoc (Collins, 1997), they are apparently crucial in accurate disambiguation. This is also problematic for providing a sufficient representation of semantics.

2. The model assumes the statistical independence of branchings, which is apparently not preserved. For example, the ambiguity of PP-attachments should be resolved by considering three words: the modifiee of the PP, its preposition, and the object of the PP.

3. The preferences of syntax and semantics are combined in the lexical dependencies of two words, i.e., features for syntactic preference and those for semantic preference are not distinguished in the model. Lexicalized grammars formalize the constraints of the relations between syntax and semantics, but the model does not assume the existence of such constraints. The model prevents further improvements to the syntax/semantics models; in addition to the linguistic analysis of the relation between syntax and semantics.

### 2.2 Derivation trees

Recent work on the automatic extraction of LTAG (Xia, 1999; Chen and Vijay-Shanker, 2000) and disambiguation models (Chiang, 2000) has been the first on the statistical model for syntactic disambiguation based on lexicalized grammars. However, the models are based on the lexical dependencies of elementary trees, which is a simple extension of the LPCFG. That is, the models are still based on decomposition into primitive lexical dependencies.

Derivation trees, the structural description in LTAG (Schabes et al., 1988), represent the association of lexical items i.e., elementary trees. In LTAG, all syntactic constraints of words are described in an elementary tree, and the dependencies of elementary trees, i.e., a derivation tree, describe the semantic relations of words more directly than lexicalized parse trees. For example, Figure 3 has a derivation tree corresponding to the parse tree in Figure 1[2]. The dotted lines represent substitution while the solid lines represent adjunction. We should note that the relations captured by ad-hoc augmentation

[1]For simplicity, we have assumed parse trees are only composed of binary branchings.

[2]The nodes in a derivation tree are denoted with the names of the elementary trees, while we have omitted details.

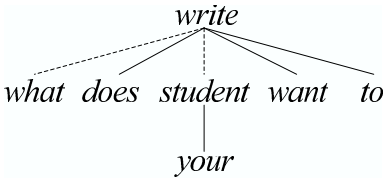


Figure 3: A derivation tree

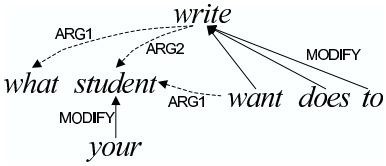


Figure 4: A dependency structure

of lexicalized parse trees, such as the distinction of arguments/modifiers and unbounded dependencies (Collins, 1997), are elegantly represented in derivation trees. Formally, a derivation tree is represented as a set of dependencies: $D = \{\langle \alpha_i, \eta_{\alpha_j}, r_i \rangle\}$, where $\alpha_i$ is an elementary tree, $\eta_{\alpha_i}$ represents a node in $\alpha_j$ where substitution/adjunction has occurred, and $r_i$ is a label of the applied rule, i.e., adjunction or substitution.

A probability of derivation tree $D = \{\langle \alpha_i, \eta_{\alpha_j}, r_i \rangle\}$ is generally defined as follows (Schabes et al., 1988; Chiang, 2000).

$$p(D) = \prod_i p(\alpha_i | \eta_{\alpha_j}, r_i)$$

Note that each probability on the right represents the syntactic/semantic preference of a dependency of two lexical items. We can readily see that the model is very similar to LPCFG models.

The first problem with LPCFG is partially solved by this model, since the dependencies not represented in LPCFG (e.g., long-distance dependencies and argument/modifier distinctions) are elegantly represented, while some relations (e.g., the control relation between "want" and "student") are not yet represented. However, the other two problems remain unsolved in this model. In particular, when we apply Feature-Based LTAG (FB-LTAG), the above probability is no longer consistent because of the non-local constraints caused by feature unification (Abney, 1997).

### 2.3 Dependency structures

A disambiguation model for wide-coverage CCG (Clark et al., 2002) aims at representing deep linguistic dependencies including long-distance dependencies and control relations. This model can represent all the syntactic/semantic dependencies of words in a sentence. However, the statistical model is still a mere extension of LPCFG, i.e., it is based on decomposition into primitive lexical dependencies.

In this model, a lexicalized grammar defines the mapping from a sentence into dependency structures, which represent all the necessary dependencies of words in a sentence, including long-distance dependencies and control relations. There is an example in Figure 4, which corresponds to the parse tree in Figure 1. Note that this representation includes a dependency not represented in the derivation tree (the control relation between "want" and "student"). A dependency structure is formally defined as a set of dependencies: $S = \{\langle w_{h_i}, w_{n_i}, \eta_i \rangle\}$, where $w_{h_i}$ and $w_{n_i}$ are a head and argument word of the dependency, and $\eta_i$ is an argument position of the head word filled by the argument word.

