Robust Learning, Smoothing, and Parameter Tying on Syntactic Ambiguity Resolution

Tung-Hui Chiang*
National Tsing Hua University

Yi-Chung Lin*†
National Tsing Hua University

Keh-Yih Su*
National Tsing Hua University

Statistical approaches to natural language processing generally obtain the parameters by using the maximum likelihood estimation (MLE) method. The MLE approaches, however, may fail to achieve good performance in difficult tasks, because the discrimination and robustness issues are not taken into consideration in the estimation processes. Motivated by that concern, a discrimination- and robustness-oriented learning algorithm is proposed in this paper for minimizing the error rate. In evaluating the robust learning procedure on a corpus of 1,000 sentences, 64.3% of the sentences are assigned their correct syntactic structures, while only 53.1% accuracy rate is obtained with the MLE approach.

In addition, parameters are usually estimated poorly when the training data is sparse. Smoothing the parameters is thus important in the estimation process. Accordingly, we use a hybrid approach combining the robust learning procedure with the smoothing method. The accuracy rate of 69.8% is attained by using this approach. Finally, a parameter tying scheme is proposed to tie those highly correlated but unreliably estimated parameters together so that the parameters can be better trained in the learning process. With this tying scheme, the number of parameters is reduced by a factor of 2,000 (from 8.7×10^8 to 4.2×10^5), and the accuracy rate for parse tree selection is improved up to 70.3% when the robust learning procedure is applied on the tied parameters.

1. Introduction

Resolution of syntactic ambiguity has been a focus in the field of natural language processing for a long time. Both rule-based and statistics-based approaches have been proposed to attack this problem in the past. For rule-based approaches, knowledge is induced by linguistic experts and is encoded in terms of rules. Since a huge amount of fine-grained knowledge is usually required to solve ambiguity problems, it is quite difficult for a rule-based approach to acquire such kinds of knowledge. In addition, the maintenance of consistency among the inductive rules is by no means easy. Therefore, a rule-based approach, in general, fails to attain satisfactory performance for large-scale applications.

In contrast, a statistical approach provides an *objective* measuring function to evaluate all possible alternative structures in terms of a set of parameters. Generally, the

* National Tsing Hua University, Department of Electrical Engineering, Hsinchu, Taiwan 300, R.O.C.
† Email: kysu@bdc.com.tw.

statistics of parameters are estimated from a training corpus by using well-developed statistical theorems. The linguistic *uncertainty* problems can thus be resolved on a solid mathematical basis. Moreover, the knowledge acquired by a statistical method is always *consistent* because all the data in the corpus are jointly considered during the acquisition process. Hence, compared with a rule-based method, the time required for knowledge acquisition and the cost needed to maintain consistency among the acquired knowledge sources are significantly reduced by adopting a statistical approach.

Among the statistical approaches, Su and Chang (1988) and Su et al. (1991) proposed a unified scoring function for resolving syntactic ambiguity. With that scoring function, various knowledge sources can be unified in a uniform formulation. Previous work has demonstrated that this scoring function is able to provide high discrimination power for a variety of applications (Su, Chiang, and Lin 1992; Chen et al. 1991; Su and Chang 1990). In this paper, we start with a baseline system based on this scoring function, and then proceed with different proposed enhancement methods. A test set of 1,000 sentences, extracted from technical manuals, is used for evaluation. A performance of 53.1% accuracy rate for parse tree selection is obtained for the baseline system, when the parameters are estimated by using the maximum likelihood estimation (MLE) method.

Note that it is the *ranking* of competitors, instead of the likelihood value, that directly affects the performance of a disambiguation task. Maximizing the likelihood values on the training corpus, therefore, does not necessarily lead to the minimum error rate. In addition, the statistical variations between the training corpus and real tasks are usually not taken into consideration in the estimation procedure. Thus, minimizing the error rate on the training corpus does not imply minimizing the error rate in the task we are really concerned with.

To deal with the problems described above, a variety of discrimination-based learning algorithms have been adopted extensively in the field of speech recognition (Bahl et al. 1988; Katagiri et al. 1991; Su and Lee 1991, 1994). Among those approaches, the robustness issue was discussed in detail by Su and Lee (1991, 1994) in particular, and encouraging results were observed. In this paper, a discrimination oriented adaptive learning algorithm is first derived based on the scoring function mentioned above and probabilistic gradient descent theory (Amari 1967; Katagiri, Lee, and Juang 1991). The parameters of the scoring function are then learned from the training corpus using the discriminative learning algorithm. The accuracy rate for parse tree selection is improved to 56.4% when the discriminative learning algorithm is applied.

In addition to the discriminative learning algorithm described above, a robust learning procedure is further applied in order to consider the possible statistical variations between the training corpus and the real task. The robust learning process continues adjusting the parameters even though the input training token has been correctly recognized, until the score difference between the correct candidate and the top competitor exceeds a preset threshold. The reason for this is to provide a tolerance zone with a large margin for better preserving the correct ranking orders for data in real tasks. An accuracy rate of 64.3% for parse tree selection is attained after this robust learning algorithm is used.

The above-mentioned robust learning procedure starts with the parameters obtained by the maximum likelihood estimation method. However, the MLE is notoriously unreliable when there is insufficient training data. The MLE for the probability of a null event is zero, which is generally inappropriate for most applications. To avoid the sparse training data problem, the parameters are first estimated by various parameter smoothing methods (Good 1953; Katz 1987). An accuracy rate for parse tree selection is improved to 69.8% by applying the robust learning procedure to the

smoothed parameters. This result demonstrates that a better initial estimate of the parameters gives the robust learning procedure a chance to obtain better results when many local maximal points exist.

Finally, a parameter tying scheme is proposed to reduce the number of parameters. In this approach, some less reliably estimated but highly correlated parameters are tied together, and then trained through the robust learning procedure. The probabilities of the events that never appear in the training corpus can thus be trained more reliably. This hybrid (tying + robust learning) approach reduces the number of parameters by a factor of 2,000 (from 8.7×10^8 to 4.2×10^5) and achieves 70.3% accuracy rate for parse tree selection.

This paper is organized as follows. A unified scoring function used for integrating knowledge from lexical and syntactic levels is introduced in Section 2. The results of using the unified scoring function are summarized in Section 3. In Section 4, the discrimination- and robustness-oriented learning algorithm is derived. The effects of the parameter smoothing techniques on the robust learning procedure are investigated in Section 5. Next, the parameter tying scheme used to enhance parameter training and reduce the number of parameters is described in Section 6. Finally, we discuss our conclusions and describe the direction of future work.

2. A Unified Probabilistic Score Function

Linguistic knowledge, including knowledge of lexicon, syntax, and semantics, is essential for resolving syntactic ambiguities. To integrate various knowledge sources in a uniform formulation, a unified probabilistic scoring function was proposed by Su et al. (1991). This scoring function has been successfully applied to resolve ambiguity problems in an English-to-Chinese machine translation system (**BehaviorTran**) (Chen et al. 1991) and a spoken language processing system (Su, Chiang, and Lin 1991; 1992). The unified probabilistic scoring function derived for the syntactic disambiguation task is summarized in the following sections.

2.1 Definition

An illustration of syntactic ambiguities for an input sentence W ($= w_1^n = \{w_1, w_2, \ldots, w_n\}$) is shown in Figure 1, where w_i ($i = 1, n$) stands for the ith word of the input sentence. In this figure, Lex_k ($1 \leq k \leq m$) stands for the kth lexical sequence out of M possible sequences. $Syn_{j,k}$ ($1 \leq j \leq N_k$) is the jth alternative syntactic structure corresponding to Lex_k, and N_k is the number of possible syntactic structures associated with Lex_k. The disambiguation process is formulated as the process of finding the most plausible syntactic structure $Syn_{\hat{j},\hat{k}}$ for the input word sequence. In other words, this process is to find the index $(\hat{j}, \hat{k})$ such that $P(Syn_{\hat{j},\hat{k}}, Lex_{\hat{k}} \mid w_1^n)$ represents the maximum value among different syntactic structures, as shown in Equation 1:

$$\left(\hat{j}, \hat{k}\right) = \underset{j,k}{\operatorname{argmax}}\{P(Syn_{j,k}, Lex_k \mid w_1^n)\} \tag{1}$$

The *integrated scoring function* for the syntactic structure $Syn_{j,k}$ is defined as

$$\begin{aligned} Score(Syn_{j,k}) &\equiv P(Syn_{j,k}, Lex_k \mid w_1^n) \\ &= P(Syn_{j,k} \mid Lex_k, w_1^n) \times P(Lex_k \mid w_1^n) \\ &= S_{syn}(Syn_{j,k}) \times S_{lex}(Lex_k) \end{aligned} \tag{2}$$

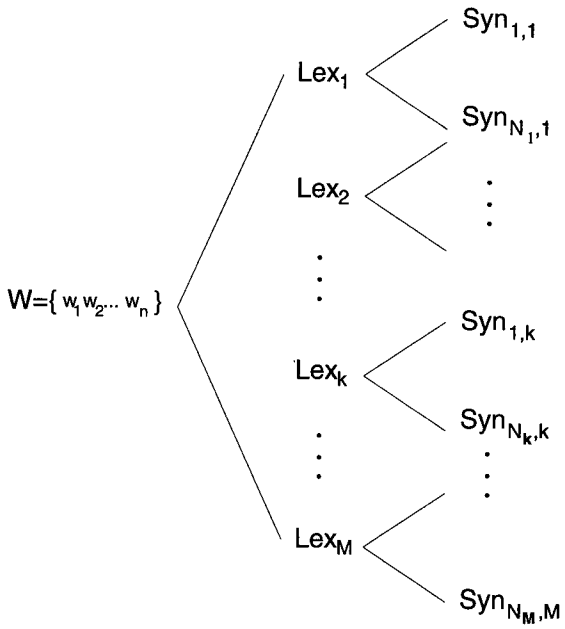

Figure 1
An illustration of the syntactic ambiguities for an input word sequence W.

where $S_{syn}(Syn_{j,k})$ $(= P(Syn_{j,k} \mid Lex_k, w_1^n))$ denotes the *syntactic scoring function*, and $S_{lex}(Lex_k)(= P(Lex_k \mid w_1^n))$ denotes the *lexical scoring function*.