An existing model assigns a probability value to dependency structure $S = \{\langle w_{h_i}, w_{n_i}, \eta_i \rangle\}$ as follows.

$$p = \prod_i p(w_{n_i} | w_{h_i}, \eta_i)$$

Primitive probability is approximated by the relative frequency of lexical dependencies of two words in a training corpus.

Since dependency structures include all necessary dependency relations, the first problem with LPCFG is now completely solved. However, the third problem still remains unsolved. The probability of a complete parse tree is defined as the product of probabilities of primitive dependencies of two words. In addition, the second problem is getting worse; the independence assumption is apparently violated in this model, since the possible dependency structures are restricted by the grammar. The probability model is no longer consistent.

## 3 Probability Model based on Lexicalized Grammars

This section introduces our model of syntactic disambiguation, which is based on the decomposition of the parsing model into the syntax and semantics models. The concept behind it is that the plausibility of a parsing result is determined by i) the plausibility of syntax, and ii) selecting the most probable semantics from the structures allowed by the given syntax. This section formalizes the general form of statistical models for disambiguation of parsing including lexicalized parse trees, derivation trees, and dependency structures. Problems with the existing models are then discussed, and our model is introduced.

Suppose that a set $\mathcal{W}$ of words and a set $\mathcal{C}$ of syntactic categories (e.g., nonterminal symbols of CFG, elementary trees of LTAG, feature structures of HPSG (Sag and Wasow, 1999)) are given. A lexicalized grammar is

Lexicalized parse tree
⟨*write, what*, S→ *write* S⟩,
⟨*write, does*, S→ *does* S⟩,
⟨*write, student*, S→ NP VP⟩,
⟨*student, your*, NP→ *your student*⟩,
⟨*write, want*, VP→ *want* VP⟩,
⟨*write, to*, VP→ *to write*⟩

Derivation tree
⟨*write, what*, SUBST⟩,
⟨*write, does*, ADJ⟩,
⟨*write, student*, SUBST⟩,
⟨*student, your*, ADJ⟩,
⟨*write, want*, ADJ⟩,
⟨*write, to*, ADJ⟩

Dependency structure
⟨*write, what*, ARG2⟩,
⟨*write, does*, MODIFY⟩,
⟨*write, student*, ARG1⟩,
⟨*student, your*, MODIFY⟩,
⟨*write, want*, MODIFY⟩,
⟨*want, student*, ARG1⟩,
⟨*write, to*, MODIFY⟩

Figure 5: Parsing results of lexicalized grammars

then defined as a tuple $G = \langle L, R\rangle$, where $L = \{l = \langle w, c\rangle | w \in \mathcal{W}, c \in \mathcal{C}\}$ is a lexicon and $R$ is a set of grammar rules. A parsing result of lexicalized grammars is defined as a labeled graph structure $A = \{a | a = \langle l_h, l_n, d\rangle\}$, where $a$ is an edge representing the dependency of head $l_h$ and argument $l_n$ labeled with $d$. For example, the lexicalized parse tree in Figure 2 is represented in this form as in Figure 5, as well as the derivation tree and the dependency structure.

Given the above definition, the existing models discussed in Section 2 yield a probability $P(A|\mathbf{w})$ for given sentence $\mathbf{w}$ as in the following general form.

$$P(A|\mathbf{w}) = \prod_{a \in A} p(a|\eta),$$

In short, the probability of the complete structure is defined as the product of probabilities of lexical dependencies. For example, $p(a|\eta)$ corresponds to the probability of branchings in LPCFG models, that of substitution/adjunction in derivation tree models, and that of primitive dependencies in dependency structure models.

The models, however, have a crucial weakness with lexicalized grammar formalism; probability values are assigned to parsing results not allowed by the grammar, i.e., the model is no longer consistent. Hence, the disambiguation model of lexicalized grammars should not be decomposed into primitive lexical dependencies.

A possible solution to this problem is to directly estimate $p(A|\mathbf{w})$ by applying a maximum entropy model (Berger et al., 1996). However, such modeling will lead us to extensive tweaking of features that is theoretically unjustifiable, and will not contribute to the theoretical investigation of the relations of syntax and semantics. Since lexicalized grammars express all syntactic constraints by syntactic categories of words, we have assumed that we first determine which syntactic category $c$ should be chosen, and then determine which argument relations are likely to appear under the constraints imposed by the syntactic categories. Formally,

$$p(A|\mathbf{w}) = p(\mathbf{c}|\mathbf{w})p(A|\mathbf{c}).$$

The first probability in the above formula is *the probability of syntactic categories*, i.e., the probability of selecting a sequence of syntactic categories in a sentence. Since syntactic categories in lexicalized grammars determine the syntactic constraints of words, this expresses the syntactic preference of each word in a sentence. Note that our objective is not only to improve parsing accuracy but also to investigate the relation between syntax and semantics. We have not adopted the local contexts of words as in the supertaggers in LTAG (Joshi and Srinivas, 1994) because they partially include the semantic preferences of a sentence. The probability is purely unigram to select the probable syntactic category for each word. The probability is then given by the product of probabilities to select a syntactic category for each word from a set of candidate categories allowed by the lexicon.

$$p(\mathbf{c}|\mathbf{w}) = \prod_i p(c_i|w_i)$$

The second describes *the probability of semantics*, which expresses the semantic preferences of relating the words in a sentence. Note that the semantics probability is dependent on the syntactic categories determined by the syntax probability, because in lexicalized grammar formalism, a series of syntactic categories determines the possible structures of parsing results. Parsing results are obtained by solving the constraints given by the grammar. Hence, we cannot simply decompose semantics probability into the dependency probabilities of two words. We define semantics probability as a discriminative model that selects the most probable parsing result from a set of candidates given by parsing.

Since semantics probability cannot be decomposed into independent sub-events, we applied a maximum entropy model, which allowed probabilistic modeling without the independence assumption. Using this model, we can assign consistent probabilities to parsing results with complex structures, such as ones represented with feature structures (Abney, 1997; Johnson et al., 1999). Given parsing result $A$, semantics probability is defined as follows:

$$p(A|\mathbf{c}) = \frac{1}{Z_{\mathbf{c}}} \exp\left(\sum_{s \in \mathcal{S}(A)} \lambda(s)\right)$$

$$Z_{\mathbf{c}} = \sum_{A' \in \mathcal{A}(\mathbf{c})} \exp\left( \sum_{s' \in \mathcal{S}(A')} \lambda(s') \right),$$

where $\mathcal{S}(A)$ is a set of connected subgraphs of $A$, $\lambda(s)$ is a weight of subgraph $s$, and $\mathcal{A}(\mathbf{c})$ is a set of parsing results allowed by the sequence of syntactic categories $\mathbf{c}$. Since we aim at separating syntactic and semantic preferences, feature functions for semantic probability distinguish only words, not syntactic categories. We should note that subgraphs should not be limited to an edge, i.e., the lexical dependency of two words. By taking more than one edge as a subgraph, we can represent the dependency of more than two words, although existing models do not adopt such dependencies. Various ambiguities should be resolved by considering the dependency of more than two words; e.g. PP-attachment ambiguity should be resolved by the dependency of three words.

Consequently, the probability model takes the following form.

$$p(A|\mathbf{w}) = \left\{ \prod_i p(c_i|w_i) \right\} \left\{ \frac{1}{Z_{\mathbf{c}}} \exp\left( \sum_{s \in \mathcal{S}(A)} \lambda(s) \right) \right\}$$

However, this model has a crucial flaw: the maximum likelihood estimation of semantics probability is intractable. This is because the estimation requires $Z_{\mathbf{c}}$ to be computed, which requires summation over $\mathcal{A}(\mathbf{c})$, exponentially many parsing results. To cope with this problem, we applied an efficient algorithm of maximum entropy estimation for feature forests (Miyao and Tsujii, 2002; Geman and Johnson, 2002). This enabled the tractable estimation of the above probability, when a set of candidates are represented in a feature forest of a tractable size.

Here, we should mention that the disadvantages of the traditional models discussed in Section 2 have been completely solved by this model. It can be applied to any parsing results given by a lexicalized grammar, does not require the independence assumption, and is defined as a combination of syntax and semantics probabilities, where the semantics probability is a discriminative model that selects a parsing result from the set of candidates given by the syntax probability.

## 4 Experiments

The model proposed in Section 3 is generally applicable to any lexicalized grammars, and this section reports the evaluation of our model with a wide-coverage LTAG grammar, which is automatically acquired from the Penn Treebank (Marcus et al., 1994) Sections 02–21. The grammar was acquired by an algorithm similar to (Xia, 1999), and consisted of 2,105 elementary trees, where 1,010 were initial trees and 1,095 were auxiliary ones. The coverage of the grammar against Section 22 (1,700 sentences) was 92.6% (1,575 sentences) in a weak sense (i.e., the grammar could output a structure consistent with the bracketing in the test corpus), and 68.0% (1,156 sentences) in a strong sense (i.e., the grammar could output exactly the correct derivation).