In the following derivation, we assume that little additional information can be provided by the words w_1^n to the syntactic structure $Syn_{j,k}$ after the lexical sequence Lex_k is given.[1] The syntactic scoring function is thus approximated as the following equation:

$$S_{syn}(Syn_{j,k}) = P(Syn_{j,k} \mid Lex_k, w_1^n) \approx P(Syn_{j,k} \mid Lex_k) \tag{3}$$

Accordingly, the integrated scoring function $P(Syn_{j,k}, Lex_k \mid w_1^n)$ is represented as follows:

$$P(Syn_{j,k}, Lex_k \mid w_1^n) \approx P(Syn_{j,k} \mid Lex_k) \times P(Lex_k \mid w_1^n). \tag{4}$$

Detailed discussion of the lexical and syntactic scoring functions is given in the following sections.

2.2 Lexical Score Function

The *lexical score* for the kth lexical sequence Lex_k associated with the input word sequence w_1^n is expressed as follows (Chiang, Lin, and Su 1992):

$$S_{lex}(Lex_k) \quad = \quad P(Lex_k \mid w_1^n) = P\left(c_{k,1}^{k,n} \mid w_1^n\right)$$

1 Note that the effect of word sense on the syntax disambiguation task is considered in a semantic scoring function, which is not discussed in this paper. Interested readers are referred to Chang, Luo, and Su (1992).

$$\begin{aligned} &= \frac{P\left(w_1^n \mid c_{k,1}^{k,n}\right) \times P\left(c_{k,1}^{k,n}\right)}{P(w_1^n)} \\ &= \frac{S_{lex}^*(Lex_k)}{P(w_1^n)} \end{aligned} \tag{5}$$

where $c_{k,i}$ stands for the part of speech assigned to w_i. Since $P(w_1^n)$ is the same for all possible lexical sequences, it can be ignored without affecting the final results. Therefore, we use $S_{lex}^*(\cdot)$ instead of $S_{lex}(\cdot)$ in the following derivation.

Like the standard tagging procedures (Garside, Leech, and Sampson 1987; Church 1989; Merialdo 1991), the probability terms $P(w_1^n \mid c_{k,1}^{k,n})$ and $P(c_{k,1}^{k,n})$ in Equation 5 can be approximated as follows, respectively:

$$P\left(w_1^n \mid c_{k,1}^{k,n}\right) = \prod_{i=1}^{n} P\left(w_i \mid w_1^{i-1}, c_{k,1}^{k,n}\right) \approx \prod_{i=1}^{n} P\left(w_i \mid c_{k,i}\right). \tag{6}$$

$$\begin{aligned} P\left(c_{k,1}^{k,n}\right) &= \prod_{i=1}^{n} P\left(c_{k,i} \mid c_{k,1}^{k,i-1}\right) \\ &\approx \begin{cases} \prod_{i=1}^{n} P(c_{k,i} \mid c_{k,i-1}), & \text{bigram model} \\ \prod_{i=1}^{n} P(c_{k,i} \mid c_{k,i-1}, c_{k,i-2}), & \text{trigram model} \end{cases} \end{aligned} \tag{7}$$

Therefore, the lexical score $S_{lex}(Lex_k)$ is expressed as:

$$\begin{aligned} S_{lex}^*(Lex_k) &\approx \prod_{i=1}^{n} P\left(c_{k,i} \mid c_{k,1}^{k,i-1}\right) \times P(w_i \mid c_{k,i}) \\ &\approx \begin{cases} \prod_{i=1}^{n} P(c_{k,i} \mid c_{k,i-1}) \times P(w_i \mid c_{k,i}), & \text{bigram model} \\ \prod_{i=1}^{n} P(c_{k,i} \mid c_{k,i-1}, c_{k,i-2}) \times P(w_i \mid c_{k,i}), & \text{trigram model} \end{cases} \end{aligned} \tag{8}$$

2.3 Syntactic Scoring Function

Conventional stochastic context-free grammar (CFG) approaches (Wright and Wrigley 1991) evaluate the likelihood probability of a syntactic tree by computing the product of the probabilities associated with the grammar rules being applied. Such a formulation implies that the application of a rule is both independent of the applications of the other rules, and independent of the context under which a context-free rule is applied. However, a language that can be expressed with a CFG does not imply that the associated rules can be applied in an independent and context-free manner, as implicitly assumed by a stochastic context-free grammar approach. To include contextual information and consider the relationship among the grammar rules, in this paper we follow the formulation in Su et al. (1989, 1991) for syntactic score evaluation.

To show the computing mechanism for the syntactic score, we take the tree in Figure 2 as an example. The basic derivation of the syntactic score includes the following

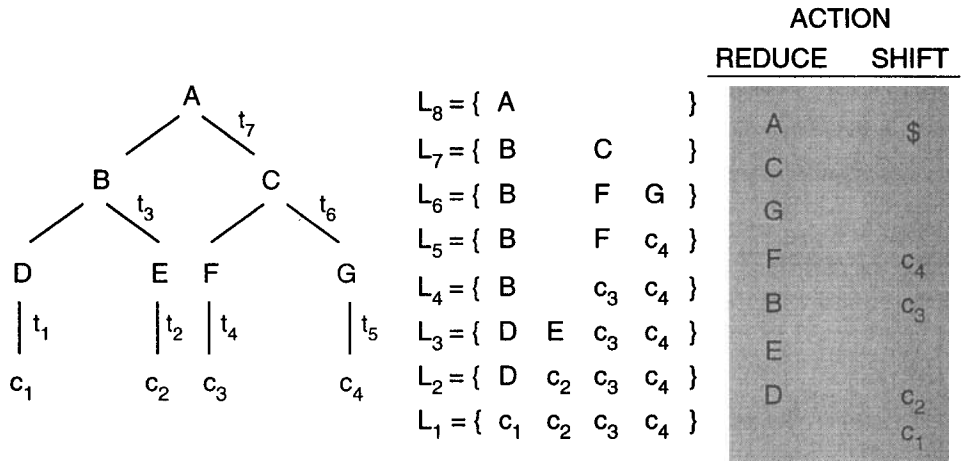

Figure 2
The decomposition of a given syntactic tree X into different phrase levels.

steps. First, the tree is decomposed into a number of *phrase levels*, such as $L_1, L_2, \ldots, L_8$ in Figure 2. A phrase level is a sequence of symbols (terminal or nonterminal) that acts as an intermediate result in parsing the input sentence, and is also called a *sentential form* in *formal language theory* (Hopcroft and Ullman 1974). In the second step, we formulate the transition between phrase levels as a *context-sensitive* rewriting process. With the formulation, each transition probability between two phrase levels is calculated by consulting a finite-length window that comprises the symbols to be reduced and their left and right contexts.

Let the label t_i in Figure 2 be the time index for the ith state transition, which corresponds to a reduce action, and L_i be the ith phrase level. Then the syntactic score of the tree in Figure 2 is defined as:

$$\begin{aligned} S_{syn}(Tree_X) &\equiv P(L_8, L_7, \ldots, L_2 \mid L_1) \\ &= P(L_8 \mid L_7, \ldots, L_1) \times P(L_7 \mid L_6, \ldots, L_1) \times \cdots \times P(L_2 \mid L1) \\ &\approx P(L_8 \mid L_7) \times P(L_7 \mid L_6) \times \cdots \times P(L_2 \mid L1). \end{aligned} \tag{9}$$

The transition probability between two phrase levels, say $P(L_7 \mid L_6)$, is the product of the probabilities of two events. Taking $P(L_7 \mid L_6)$ as an example, the first probability corresponds to the event that $\{F, G\}$ are the constituents to be reduced, and the second probability corresponds to the event that they are reduced to C. The transition probability can thus be expressed as follows:

$$\begin{aligned} P(L_7 \mid L_6) &= P(F, G \textit{ are reduced } \mid \textit{input is } \{B, F, G\}) \\ &\quad \times P(C \leftarrow FG \mid F, G \textit{ are reduced; input is } \{B, F, G\}). \end{aligned} \tag{10}$$

According to the results of our experiments, the first term is equal to one in most cases, and it makes little contribution to discriminating different syntactic structures. In addition, to simplify the computation, we approximate the full context $\{B, F, G\}$ with a window of finite length around $\{F, G\}$. The formulation for the syntactic scoring

function can thus be expressed as follows:

$$\begin{aligned} S_{syn}(Tree_X) &\approx P(A \mid \{\emptyset\}, B, C, \{\$\}) \times P(C \mid \{B\}, F, G, \{\$\}) \\ &\quad \times \cdots \times P(D \mid \{\emptyset\}, c_1, \{c_2, c_3, c_4\}) \\ &\approx P(A \mid l_7, B, C, r_7) \times P(C \mid l_6, F, G, r_6) \times \cdots \times P(D \mid l_1, c_1, r_1), \end{aligned} \quad (11)$$

where $Tree_X$ is the parse tree X, \$ and $\emptyset$ correspond to the end-of-sentence marker and the null symbol, respectively; and l_i and r_i represent the left and right contexts to be consulted in the ith phrase level, respectively. In the above equation, it is assumed that each phrase level is highly correlated with its immediately preceding phrase level but less correlated with other preceding phrase levels. In other words, the inter-level correlation is assumed to be a first-order Markov process. In addition, for computational feasibility, only a finite number of left and right contextual symbols are considered in the formulation. If **M** left context symbols and **N** right context symbols are consulted in evaluating Equation 9, the model is said to operate in the $\mathbf{L_M R_N}$ mode.

Notice that the last formula in Equation 9 corresponds to the rightmost derivation sequence in a generalized LR parser *with left and right contexts* taken into account (Su et al. 1991). Such a formulation is particularly useful for a generalized LR parsing algorithm, in which context-sensitive processing power is desirable. Although the context-sensitive model in the above equation provides the ability to deal with *intra-level context-sensitivity*, it fails to catch *inter-level correlation*. In addition, the formulation of Equation 9 will result in the *normalization problem* (Su et al. 1991; Briscoe and Carroll 1993) when various syntactic trees have different number of nodes. An alternative formulation, which compacts the highly correlated phrase levels into a single one, was proposed by Su et al. (1991) to resolve the normalization problem. For instance, for the syntactic tree in Figure 2, the syntactic score for the modified formulation is expressed as follows:

$$\begin{aligned} S_{syn}(Tree_X) &\approx P(L_8, L_7, L_6 \mid L_5) \times P(L_5 \mid L_4) \times P(L_4, L_3 \mid L_2) \times P(L_2 \mid L_1) \\ &\approx P(L_8 \mid L_5) \times P(L_5 \mid L_4) \times P(L_4 \mid L_2) \times P(L_2 \mid L_1). \end{aligned} \quad (12)$$

Each pair of phrase levels in the above equation corresponds to a change in the LR parser's stack before and after an input word is consumed by a *shift* operation. Because the total number of shift actions, equal to the number of product terms in Equation 12, is always the same for all alternative syntactic trees, the normalization problem is resolved in such a formulation. Moreover, the formulation in Equation 12 provides a way to consider both *intra-level context-sensitivity* and *inter-level correlation* of the underlying context-free grammar. With such a formulation, the capability of *context-sensitive parsing* (in probabilistic sense) can be achieved with a *context-free grammar*.