Since the grammar acquisition algorithm could output derivation trees for the sentences in the training corpus (Section 02–21), we used them as a training set of the probability model. The model of syntax probability was estimated with syntactic categories appearing in the training set. For estimating the semantics probability, a parser produced all possible derivation trees for each sequence of syntactic categories (corresponding to each sentence) in the training set, and the obtained derivation trees, i.e., $\mathcal{A}(\mathbf{c})$, are passed to a maximum entropy estimator. By applying the grammar acquisition algorithm to Section 22, we obtained the derivation trees of the sentences in this section, and from this set we prepared a test set by eliminating non-sententials, long sentences (including more than 40 words), sentences not covered by the grammar, and sentences that caused time-outs in parsing. The resulting set consisted of 917 derivation trees.

The following three disambiguation models were prepared using the training set.

***syntax*** Only composed of the syntax probability, i.e., $p(\mathbf{c}|\mathbf{w})$

***traditional*** Similar to our model, but semantics probability $p(A|\mathbf{c})$ was decomposed into the probabilities of the primitive dependencies of two words as in the traditional modeling, i.e., this model is an inconsistent probability model

***our model*** The model by maximum entropy estimation for feature forests

The syntax probability was a unigram model, and contexts around the word such as previous words/categories were not used. Hence, it includes only syntactic preferences of words. The semantics parts of *traditional* and *our model* were maximum entropy models, where exactly the same set of features were used, i.e., the difference between the two models was only in an event representation: derivation trees were decomposed into primitive dependencies in *traditional*, while in *our model* they were represented by a feature forest without decomposition. Hence, we can evaluate the effects of applying maximum entropy estimation for feature forests by comparing *our model* with *traditional*. While *our model* allowed features to be incorporated that were not limited to the dependencies of two words (Section 3), the models used throughout the experiments only included features of the dependencies of two words. The semantics probabilities were developed with two sets of features includ-

| | *exact* | *partial* |
|---|---|---|
| *syntax* | 73.4 | 77.3 |
| *traditional* | 79.2 | 83.4 |
| *our model* | 79.6 | 83.6 |

Table 1: Accuracy of dependencies (1)

| | *exact* | *partial* |
|---|---|---|
| *syntax* | 73.4 | 77.3 |
| *traditional* | 79.6 | 83.6 |
| *our model* | 78.9 | 82.8 |

Table 2: Accuracy of dependencies (2)

ing surface forms/POSs of words, the labels of dependencies (substitution/adjunction), and the distance between two words. The first feature set had 283,755 features and the other had 150,156 features excluding fine-grained features of the first set. There were 701,819 events for *traditional*, and 32,371 for *our model*. The difference in the number of events was caused by the difference in the units of events, i.e., an event corresponded to a dependency in *traditional*, while it corresponded to a sentence in *our model*.

The parameters of the models were estimated by the limited-memory BFGS algorithm (Nocedal, 1980) with a Gaussian distribution as the prior probability distribution for smoothing (Chen and Rosenfeld, 1999) implemented in a maximum entropy estimator for feature forests (Miyao, 2002). The estimation for *traditional* was converged in 67 iterations in 127 seconds, and *our model* in 29 iterations in 111 seconds on a Pentium III 1.26-GHz CPU with 4 GB of memory. These results reveal that the estimation with *our model* is comparatively efficient with *traditional*. The parsing algorithm was CKY-style parsing with beam thresholding, which was similar to ones used in (Collins, 1996; Clark et al., 2002). Although we needed to compute normalizing factor $Z_{\mathbf{c}}$ to obtain probability values, we used unnormalized products as the preference score for beam thresholding, following (Clark et al., 2002). We did not use any preprocessing such as supertagging (Joshi and Srinivas, 1994) and the parser searched for the most plausible derivation tree from the derivation forest in terms of the probability given by the combination of syntax and semantics probabilities.

Tables 1 and 2 list the accuracy of dependencies, i.e., edges in derivation trees, for each model with two sets of features for the semantics model[3]. Since in derivation trees each word in a sentence depends on one and only one word (see Figure 3), the accuracy is the number of correct edges divided by the number of all edges in the tree. The *exact* column indicates the ratio of dependencies where the syntactic category, the argument position, and the dependee head word of the argument word are correctly output. The *partial* column shows the ratio of dependencies where the words are related regardless of the label. We should note that the *exact* measure is a very stringent because the model must select the correct syntactic category from 2,105 categories.

First, we can see that *syntax* achieved a high level of accuracy although it was not quite sufficient yet. We think this was because the grammar could adequately restrict the possible structure of parsing results, and the disambiguation model tried to search for the most probable structure from the candidates allowed by the grammar. Second, *traditional* and *our model* recorded significantly higher accuracy than *syntax*. The accuracy of *our model* was almost matched *traditional*, which proved the validity of probabilistic modeling with maximum entropy estimation for feature forests. The differences between *traditional* and *our model* were insignificant and the results proved that a consistent probability model of parsing can be built without the independence assumption, and attains performance that rivals the traditional models in terms of parsing accuracy.

We should note that accuracy can further be improved with our model because it allows other features to be incorporated that were not used in these experiments because the model is not rely on the decomposition into the dependencies of two words. Another possibility to increase the accuracy is to refine the LTAG grammar. Although we assumed that all syntactic constraints were expressed with syntactic categories (Section 3), i.e., elementary trees, the grammar used in the experiments were not augmented with feature structures and not sufficiently restrictive to eliminate syntactically invalid structures. Since our model did not include the preferences of syntactic relations of words, we expect the refinement of the grammar will greatly improve the accuracy.

## 5 Conclusion

This paper described a novel model for syntactic disambiguation based on lexicalized grammars, where the model selects the most probable parsing result from the candidates allowed by a lexicalized grammar. Since lexicalized grammars can restrict the possible structure of parsing results, the probabilistic model cannot simply be decomposed into independent events as in the existing disambiguation models for parsing. By applying a maximum entropy model for feature forests, we achieved probabilistic modeling without decomposition. Through experiments, we proved the syntax-only model could record with high level of accuracy with a lexicalized grammar, and maximum entropy estimation for fea-

[3]Since the features of the syntax part were not changed, the results for *syntax* are exactly the same.

ture forests could attain competitive accuracy compared to the traditional model. We see this work as the first step in the application of linguistically motivated grammars to the parsing of real-world texts as well as the evaluation of linguistic theories by consulting extensive corpora.

Future work should include the application of our model to other lexicalized grammars including HPSG. The development of sophisticated parsing strategies is also required to improve the accuracy and efficiency of parsing. Since parsing results of lexicalized grammars such as HPSG and CCG can include non-local dependencies, we cannot simply apply well-known parsing strategies, such as beam thresholding, which assume the local computation of preference scores. Further investigations must be left for future research.

## References


Steven P. Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4).

Adam L. Berger, Stephen A. Della Pietra, and Vincent. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of 14th National Conference on Artificial Intelligence*, pages 598–603.

Stanley Chen and Ronald Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon University.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn Treebank. In *Proceedings of 6th IWPT*.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of ACL 2000*, pages 456–463.

Stephen Clark, Julia Hockenmaier, and Mark Steedman. 2002. Building deep dependency structures with a wide-coverage CCG parser. In *Proceedings of 40th ACL*.

Michael Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of 34th ACL*, pages 184–191.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of 35th ACL*.

Stuart Geman and Mark Johnson. 2002. Dynamic programming for parsing and estimation of stochastic unification-based grammars. In *Proceedings of 40th ACL*, pages 279–286.

Julia Hockenmaier and Mark Steedman. 2002a. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of 3rd LREC*.

Julia Hockenmaier and Mark Steedman. 2002b. Generative models for statistical parsing with Combinatory Categorial Grammar. In *Proceedings of 40th ACL*, pages 335–342.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of 37th ACL*, pages 535–541.

Aravind K. Joshi and B. Srinivas. 1994. Disambiguation of super parts of speech (or supertags): Almost parsing. In *Proceedings of 17th COLING*, pages 161–165.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proceedings of HLT 2002*.

Yusuke Miyao. 2002. Amis – a maximum entropy estimator for feature forests. Available via http://www-tsujii.is.s.u-tokyo.ac.jp/%7Eyusuke/amis/.

Jorge Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–783.

Stefan Riezler, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of 38th ACL*.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of 40th ACL*.

Ivan A. Sag and Thomas Wasow. 1999. *Syntactic Theory – A Formal Introduction*. CSLI Lecture Notes no. 92. CSLI Publications.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized grammars': Application to tree adjoining grammars. In *Proceedings of 12th COLING*, pages 578–583.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of 5th NLPRS*.