It is interesting to compare our frameworks (Su et al. 1991) with the work by Briscoe and Carroll (1993) on probabilistic LR parsing. Instead of assigning probabilities to the production rules as a conventional stochastic context-free grammar parser does, Briscoe and Carroll distribute probability to each state so that the probabilities of the transitions from a state sum to one; the preference to a SHIFT action is based on one right context symbol (i.e., the lookahead symbol), and the preference for a REDUCE action depends on the lookahead symbol and the previous state reached after the REDUCE action. With such an approach, it is very easy to implement (mildly) context-sensitive probabilistic parsing on existing LR parsers, and the probabilities can be easily trained. The probabilities assigned to the states implicitly imply different preferences for left-hand side contextual environment of the reduced symbol, since a

state, in general, can indicate part of the past parsing history (i.e., the left context) from which the current reduced symbol follows.

However, because of the implicit encoding of the parsing history, a state may fail to distinguish some left contextual environments correctly. This is not surprising, because the LR parsing table generator would merge certain states according to the context-free grammar and the closure operations on the sets of items. Therefore, there are cases in which the same string is reduced, under *different* left contexts, to the same symbol at the same state and return to the same state after reduction. For instance, if several identical constructs, e.g., [X → a], are allowed in a recursive structure X*, and the input contains a Y followed by three (or more) consecutive Xs, e.g., "YXXX," then the reduction of the second and third Xs will return to the same state after the same rule is applied at that state. Under such circumstances, the associated probabilities for these two REDUCE actions will be identical and thus will not reflect the different preferences between them.

In our framework, it is easy to tell that the first REDUCE action is applied when the two left context symbols are {Y,X}, and the second REDUCE is applied when the left context is two Xs under an L2R1 mode of operation. Because such recursion is not rare, for example, in groups of adjectives, nouns, conjunction constructs, prepositional phrases in English, the estimated scores will be affected by such differences. In other words, we use context symbols explicitly and directly to evaluate the probabilities of a substructure instead of using the parsing state to implicitly encode past history, which may fail to provide a sufficient characterization of the left context. In addition, explicitly using the left context symbols allows easy use of smoothing techniques, such as deleted interpolation (Bahl, Jelinek, and Mercer 1983), clustering techniques (Brown et al. 1992), and model refinement techniques (Lin, Chiang, and Su 1994) to estimate the probabilities more reliably by changing the window sizes of the context and weighting the various estimates dynamically. This kind of improvement is desirable when the training data is limited.

Furthermore, Briscoe and Carroll (1993) use the geometric mean of the probabilities, not their product, as the preference score, to avoid biasing their procedure in favor of parse trees that have a smaller number of nodes (i.e., a smaller number of rules being applied.) The geometric mean, however, fails to fit into the probabilistic framework for disambiguation. In our approach, such a normalization problem is avoided by considering a group of highly correlated phrase levels as a single phrase level and evaluating the sequence of transitions for such phrase levels between the SHIFT actions. Alternatively, it is also possible to consider each group of highly correlated phrase levels as a joint event for evaluating its probability when enough data is available. The optimization criteria are thus not compromised by the topologies of the parse trees, because the number of SHIFT actions (i.e., the number of input tokens) is fixed for an input sentence.

3. Baseline Model

To establish a benchmark for examining the power of the proposed algorithms, we begin with a baseline system, in which the parameters are estimated by using the MLE method. Later, we will show how to improve the baseline model with the proposed enhancement mechanisms.

3.1 Experimental Setup

First of all, 10,000 parsed sentences generated by **BehaviorTran** (Chen et al. 1991), a commercialized English-to-Chinese machine translation system designed by Behavior

Design Corporation (BDC), were collected. The domain for this corpus is computer manuals and documents. The correct parts of speech and parse trees for the collected sentences were verified by linguistic experts. The corpus was then randomly partitioned into a training set of 9,000 sentences and a test set of the remaining 1,000 sentences to eliminate possible systematic biases. The average number of words per sentence for the training set and the test set were 13.9 and 13.8, respectively. In the training set, there were 1,030 unambiguous sentences, while 122 sentences were unambiguous in the test set. On the average, there were 34.2 alternative parse trees per sentence for the training set, and 31.2 for the test set. If we excluded those unambiguous sentences, there were 38.49 and 35.38 alternative syntactic structures per sentence for the training set and the test set, respectively.

3.1.1 Lexicon and Phrase Structure Rules. In the current system, there are 10,418 distinct lexicon entries, extracted from the 10,000-sentence corpus. The grammar is composed of 1,088 phrase structure rules that are expressed in terms of 35 terminal symbols (parts of speech) and 95 nonterminal symbols.

3.1.2 Language Models. Usually, a more complex model requires more parameters; hence it frequently introduces more estimation error, although it may lead to less modeling error. To investigate the effects of model complexity and estimation error on the disambiguation task, the following models, which account for various lexical and syntactic contextual information, were evaluated:

1. Lex(L1[2]) + Syn(L1): this model uses a bigram model in computing lexical scores and the L1 mode of operation in computing syntactic scores. The number of parameters required is $(10,418 \times 35) + (35 \times 35) + (96,699 \times 95) = 9,492,260$.[3]

2. Lex(L2)+Syn(L1): this model uses a trigram model in computing lexical scores and the L1 mode of operation in computing syntactic scores. The number of parameters required is $(10,418 \times 35) + (35 \times 35 \times 35) + (96,699 \times 95) = 9,533,910$.

3. Lex(L1)+Syn(L2): this model uses a bigram model in computing lexical scores and the L2 mode of operation in computing syntactic scores. The number of parameters required is $(10,418 \times 35) + (35 \times 35) + (96,699 \times 95 \times 95) = 873,014,330$.

4. Lex(L2)+Syn(L2): this model uses a trigram model in computing lexical scores and the L2 mode of operation in computing syntactic scores. The number of parameters required is $(10,418 \times 35) + (35 \times 35 \times 35) + (96,699 \times 95 \times 95) = 873,055,980$.

2 L1 means to consult one left-hand side part of speech, and L2 means to consult two left-hand side parts of speech.

3 The number of parameters for Lex(L1) and Lex(L2) modes is $(N_w \times N_t) + N_t^2$ and $(N_w \times N_t) + N_t^3$, respectively, where $N_w(= 10,418)$ stands for the number of words in the lexicon, and $N_t(= 35)$ denotes the number of distinct terminal symbols (parts of speech). The number of parameters for Syn(L1) and Syn(L2) modes is $N_p \times N_{nt}$ and $N_p \times N_{nt}^2$, respectively, where $N_{nt}(= 95)$ denotes the number of nonterminal symbols, and $N_p(= 96,699)$ is the number of patterns corresponding to all possible reduce actions. Each pattern is represented as a pair of [current symbols, reduced symbol]. For instance, [{B,C},{A}] is the pattern corresponding to the reduce action $A \leftarrow BC$ in Figure 2.

3.1.3 Performance Evaluations. We will evaluate the above-mentioned models in two measures: *accuracy rate* and *selection power*. The measure of *accuracy rate* of parse tree selection has been widely used in the literature. However, this measure is unable to identify which model is better if the average number of alternative syntactic structures in various tasks is different. For example, a language model with 91% accuracy rate for a task with an average of 1.1 alternative syntactic structures per sentence, which corresponds to the performance of random selection, is by no means better than the language model that attains 70% accuracy rate when there are an average of 100 alternative syntactic structures per sentence. Therefore, a measure, namely **Selection Power (SP)**, is proposed in this paper to give additional information for evaluation. SP is defined as the average *selection factor* (S_F) of the disambiguation mechanism on the task of interest. The selection factor for an input sentence is defined as the least proportion of all possible alternative structures that includes the selected syntactic structure.[4] A smaller SP value would, in principle, imply better disambiguation power. Formally, *SP* is expressed as

$$SP \equiv E[S_F] \approx \frac{1}{N}\sum_{i=1}^{N} s_f(i) = \frac{1}{N}\sum_{i=1}^{N} \frac{r_i}{n_i} \tag{13}$$

where $s_f(i) = \frac{r_i}{n_i}$ is the selection factor for the ith sentence; n_i is the total number of alternative syntactic structures for the ith sentence; r_i is the rank of the most preferred candidate. The selection power for a disambiguation mechanism basically serves as an indicator of the selection ability that includes the most preferred candidate within a particular (N-best) region. A mechanism with a smaller SP value is more likely to include the most preferred candidate for some given N-best hypotheses.

In general, the measures of accuracy rate and the selection power are highly correlated. But it is more informative to report performance with both accuracy rate and selection power. Selection power supplements accuracy rate when two language models to be compared are tested on different tasks.

3.2 Summary of Baseline Results

The performances of the various models in terms of accuracy rate and selection power are shown in Table 1; the values in the parentheses correspond to performance excluding unambiguous sentences. Table 1 shows that better performance (both in terms of accuracy rate and selection power) can be attained when more contextual information is consulted (or when more parameters are used). The improvement in resolution of syntactic ambiguity by using more lexical contextual information, however, is not statistically significant[5] when the consulted contextual information in the syntactic models is fixed. For instance, the test set performance for the Lex(L1)+Syn(L2) model is 52.8%, while the performance for the Lex(L2)+Syn(L2) model is only 53.1%. With this small performance difference, we cannot reject the hypothesis that the performance of the Lex(L1)+Syn(L2) model is the same as that of the Lex(L1)+Syn(L2) model. On the other hand, if the consulted lexical contexts are fixed, the performance of the syntactic disambiguation process is improved significantly by using more syntactic contextual

4 The term "most preferred candidate" means the syntactic structure most preferred by people even when there is more than one arguably correct syntactic structure. However, throughout this paper, both the expressions "*most preferred* syntactic structure" and "*correct* syntactic structure" refer to the syntactic structure most preferred by our linguistic experts.

5 The conclusions drawn throughout this paper are all examined based on the testing hypothesis procedure for a significance level $\alpha = 0.01$ (Gillick and Cox 1989).

Table 1
The baseline performance: (a) training set; (b) test set. Values in parentheses correspond to performance excluding unambiguous sentences.

Model	Part-of-Speech Accuracy Rate		Parse Tree	
	in Word (%)	in Sentence (%)	Accuracy Rate (%)	Selection Power
Lex(L1)+Syn(L1)	99.62 (99.59)	95.6 (95.0)	75.4 (72.3)	0.34 (0.26)
Lex(L2)+Syn(L1)	99.64 (99.61)	95.9 (95.4)	75.8 (72.7)	0.34 (0.26)
Lex(L1)+Syn(L2)	99.67 (99.64)	96.1 (95.6)	78.7 (75.9)	0.34 (0.25)
Lex(L2)+Syn(L2)	99.69 (99.67)	96.5 (96.0)	79.0 (76.4)	0.33 (0.25)

(a) Training set performance

Model	Part-of-Speech Accuracy Rate		Parse Tree	
	in Word (%)	in Sentence (%)	Accuracy Rate (%)	Selection Power
Lex(L1)+Syn(L1)	98.89 (98.80)	88.7 (87.13)	49.3 (42.3)	0.45 (0.38)
Lex(L2)+Syn(L1)	98.93 (98.84)	88.9 (87.36)	49.7 (42.7)	0.45 (0.38)
Lex(L1)+Syn(L2)	98.82 (98.71)	88.0 (86.33)	52.8 (46.2)	0.44 (0.37)
Lex(L2)+Syn(L2)	98.89 (98.79)	88.5 (86.90)	53.1 (46.6)	0.44 (0.37)

(b) Test set performance

information. For example, a 53.1% accuracy rate is attained for the Lex(L2)+Syn(L2) model, while the accuracy rate is 49.7% for the Lex(L2)+Syn(L1) model. This result indicates that the context-free assumption adopted by most stochastic parsers might not hold.

4. Discrimination- and Robustness-Oriented Learning

Although MLE possesses many nice properties (Kendall and Stuart 1979), the criterion of maximizing likelihood value is not equivalent to that of minimizing the error rate in a training set. The maximum likelihood approach achieves *disambiguation* indirectly and implicitly through the estimation procedure. However, correct disambiguation only depends on the ranks, rather than the likelihood values, of the candidates. In other words, correct recognition will still be obtained if the score of the correct candidate is the highest, even though the likelihood values of the various candidates are estimated poorly. Motivated by this concern, a discrimination-oriented learning procedure is proposed in this paper to adjust the parameters iteratively such that the correct ranking orders can be achieved.

A general adaptive learning algorithm for minimizing the error rate in the training set was proposed by Amari (1967) using the probability descent (PD) method. The extension of PD, namely the generalized probability descent method (GPD), was also developed by Katagiri, Lee, and Juang (1991). However, minimizing the error rate in the training set cannot guarantee that the error rate in the test set is also minimized. Discrimination-based learning procedures, in general, tend to overtune the training set performance unless the number of available data is several times larger

than the number of parameters (based on our experience). Overtuning the training set performance usually causes performance on the test set to deteriorate. Hence, the *robustness* issue, which concerns the possible statistical variations between the training set and the test set, must be taken into consideration when we adopt an adaptive learning procedure. In this section, we start with a learning algorithm derived from the probabilistic descent procedure (Katagiri, Lee, and Juang 1991). The robust learning algorithm explored by Su and Lee (1991, 1994) is then introduced to enhance the robustness of the system.

4.1 Discrimination-Oriented Learning

To link the syntactic disambiguation process with the learning procedure, a discrimination function, namely $g_{j,k}(w_1^n)$, for the syntactic tree $Syn_{j,k}$, corresponding to the lexical sequence Lex_k and the input sentence (or word sequence) w_1^n, is defined as

$$g_{j,k}(w_1^n) = \log P(Syn_{j,k}, Lex_k \mid w_1^n) \tag{14}$$

Since $\log(\cdot)$ is a monotonic increasing function, we can rewrite the criterion for syntactic disambiguation in Equation 1 as the following equation:

$$\left(\hat{j}, \hat{k}\right) = \underset{j,k}{\operatorname{argmax}}\{g_{j,k}(w_1^n)\} \tag{15}$$

According to Equation 2, Equation 8, and Equation 12, the discrimination function can be further derived as follows:

$$\begin{aligned} g_{j,k}(w_1^n) &= \log S_{syn}(Syn_j) + \log S_{lex}(Lex_k) \\ &= -\left\{\sum_{i=1}^{n}\left[-\log P\left(L_{j,i} \mid L_{j,1}^{j,i-1}\right)\right] + \sum_{i=1}^{n}\left[-\log P\left(c_{k,i} \mid c_{k,1}^{k,i-1}, w_1^n\right)\right]\right\} \\ &= -\left\{\sum_{i=1}^{n}\lambda_{syn}^2(j,i) + \sum_{i=1}^{n}\lambda_{lex}^2(k,i)\right\} \\ &= -\|\Phi_{j,k}\|^2 \end{aligned} \tag{16}$$

where $\lambda_{lex}(k,i) = [-\log P(c_{k,i} \mid c_{k,1}^{k,i-1}, w_1^n)]^{1/2}$; $\lambda_{syn}(j,i) = [-\log P(L_{j,i} \mid L_{j,1}^{j,i-1})]^{1/2}$. $\Phi_{j,k} = [\lambda_{syn}(j,1), \lambda_{lex}(k,1), \ldots, \lambda_{syn}(j,n), \lambda_{lex}(k,n)]$ is regarded as a parameter vector composed of the lexical and syntactic score components, and $\|\Phi_{j,k}\|$ is defined as the Euclidean norm of the vector $\Phi_{j,k}$. However, in such a formulation, the lexical scores as well as the syntactic scores are assumed to contribute equally to the disambiguation process. This assumption is inappropriate because different linguistic information may contribute differently to various disambiguation tasks. Moreover, the preference scores related to various types of linguistic information may have different dynamic ranges. Therefore, different scores should be assigned different weights to account for both the contribution in discrimination and the dynamic ranges. The discrimination function is thus modified into the following form:

$$\begin{aligned} g_{j,k}(w_1^n) &= -\left\{w_{syn}\sum_{i=1}^{n}\lambda_{syn}^2(j,i) + w_{lex}\sum_{i=1}^{n}\lambda_{lex}^2(k,i)\right\} \\ &= -\left\|\hat{\Phi}_{j,k}\right\|^2 \end{aligned} \tag{17}$$

where w_{lex} and w_{syn} stand for the lexical and syntactic weights, respectively; they are set to 1.0 initially. $\hat{\Phi}_{j,k}$ corresponds to a transformation of the original vector $\Phi_{j,k}$ and is represented as the following equation:

$$\begin{aligned}\hat{\Phi}_{j,k} &= \left[w_{syn}^{\frac{1}{2}}\lambda_{syn}(j,1), w_{lex}^{\frac{1}{2}}\lambda_{lex}(k,1), \ldots, w_{syn}^{\frac{1}{2}}\lambda_{syn}(j,n), w_{lex}^{\frac{1}{2}}\lambda_{lex}(k,n)\right] \\ &= \left[\hat{\lambda}_{syn}(j,1), \hat{\lambda}_{lex}(k,1), \ldots, \hat{\lambda}_{syn}(j,n), \hat{\lambda}_{lex}(k,n)\right] \end{aligned} \tag{18}$$

The whole parameter set, denoted by Λ, thus includes the lexical weight, w_{lex}, the syntactic weight, w_{syn}, the lexical parameters $\Lambda_{lex} = \{\lambda_{lex}(i,j)\}_{\forall i,j}$ and the syntactic parameters $\Lambda_{syn} = \{\lambda_{syn}(i,j)\}_{\forall i,j}$; i.e.,

$$\Lambda = \{w_{lex}, w_{syn}\} \cup \Lambda_{lex} \cup \Lambda_{syn} \tag{19}$$

The decision rule for the classifier to select the desired output, according to Eq. (17), is represented as follows:

$$(\hat{j},\hat{k}) = \underset{j,k}{\operatorname{argmax}}\, g_{j,k}(w_1^n)$$

or

$$(\hat{j},\hat{k}) = \underset{j,k}{\operatorname{argmax}} \left\{ -\left\|\hat{\Phi}_{j,k}\right\|^2 \right\}. \tag{20}$$

Let the correct syntactic structure associated with the input sentence be $Syn_{\alpha,\beta}$. Then the misclassification distance, denoted by $d_{\hat{j},\hat{k}}$, for selecting the syntactic structure $Syn_{\hat{j},\hat{k}}$ as the final output is defined by the following equation:

$$\begin{aligned} d_{\hat{j},\hat{k}}(w_1^n;\Lambda) &= [-g_{\alpha,\beta}(w_1^n)]^{\frac{1}{2}} - \left[-g_{\hat{j},\hat{k}}(w_1^n)\right]^{\frac{1}{2}} \\ &= \left\|\hat{\Phi}_{\alpha,\beta}\right\| - \left\|\hat{\Phi}_{\hat{j},\hat{k}}\right\| \end{aligned} \tag{21}$$

Such a definition makes the distance be the difference of the lengths (or norms) of the score vectors in the parameter space. Furthermore, $d_{\hat{j},\hat{k}}$ is differentiable with respect to the parameters. Note that according to the definition in Equation 21, an error will occur if $d_{\hat{j},\hat{k}} > 0$, i.e., $\|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{j},\hat{k}}\|$.

Next, similar to the probabilistic-descent approach (Amari 1967), a loss function $l_{\hat{j},\hat{k}}(\Lambda)$ is defined as a nondecreasing and differentiable function of the misclassification distance; i.e., $l_{\hat{j},\hat{k}}(\Lambda) = l(d_{\hat{j},\hat{k}}(w_1^n;\Lambda))$. To approximate the *zero-one* loss function defined for the minimum-error-rate classification, the loss function, as in Amari (1967), is defined as

$$l(d_{\hat{j},\hat{k}}) = \begin{cases} \tan^{-1}\left(\frac{d_{\hat{j},\hat{k}}}{d_0}\right) & d_{\hat{j},\hat{k}} > 0 \\ 0 & otherwise \end{cases} \tag{22}$$

where d_0 is a small positive constant. It has been proved by Amari (1967) that the average loss function will decrease if the adjustments in the learning process satisfy the following equation:

$$\begin{aligned} \Lambda_{t+1} &= \Lambda_t + \delta\Lambda_t, \\ \delta\Lambda_t &= -\epsilon(t)U\nabla l(d_{\hat{j},\hat{k}}(w_1^n;\Lambda)), \end{aligned} \tag{23}$$

where $\epsilon(t)$ is a positive function, which usually decreases with time, to control the convergence speed of the learning process; U is a positive-definite matrix, which is assumed to be an identity matrix in the current implementation, and ∇ is the gradient operator. Hence, it follows from Equation 23 that the ith syntactic parameter component $\lambda_{syn}^{(t+1)}(\alpha, i)$, corresponding to the *correct candidate*, $Syn_{\alpha,\beta}$ in the $(t+1)$-th iteration, would be adjusted according to the following equation:

$$\begin{cases} \lambda_{syn}^{(t+1)}(\alpha, i) = \lambda_{syn}^{(t)}(\alpha, i) + \Delta\lambda_{syn}^{(t)}(\alpha, i), & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ \lambda_{syn}^{(t+1)}(\alpha, i) = \lambda_{syn}^{(t)}(\alpha, i), & \text{otherwise,} \end{cases} \tag{24}$$

where $\Delta\lambda_{syn}^{(t)}(\alpha, i)$ is the amount of adjustment and is computed as follows:

$$\Delta\lambda_{syn}^{(t)}(\alpha, i) = -\epsilon(t) \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_o^2} \cdot w_{syn} \cdot \frac{\lambda_{syn}^{(t)}(\alpha, i)}{\|\hat{\Phi}_{\alpha,\beta}\|} \tag{25}$$

Meanwhile, the syntactic parameter component corresponding to the *top incorrect candidate* would be adjusted according to the following formulae:

$$\begin{cases} \lambda_{syn}^{(t+1)}(\hat{\jmath}, i) = \lambda_{syn}^{(t)}(\hat{\jmath}, i) - \Delta\lambda_{syn}^{(t)}(\hat{\jmath}, i), & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ \lambda_{syn}^{(t+1)}(\hat{\jmath}, i) = \lambda_{syn}^{(t)}(\hat{\jmath}, i), & \text{otherwise,} \end{cases} \tag{26}$$

$$\Delta\lambda_{syn}^{(t)}(\hat{\jmath}, i) = -\epsilon(t) \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_0^2} \cdot w_{syn} \cdot \frac{\lambda_{syn}^{(t)}(\hat{\jmath}, i)}{\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|}$$

The learning rules for adjusting the lexical parameters can be represented in a similar manner:

1. For the lexical parameters corresponding to the correct candidates:

$$\begin{cases} \lambda_{lex}^{(t+1)}(\beta, i) = \lambda_{lex}^{(t)}(\beta, i) - \Delta\lambda_{lex}^{(t)}(\beta, i), & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ \lambda_{lex}^{(t+1)}(\beta, i) = \lambda_{lex}^{(t)}(\beta, i), & \text{otherwise,} \end{cases} \tag{27}$$

$$\Delta\lambda_{lex}^{(t)}(\beta, i) = -\epsilon(t) \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_0^2} \cdot w_{lex} \cdot \frac{\lambda_{lex}^{(t)}(\beta, i)}{\|\hat{\Phi}_{\alpha,\beta}\|}$$

2. For the lexical parameters corresponding to the top candidate:

$$\begin{cases} \lambda_{lex}^{(t+1)}(\hat{k}, i) = \lambda_{lex}^{(t)}(\hat{k}, i) - \Delta\lambda_{lex}^{(t)}(\hat{k}, i), & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ \lambda_{lex}^{(t+1)}(\hat{k}, i) = \lambda_{lex}^{(t)}(\hat{k}, i), & \text{otherwise,} \end{cases} \tag{28}$$

$$\Delta\lambda_{lex}^{(t)}(\hat{k}, i) = -\epsilon(t) \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_0^2} \cdot w_{lex} \cdot \frac{\lambda_{lex}^{(t)}(\hat{k}, i)}{\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|}$$

In addition, the syntactic and lexical weights are adjusted as follows:

$$\begin{cases} w_{syn}^{(t+1)} = w_{syn}^{(t)} + \Delta w_{syn}^{(t)}, & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ w_{syn}^{(t+1)} = w_{syn}^{(t)}, & \text{otherwise} \end{cases} \tag{29}$$

$$\Delta w_{syn}^{(t)} = -\frac{\epsilon(t)}{2} \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_0^2} \cdot \left[\frac{\sum_{i=1}^{n} \lambda_{syn}^2(\alpha, i)}{\|\hat{\Phi}_{\alpha,\beta}\|} - \frac{\sum_{i=1}^{n} \lambda_{syn}^2(\hat{\jmath}, i)}{\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|} \right]$$

$$\begin{cases} w_{lex}^{(t+1)} = w_{lex}^{(t)} + \Delta w_{lex}^{(t)}, & \text{if } \|\hat{\Phi}_{\alpha,\beta}\| > \|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|, \\ w_{lex}^{(t+1)} = w_{lex}^{(t)}, & \text{otherwise,} \end{cases} \tag{30}$$

$$\Delta w_{lex}^{(t)} = -\frac{\epsilon(t)}{2} \cdot \frac{d_0}{d_{\hat{\jmath},\hat{k}}^2 + d_0^2} \cdot \left[\frac{\sum_{i=1}^{n} \lambda_{lex}^2(\beta, i)}{\|\hat{\Phi}_{\alpha,\beta}\|} - \frac{\sum_{i=1}^{n} \lambda_{lex}^2(\hat{k}, i)}{\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\|} \right]$$

As the parameters are adjusted according to the learning rules described above, the score of the correct candidate will increase and the score of the incorrect candidate will decrease from iteration to iteration until the correct candidate is selected.

The ratio of the syntactic weight to the lexical weight, i.e., w_{syn}/w_{lex}, finally turns out to be 1.3 for the Lex(L2)+Syn(L2) model after the discriminative learning procedure is applied. This ratio varies with the adopted language models, but is always larger than 1.0. This result matches our expectation, because the syntactic score should provide more discrimination power than the lexical score in the syntactic disambiguation task.

The experimental results of using the discriminative learning procedure with 20 iterations are shown in Table 2. For comparison, the corresponding results before learning, i.e., the baseline results, are repeated in the upper row of each table entry. For the Lex(L2)+Syn(L2) model, the accuracy rate for parse tree disambiguation in the training set is improved from 79.04% to 92.77%, which corresponds to a 65.5% error reduction rate. However, only a 7.03% error reduction rate is observed in the test set, from 53.10% to 56.40%. Similar tendencies are also observed for the other models.

Since the discriminative learning procedure only aims at minimizing the error rate in the training set, the training set performance can usually be tuned very closely to 100% when a large number of parameters are available. However, the performance improvement for the test set is far less than that for the training set, since the statistical variations between the training set and the test set are not taken into consideration in the learning procedure. For investigating robustness issues in more detail, a robust learning procedure and the associated analyses are provided in the following section.

4.2 Robust Learning

As discussed in the previous section, the discriminative learning approach aims at minimizing the training set errors. The error rate measured in the training set is, in general, over-optimistic (Efron and Gong 1983), because the training set performance can be tuned to approach 100% by using a large number of parameters. The parameters obtained in such a way frequently fail to attain an optimal performance when used in

Table 2
Performance with discriminative learning: (a) training set; (b) test set. Values in parentheses correspond to performance excluding unambiguous sentences.

Model	Part-of-Speech Accuracy Rate		Parse Tree	
	in Word (%)	in Sentence (%)	Accuracy Rate (%)	Selection Power
Lex(L1)+Syn(L1)	99.62 (99.59)	95.57 (94.99)	75.43 (72.26)	0.34 (0.26)
+ Discrimination Learning	99.95 (99.94)	99.32 (99.23)	92.04 (91.02)	0.30 (0.21)
Lex(L2)+Syn(L1)	99.64 (99.61)	95.93 (95.41)	75.81 (72.69)	0.34 (0.26)
+ Discrimination Learning	99.97 (99.96)	99.53 (99.47)	92.29 (91.29)	0.30 (0.21)
Lex(L1)+Syn(L2)	99.67 (99.64)	96.07 (95.56)	78.69 (75.93)	0.34 (0.25)
+ Discrimination Learning	99.96 (99.95)	99.40 (99.32)	92.54 (91.58)	0.30 (0.21)
Lex(L2)+Syn(L2)	99.69 (99.67)	96.46 (96.00)	79.04 (76.34)	0.33 (0.25)
+ Discrimination Learning	99.97 (99.97)	99.61 (99.56)	92.77 (91.83)	0.30 (0.21)

(a) Training set performance

Model	Part-of-Speech Accuracy Rate		Parse Tree	
	in Word (%)	in Sentence (%)	Accuracy Rate (%)	Selection Power
Lex(L1)+Syn(L1)	98.89 (98.80)	88.7 (87.1)	49.3 (42.3)	0.45 (0.38)
+ Discrimination Learning	98.82 (98.72)	88.0 (86.3)	55.5 (49.3)	0.42 (0.34)
Lex(L2)+Syn(L1)	98.93 (98.84)	88.9 (87.4)	49.7 (42.7)	0.45 (0.38)
+ Discrimination Learning	99.05 (98.97)	90.1 (88.7)	55.3 (49.1)	0.42 (0.34)
Lex(L1)+Syn(L2)	98.82 (98.71)	88.0 (86.3)	52.8 (46.3)	0.44 (0.37)
+ Discrimination Learning	98.88 (98.78)	88.2 (88.6)	56.6 (50.6)	0.42 (0.34)
Lex(L2)+Syn(L2)	98.89 (98.79)	88.5 (86.9)	53.1 (46.6)	0.44 (0.37)
+ Discrimination Learning	98.92 (98.83)	88.3 (86.7)	56.4 (50.3)	0.42 (0.34)

(b) Test set performance

a real application. This over-tuning phenomenon happens mainly because of the lack of sufficient sampling data and the possible statistical variations between the training set and the test set.

To achieve better performance for a real application, one must deal with statistical variation problems. Most adaptive learning procedures stop adjusting the parameters once the input training token has been classified correctly. For such learning procedures, the distance between the correct candidate and other competitive ones may be too small to cover the possible statistical variations between the training corpus and the real application. To remedy this problem, Su and Lee (1991, 1994) suggested that the distance margin between the correct candidate and the top competitor should be enlarged, even though the input token is correctly recognized, until the margin exceeds a given threshold. A large distance margin would provide a tolerance region in the neighborhood of the decision boundary to allow possible data scattering in the real applications (Su and Lee 1994). A promising result has been observed by applying

this robust learning procedure to recognize the alphabet of E-set in English (Su and Lee 1991, 1994).

To enhance robustness, the learning rules from Equation 24 to Equation 30 are modified as follows. Following the notations in the previous section, the correct syntactic structure is denoted by $Syn_{\alpha,\beta}$, and the syntactic structure of the strongest competitor is denoted by $Syn_{\hat{\jmath},\hat{k}}$, whose score may either rank first or second.

1. For the syntactic and lexical parameters corresponding to the correct candidate:

$$\begin{cases} \lambda_{syn}^{(t+1)}(\alpha,i) = \lambda_{syn}^{(t)}(\alpha,i) + \Delta\lambda_{syn}^{(t)}(\alpha,i), & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta \\ \lambda_{syn}^{(t+1)}(\alpha,i) = \lambda_{syn}^{(t)}(\alpha,i), & \text{otherwise} \end{cases} \tag{31}$$

$$\begin{cases} \lambda_{lex}^{(t+1)}(\beta,i) = \lambda_{lex}^{(t)}(\beta,i) + \Delta\lambda_{lex}^{(t)}(\beta,i), & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta \\ \lambda_{lex}^{(t+1)}(\beta,i) = \lambda_{lex}^{(t)}(\beta,i), & \text{otherwise} \end{cases} \tag{32}$$

2. For the syntactic and lexical parameters corresponding to the strongest competitor:

$$\begin{cases} \lambda_{syn}^{(t+1)}(\hat{\jmath},i) = \lambda_{syn}^{(t)}(\hat{\jmath},i) - \Delta\lambda_{syn}^{(t)}(\hat{\jmath},i), & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta, \\ \lambda_{syn}^{(t+1)}(\hat{\jmath},i) = \lambda_{syn}^{(t)}(\hat{\jmath},i), & \text{otherwise} \end{cases} \tag{33}$$

$$\begin{cases} \lambda_{lex}^{(t+1)}(\hat{k},i) = \lambda_{lex}^{(t)}(\hat{k},i) - \Delta\lambda_{lex}^{(t)}(\hat{k},i), & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta, \\ \lambda_{lex}^{(t+1)}(\hat{k},i) = \lambda_{lex}^{(t)}(\hat{k},i), & \text{otherwise} \end{cases} \tag{34}$$

The learning rules of the syntactic and lexical weights are modified as follows:

$$\begin{cases} w_{syn}^{(t+1)} = w_{syn}^{(t)} + \Delta w_{syn}^{(t)} & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta, \\ w_{syn}^{(t+1)} = w_{syn}^{(t)}, & \text{otherwise} \end{cases} \tag{35}$$

$$\begin{cases} w_{lex}^{(t+1)} = w_{lex}^{(t)} + \Delta w_{lex}^{(t)}, & \text{if } (\|\hat{\Phi}_{\hat{\jmath},\hat{k}}\| - \|\hat{\Phi}_{\alpha,\beta}\|) < \delta, \\ w_{lex}^{(t+1)} = w_{lex}^{(t)}, & \text{otherwise} \end{cases} \tag{36}$$

The margin δ in the above equations can be assigned either absolutely or relatively, as suggested in Su and Lee (1991, 1994). Currently, the relative mode with a 30% passing rate (i.e., 30% of the training tokens pass through the margin) is used in our implementation.

The simulation results, compared with the results obtained by using the discriminative learning procedure, are shown in Table 3. Table 3(a) shows that performances with robust learning in the training set are a little worse than those with discrimination learning for the L1 syntactic language models. Nevertheless, they are a little better for the L2 syntactic language model. All these differences, however, are not statistically significant. On the contrary, the results with robust learning for the *test* set, as shown in Table 3(b), are much better in all cases. The robust learning procedure achieves more than 8% improvement compared with the discriminative learning procedure for all language models. It is evident that the robust learning procedure is superior to the discriminative learning procedure in the test set.

Table 3
Performance with robust learning: (a) training set; (b) test set. Values in parentheses correspond to performance excluding unambiguous sentences.

Model	*Learning Procedure	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	+DL	99.95 (99.94)	99.32 (99.23)	92.04 (91.02)	0.42 (0.34)
	+RL	99.90 (99.89)	98.77 (98.61)	91.84 (90.79)	0.38 (0.29)
Lex(L2)+Syn(L1)	+DL	99.97 (99.96)	99.53 (99.47)	92.29 (91.29)	0.42 (0.34)
	+RL	99.92 (99.92)	99.06 (98.93)	92.08 (91.05)	0.38 (0.30)
Lex(L1)+Syn(L2)	+DL	99.96 (99.95)	99.40 (99.32)	92.54 (91.58)	0.42 (0.34)
	+RL	99.92 (99.92)	99.03 (98.91)	92.94 (92.03)	0.38 (0.30)
Lex(L2)+Syn(L2)	+DL	99.97 (99.97)	99.61 (99.56)	92.77 (91.83)	0.42 (0.34)
	+RL	99.93 (99.93)	99.19 (99.08)	93.12 (92.23)	0.38 (0.30)

(a) Training set performance

Model	*Learning Procedure	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	+DL	98.82 (98.72)	88.0 (86.3)	55.5 (49.3)	0.42 (0.34)
	+RL	99.23 (99.16)	91.5 (90.3)	63.8 (58.8)	0.38 (0.29)
Lex(L2)+Syn(L1)	+DL	99.05 (98.97)	90.1 (88.7)	55.3 (49.1)	0.42 (0.34)
	+RL	99.27 (99.21)	91.5 (90.3)	64.2 (59.2)	0.38 (0.29)
Lex(L1)+Syn(L2)	+DL	98.88 (98.78)	88.2 (88.6)	56.6 (50.6)	0.42 (0.34)
	+RL	99.19 (99.12)	90.9 (89.6)	63.7 (58.7)	0.38 (0.30)
Lex(L2)+Syn(L2)	+DL	98.92 (93.83)	88.3 (86.7)	56.4 (50.3)	0.42 (0.34)
	+RL	99.18 (99.10)	90.7 (89.4)	64.3 (59.3)	0.38 (0.30)

(b) Test set performance

*DL and RL denote "Discriminative Learning" and "Robust Learning," respectively

5. Parameter Smoothing for Sparse Data

The above-mentioned robust learning algorithm starts with the initial parameters estimated by using MLE method. MLE, however, frequently suffers from the large estimation error caused by the lack of sufficient training data in many statistical approaches. For example, MLE gives a zero probability to events that were never observed in the training set. Therefore, MLE fails to provide a reliable result if only a small number of sampling data are available. To overcome this problem, Good (1953) proposed using Turing's formula as an improved estimate over the well-known MLE. In addition, Katz (1987) proposed a different smoothing technique, called the Back-Off procedure, for smoothing unreliably estimated n-gram parameters with their correlated (n-1)-gram parameters. To investigate the effects of parameter smoothing on robust learning, both these techniques are used to smooth the estimated parameters, and then the robust learning procedure is applied based on those smoothed parameters. These two smooth-

ing techniques are first summarized in the following section. The investigation for the smoothing/robust learning hybrid approach is presented next.

5.1 The Smoothing Procedures

5.1.1 Turing's Formula. Let N be the sample size (the number of training tokens) and n_r be the number of events that occur exactly r times. Then the following equation holds:

$$N = \sum_r r \cdot n_r \tag{37}$$

The *maximum likelihood estimate* P_{ML} for the probability of an event e occurring r times is defined as follows:

$$P_{ML}(e) = \frac{r}{N} \tag{38}$$

The estimate based on Turing's formula (Good 1953) is given by the following equation:

$$P_{TU}(e) = \frac{r^*}{N} \tag{39}$$

where

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{40}$$

The total probability estimate, using Turing's formula, for all the events that actually occurred in the sample space is equal to

$$\sum_{e:C(e)>0} P_{TU}(e) = 1 - \frac{n_1}{N} \tag{41}$$

where $C(e)$ stands for the frequency count of the event e in the sample. This, in turn, leads to the following equation:

$$\sum_{e:C(e)=0} P_{TU}(e) = \frac{n_1}{N} \tag{42}$$

According to Turing's formula, the probability mass n_1/N is then equally distributed over the events that never occur in the sample.

5.1.2 Back-off Procedure. Katz (1987) proposed a *back-off* procedure to estimate parameters for an m-gram model, i.e., the conditional probability of a word given the (m-1) preceding words. This procedure is summarized as follows:

$$P_{BF}(w_m \mid w_1^{m-1}) = \begin{cases} P_{TU}(w_m \mid w_1^{m-1}), & \text{if } C(w_1^m) > 0 \\ \alpha(w_2^{m-1}) P_{BF}(w_m \mid w_2^{m-1}) & \text{if } C(w_1^m) = 0, \text{ and } C(w_2^m) > 0, \\ P_{BF}(w_m \mid w_2^{m-1}) & \text{if } \sum_{w_m} C(w_1^m) = 0 \end{cases} \tag{43}$$

where

$$\alpha(w_1^{m-1}) = \frac{1 - \sum_{w_m:C(w_1^m)>0} P_{BF}(w_m \mid w_1^{m-1})}{1 - \sum_{w_m:C(w_1^m)>0} P_{BF}(w_m \mid w_2^{m-1})} \tag{44}$$

is a normalized factor such that

$$\sum_{w_m:C(w_1^m)>0} P_{BF}(w_m \mid w_1^m) + \sum_{w_m:C(w_1^m)=0} P_{BF}(w_m \mid w_1^m) = 1 \quad (45)$$

Compared with Turing's formula, the probability for an m-gram that does not occur in the sample is "backed off" to refer to its corresponding (m-1)-gram probability.

Table 4 gives the experimental results for using the maximum likelihood (ML), Turing (TU) and back-off (BF) estimation procedures. The results show that smoothing the unreliable parameters degrades the training set performance; however, it improves the performance for the test set. Among the estimators, the maximum likelihood estimator provides the best results for the training set, but it is the worst on the test set. Both Turing's and the back-off procedures perform better than the maximum likelihood procedure. This means that smoothing unreliable parameters is absolutely essential if only limited training data are available.

Compared with Turing's procedure, the back-off procedure is $1 \sim 2\%$ worse in all cases. After examining the estimated parameters by using these two smoothing procedures, we found that some syntactic parameters for null events were assigned very large values by the Back-Off procedure, while they were assigned small probabilities by Turing's formula. A typical example is shown as follows. The reduce action "n quan → NLM*" given the left contexts [P*, N2] never occurred in the training set. But, the probability of P(n quan → NLM* | [n quan] reduced; L2=P*, L1=N2) is finally replaced by the probability of P(n quan → NLM* | [n quan] reduced) in the Back-Off estimation procedure. Since the probability P(n quan → NLM* | [n quan] reduced) has a large value (= 0.25), the probability P(n quan → NLM* | [n quan] are reduced; L2=P*, L1=N2) is accordingly large also. From the estimation point of view, the parameters for null events may be assigned better estimated values by using the Back-Off method; however, these parameters do not necessarily guarantee that the discrimination power will be better improved. Take the sentence *"A stack of pinfeed paper three inches high may be placed underneath it"* as an example. The decomposed phrase levels and the corresponding syntactic scores for the correct and the top candidate are shown in Table 5 (a) and (b), respectively. We find that the main factor affecting the tree selection is the sixth phrase level, which corresponds to the reduce action "n quan → NLM*" with the left two contextual symbols P* and N2 for the top candidate. As described above, the probability P(n quan → NLM* | [n quan] reduced; L2=P*, L1=N2) is assigned a large value in the Back-Off estimation procedure. However, to correctly select the right syntactic structure in this example, P(quan → QUAN | [quan] reduced; L2=P*, L1=N2) should be greater than P(n quan → NLM* | [n quan] reduced; L2=P*, L1=N2). This requirement may not be met by any estimation procedure, since the above two probabilities are estimated from two different outcome spaces (one conditioned on [quan], and the other conditioned on [n, quan]). Therefore, even though the Back-Off procedure may give better estimates for the parameters, it cannot guarantee that the recognition result can be improved. The comparison between Turing's procedure and the Back-Off procedure thus varies in different cases. In fact, the Back-Off estimation did show better results in our previous research (Lin, Chiang, and Su 1994). Nevertheless, we will show in the next section that the selection of a smoothing method is not crucial after the robust learning procedure has been applied.

Furthermore, comparing the results in Table 3 and Table 4, we find that the performance with the robust learning procedure is much better than that with the smoothing techniques. Although both the adaptive learning procedures and the smoothing techniques show improvement, the robust learning procedure, which emphasizes dis-

Table 4
Performance for lexical and syntactic disambiguation with various estimators. Values in parentheses correspond to performance excluding unambiguous sentences. (ML: Maximum Likelihood estimator; TU: TUring's formula; BF: Back-Off technique.)

Model	Estimation Method	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	ML	99.62 (99.59)	95.57 (94.99)	75.43 (72.26)	0.34 (0.26)
	TU	99.43 (99.38)	93.47 (92.62)	69.63 (65.71)	0.36 (0.28)
	BF	99.40 (99.35)	93.18 (92.30)	67.33 (63.11)	0.37 (0.28)
Lex(L2)+Syn(L1)	ML	99.64 (99.61)	95.93 (95.41)	75.81 (72.69)	0.34 (0.26)
	TU	99.48 (99.44)	94.09 (93.32)	70.12 (66.26)	0.36 (0.28)
	BF	99.45 (99.41)	93.81 (93.01)	67.86 (63.70)	0.37 (0.28)
Lex(L1)+Syn(L2)	ML	99.67 (99.64)	96.07 (95.56)	78.69 (75.93)	0.34 (0.25)
	TU	99.45 (99.40)	93.79 (92.99)	72.13 (68.53)	0.35 (0.27)
	BF	99.39 (99.33)	93.03 (92.13)	67.48 (63.27)	0.36 (0.28)
Lex(L2)+Syn(L2)	ML	99.69 (99.67)	96.46 (96.00)	79.04 (76.34)	0.33 (0.25)
	TU	99.49 (99.45)	94.22 (93.48)	72.48 (68.92)	0.35 (0.27)
	BF	99.44 (99.39)	93.56 (92.72)	67.87 (63.71)	0.36 (0.28)

(a) Training set performance

Model	Estimation Method	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	ML	98.89 (98.80)	88.7 (87.1)	49.3 (42.3)	0.45 (0.38)
	TU	99.03 (98.95)	89.5 (88.0)	53.9 (47.5)	0.43 (0.36)
	BF	99.01 (98.92)	88.9 (87.4)	52.4 (45.8)	0.44 (0.36)
Lex(L2)+Syn(L1)	ML	98.93 (98.84)	88.9 (87.4)	49.7 (42.7)	0.45 (0.38)
	TU	99.08 (99.00)	90.0 (88.6)	54.3 (48.0)	0.43 (0.35)
	BF	99.09 (99.01)	90.1 (88.7)	53.2 (46.7)	0.44 (0.36)
Lex(L1)+Syn(L2)	ML	98.82 (98.71)	88.0 (86.3)	52.8 (46.2)	0.44 (0.37)
	TU	98.98 (98.89)	89.1 (87.6)	56.5 (50.5)	0.42 (0.34)
	BF	99.02 (98.94)	89.1 (87.6)	54.4 (48.1)	0.43 (0.35)
Lex(L2)+Syn(L2)	ML	98.89 (98.79)	88.5 (86.9)	53.1 (46.6)	0.44 (0.37)
	TU	99.05 (98.97)	89.7 (88.3)	56.6 (50.6)	0.42 (0.34)
	BF	99.10 (99.02)	90.1 (88.7)	55.1 (48.9)	0.43 (0.35)

(b) Test set performance

crimination capability rather than merely improving estimation process, achieves a better result. Since the philosophies of performance improvement for these two algorithms are different (one from the *estimation* point of view and the other from the *discrimination* point of view), it is interesting to combine these two algorithms and investigate the effect of the robust learning procedure on the smoothed parameters. Detailed discussion on this hybrid approach is addressed in the following section.

Table 5
The decomposed phrase levels associated with the sentence *"A stack of pinfeed paper three inches high may be placed underneath it,"* and the corresponding scores with the Back-Off estimation method for (a) the correct candidate and (b) the top candidate. The shaded rows indicate the different patterns between the two parse trees.

	word	L2	L1	current symbols → reduced symbol	score
1	*A*	$	$	art → DET	-0.0030
2	*stack*	$	$	DET n → N2	-0.5483
3	*of*	$	N2	p → P*	-0.0072
4	*pinfeed*	N2	P*	n → N1	-0.4221
5	*paper*	N2	P*	N1 n → N2	-0.3611
6	*three*	P*	N2	quan → QUAN	-2.7226
7	*inches*	N2	QUAN	n → n	-1.8434
8	*high*	$	$	N2 P* N2 QUAN n a → N3	-1.7924
9	*may*	$	N3	mod1 → mod 1	-0.5172
10	*be*	$	N3	mod1 be → AUX	-0.0048
11	*placed*	N3	AUX	v → V1	-0.4007
12	*underneath*	AUX	V1	p → P*	-0.0044
13	*it*	$	$	N3 AUX V1 P* pron → S2	-1.7924

(a) Correct Candidate

	word	L2	L1	current symbols → reduced symbol	score
1	*A*	$	$	art → DET	-0.0030
2	*stack*	$	$	DET n → N2	-0.5483
3	*of*	$	N2	p → P*	-0.0072
4	*pinfeed*	N2	P*	n → N2	-0.3658
5	*paper*	P*	N2	n → n	-0.3528
6	*three*	P*	N2	n quan → NLM*	-0.6049
7	*inches*	P*	N2	NLM* n → N2	-1.2297
8	*high*	P*	N2	N2 a → N3	-1.1606
9	*may*	N2	N3	mod1 → AUX	-0.0199
10	*be*	$	$	N2 P* N2 N3 AUX v → N3	-1.1606
11	*placed*	$	N3	v → V1	-1.1324
12	*underneath*	N3	V1	p → P*	-0.0084
13	*it*	$	$	N3 V1 P* pron → S2	-0.3500

(b) Top Candidate

5.2 Robust Learning on the Smoothed Parameters

The hybrid approach first uses a smoothing technique to estimate the initial parameters. Afterwards, the robust learning procedure is applied based on the smoothed parameters. The advantages of this approach are two-fold. First, the power of the scoring function is enhanced since the smoothing techniques can reduce the estimation errors, especially for unseen events. Second, the parameters estimated from the smoothing techniques give the robust learning procedure a better initial point and are more likely to reach a better solution when many local optima exist in the parameter space. In other words, the smoothing techniques indirectly prevent the learning process from being trapped in a poor local optimum, although reducing the estimation

Table 6
Performance with the smoothing/robust learning hybrid approach. Values in parentheses correspond to performance excluding unambiguous sentences. (ML+RL: Maximum Likelihood estimator/Robust Learning; TU+RL: TUring's formula/Robust Learning; BF+RL: Back-Off technique/Robust Learning.)

Model	Estimation/ Learning	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	ML+RL	99.90 (99.89)	98.77 (98.61)	91.84 (90.79)	0.31 (0.22)
	TU+RL	99.88 (99.87)	98.59 (98.41)	90.89 (89.71)	0.31 (0.22)
	BF+RL	99.88 (99.87)	98.57 (98.38)	90.76 (89.56)	0.31 (0.22)
Lex(L2)+Syn(L1)	ML+RL	99.92 (99.92)	99.06 (98.93)	92.08 (91.05)	0.31 (0.22)
	TU+RL	99.90 (99.89)	98.82 (98.67)	91.20 (90.06)	0.31 (0.22)
	BF+RL	99.89 (99.89)	98.76 (98.59)	90.93 (89.76)	0.31 (0.22)
Lex(L1)+Syn(L2)	ML+RL	99.92 (99.92)	99.03 (98.91)	92.94 (92.03)	0.30 (0.21)
	TU+RL	99.90 (99.90)	98.89 (98.71)	91.72 (90.65)	0.31 (0.22)
	BF+RL	99.89 (99.88)	98.74 (98.58)	91.18 (90.04)	0.31 (0.22)
Lex(L2)+Syn(L2)	ML+RL	99.93 (99.93)	99.19 (99.08)	93.12 (92.23)	0.30 (0.21)
	TU+RL	99.91 (99.90)	98.92 (98.78)	91.79 (90.73)	0.31 (0.22)
	BF+RL	99.90 (99.89)	98.90 (98.76)	91.40 (90.29)	0.31 (0.22)

(a) Training set performance

Model	Estimation/ Learning	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	ML+RL	99.23 (99.16)	91.5 (90.3)	63.8 (58.8)	0.38 (0.30)
	TU+RL	99.37 (99.32)	92.3 (91.7)	67.1 (62.5)	0.37 (0.28)
	BF+RL	99.34 (99.28)	92.3 (91.2)	67.1 (62.5)	0.37 (0.28)
Lex(L2)+Syn(L1)	ML+RL	99.27 (99.21)	91.5 (90.3)	64.2 (59.2)	0.38 (0.29)
	TU+RL	99.39 (99.33)	92.8 (91.8)	68.0 (63.6)	0.37 (0.28)
	BF+RL	99.36 (99.30)	92.5 (91.5)	67.9 (63.4)	0.37 (0.28)
Lex(L1)+Syn(L2)	ML+RL	99.19 (99.12)	90.9 (89.6)	63.7 (58.7)	0.38 (0.30)
	TU+RL	99.38 (99.32)	92.9 (91.9)	69.3 (65.0)	0.37 (0.28)
	BF+RL	99.37 (99.32)	92.8 (91.8)	69.1 (64.8)	0.37 (0.28)
Lex(L2)+Syn(L2)	ML+RL	99.18 (99.10)	90.7 (89.4)	64.3 (59.3)	0.38 (0.30)
	TU+RL	99.45 (99.40)	93.7 (92.8)	69.8 (65.6)	0.37 (0.28)
	BF+RL	99.39 (99.34)	93.3 (92.4)	69.2 (64.9)	0.37 (0.28)

(b) Test set performance

errors by using these methods does not directly improve the discrimination capability. Experimental results using this hybrid approach are shown in Table 6, where the results using the (ML+RL) mode are also listed for reference.

Significant improvement, compared with the (ML+RL) mode, has been observed

by using the smoothed parameters at the initial step before the robust learning procedure is applied. With this hybrid approach, better results are obtained using a more complex language model, such as Lex(L2)+Syn(L2). However, there is no significant performance difference achieved by using the (TU+RL) and the (BF+RL) approaches for all language models, even though Turing's smoothing formula was shown to behave better than the Back-Off procedure before applying the robust learning procedure. This is not surprising because starting the robust learning procedure with different initial points would still lead to the same local optimum if the starting region, where the initial points are located, has only one local optimum. By using Turing's formula/Robust Learning hybrid approach for the Lex(L2)+Syn(L2) model, the accuracy rate for parse tree selection is improved to 69.2%, which corresponds to a 34.3% error reduction compared with the baseline of 53.1% accuracy. The superiority in terms of both discrimination and robustness for the hybrid approach is thus clearly demonstrated.

6. Parameter Tying

The investigation described in Section 5 has shown that smoothing is essential before the robust learning procedure is applied. Nevertheless, although we get better initial estimates by smoothing parameters corresponding to rare events, these parameters still cannot be trained well in the robust learning procedure, because such parameters are seldom or never touched by the training process. Unfortunately, this problem occurs frequently in statistical language modeling. This happens because, in general, to reduce modeling errors, a model accounting for more contextual information is desired. However, a model incorporating more contextual information would have a larger number of *null event* parameters, which will not be touched in the learning procedure.

To overcome this problem, a novel approach is proposed in this paper to train the null event parameters by tying them to their highly correlated parameters, and then adjusting them through the robust learning procedure. Basically, the reasons for using this approach are two-fold. First, the number of parameters can be reduced by using the tying scheme. Secondly, this tying scheme gives parameters for rare events more chance to be touched in the learning procedure and thus they can be trained more reliably. The details are addressed below.

6.1 Tying Procedure

The tying procedure includes the following two steps:

1. **Initial Estimation**: For an m-gram model, the conditional probability $P(x_m \mid x_1^{m-1})$ is estimated by the following equation:

$$P(x_m \mid x_1^{m-1}) = \frac{C(x_1, \ldots, x_{m-1}, x_m)}{\sum_{y \in V} C(x_1, \ldots, x_{m-1}, y)}, \tag{46}$$

where V denotes the vocabulary and $C(\cdot)$ stands for the frequency count of an event in the training set. If $\sum_{y \in V} C(x_1, \ldots, x_{m-1}, y) \geq Q_d$, where Q_d is a present threshold, it is assumed that the estimated value of $P(x_m \mid x_1^{m-1})$ is reliable and no action is required in this situation. On the other hand, if $\sum_{y \in V} C(x_1, \ldots, x_{m-1}, y) < Q_d$, the estimated value of $P(x_m \mid x_1^{m-1})$ is regarded as unreliable. In this case, $P(x_m \mid x_1^{m-1})$ is substituted by the smoothed value of the $(m-1)$-gram probability

Table 7
The number of parameters before and after the tying process. Note that the parameters of P(W | C) are not tied.

Model		Number of Lexical Parameter		Number of Syntactic Parameter	Number of Total Parameters
		P(W|C)	P(Ci|Cj)		
Lex(L1)+Syn(L1)	Before Tying	304,630	1225	96699*95	9,492,260
	After Tying	304,630	760	98,195	403,595
	Ratio	1.0	0.62	0.0106	0.0425
Lex(L2)+Syn(L1)	Before Tying	304,630	42875	96699*95	9,533,910
	After Tying	304,630	4,199	98,195	407,024
	Ratio	1.0	0.098	0.0106	0.0427
Lex(L1)+Syn(L2)	Before Tying	304,630	1225	96699*(95*95)	873,014,330
	After Tying	304,630	760	112,114	417,504
	Ratio	1.0	0.62	0.00013	0.000478
Lex(L2)+Syn(L2)	Before Tying	304,630	42875	96699*(95*95)	873,055,980
	After Tying	304,630	4,199	112,114	420,943
	Ratio	1.0	0.098	0.00013	0.000482

$P(x_m \mid x_2^{m-1})$. Currently, Q_d is set to ten times the size of the possible outcomes of x_m, i.e., $Q_d = [10 \times (\textit{the number of possible tags})]$ for the part-of-speech transition parameters.

2. **Tying Procedure**: Consider the m-gram events $\{x_1, \ldots, x_{m-1}, y_i\}$, $\forall y_i \in V$, which have the same (m-1)-gram history $\{x_1, \ldots, x_{m-1}\}$. Each of the probabilities $P(y_i \mid x_1, \ldots, x_{m-1})$, $\forall y_i \in V$ is first assigned a smoothed value in the above step. To give these parameters more chance to be trained during the robust learning process, we tie together the parameters whose corresponding events appear less than Q_n times in the training set. That is, the parameters $P(y_k \mid x_1, x_2, \ldots, x_{m-1})$, $y_k \in V$, are tied if the associated events satisfy the following conditions:

$$\sum_{y_i \in V} C(x_1, \ldots, x_{m-1}, y_i) < Q_d, \textit{ and } C(x_1, \ldots, x_{m-1}, y_k) < Q_n, \; y_k \in V, \quad (47)$$

where Q_n is currently set to 2.

The numbers of parameters before and after tying for each language model are tabulated in Table 7. This table shows that the number of parameters is greatly reduced after the tying process, especially for the L2 syntactic models.

6.2 Robust Learning on the Tied Parameters

After the parameters are estimated and tied through the tying procedure, the robust learning algorithm is applied on the tied parameters. The experimental results are shown in Table 8. The results with the TU+RL hybrid approach are also listed for reference. The performance with the Tying/Robust Learning hybrid approach, as shown in Table 8, deteriorates somewhat in the training set because the tying procedure decreases the modeling resolution. However, the test set performance with this hybrid approach is slightly (but not significantly) better than the Turing's formula/Robust

Table 8
Performance of different language models with the various hybrid approaches. Values in parentheses correspond to performance excluding unambiguous sentences. TY+RL: TYing parameters/Robust Learning.

Model	Estimation/ Learning	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	TU+RL	99.88 (99.87)	98.59 (98.41)	90.89 (89.71)	0.31 (0.22)
	TY+RL	99.86 (99.85)	98.33 (98.12)	89.81 (88.49)	0.31 (0.23)
Lex(L2)+Syn(L1)	TU+RL	99.90 (99.89)	98.82 (98.67)	91.20 (90.06)	0.31 (0.22)
	TY+RL	99.87 (99.86)	98.48 (98.28)	89.89 (88.58)	0.31 (0.23)
Lex(L1)+Syn(L2)	TU+RL	99.90 (99.90)	98.89 (98.71)	91.72 (90.65)	0.30 (0.21)
	TY+RL	99.88 (99.87)	98.60 (98.42)	90.61 (89.40)	0.31 (0.22)
Lex(L2)+Syn(L2)	TU+RL	99.91 (99.90)	98.92 (98.78)	91.79 (90.73)	0.30 (0.21)
	TY+RL	99.89 (99.88)	98.80 (98.64)	90.71 (89.51)	0.31 (0.22)

(a) Training set performance

Model	Estimation/ Learning	Part-of-Speech Accuracy Rate in Word (%)	Part-of-Speech Accuracy Rate in Sentence (%)	Parse Tree Accuracy Rate (%)	Parse Tree Selection Power
Lex(L1)+Syn(L1)	TU+RL	99.37 (99.32)	92.7 (91.7)	67.1 (62.5)	0.37 (0.28)
	TY+RL	99.36 (99.31)	92.8 (91.8)	67.5 (63.0)	0.37 (0.28)
Lex(L2)+Syn(L1)	TU+RL	99.39 (99.33)	92.8 (91.8)	68.0 (63.6)	0.37 (0.28)
	TY+RL	99.39 (99.33)	92.9 (91.9)	68.3 (63.9)	0.37 (0.28)
Lex(L1)+Syn(L2)	TU+RL	99.38 (99.32)	92.9 (91.9)	69.3 (65.0)	0.37 (0.28)
	TY+RL	99.39 (99.33)	92.9 (91.9)	69.4 (65.2)	0.36 (0.28)
Lex(L2)+Syn(L2)	TU+RL	99.45 (99.40)	93.7 (92.8)	69.8 (65.6)	0.37 (0.28)
	TY+RL	99.43 (99.38)	93.5 (92.6)	70.3 (66.2)	0.36 (0.27)

(b) Test set performance

Learning approach. In addition, it reduces the large number of parameters, and thus greatly eases the memory constraints for implementing the system.

A summary illustrating the performance improvement by using the proposed enhancement mechanisms for the Lex(L2)+Syn(L2) model is shown in Table 9. The proposed tying approach, after being combined with the robust learning procedure, significantly reduces the error rate compared with the baseline (36.67% error reduction is achieved, from 53.1% to 70.3%). Moreover, the number of parameters is reduced to less than 1/2000 of the original parameter space.

7. Conclusions and Future Work

An integrated scoring function capable of incorporating various knowledge sources to resolve syntactic ambiguity problems is explored in this paper. In the baseline model, the parameters are estimated by using the maximum likelihood method. The MLE

Table 9
Summary of performance for the Lex(L2)+Syn(L2) model using various performance enhancement methods. Values in parentheses correspond to performance excluding unambiguous sentences.

	Testing Set Performance			
	Part-of-Speech Accuracy Rate		Parse Tree	
Model: Lex(L2)+Syn(L2)	in Word (%)	in Sentence (%)	Accuracy Rate (%)	Selection Power
Baseline	98.89 (98.79)	88.50 (86.90)	53.1 (46.6)	0.44 (0.37)
+ Robust Learning	99.18 (99.10)	90.70 (89.41)	64.3 (59.3)	0.38 (0.30)
+ TU Smoothing	99.05 (98.97)	89.70 (88.27)	56.6 (50.6)	0.42 (0.34)
+ TU Smoothing + Robust Learning	99.45 (99.40)	93.70 (92.82)	69.8 (65.6)	0.37 (0.28)
+ Tying parameter + Robust Learning	99.43 (99.38)	93.50 (92.60)	70.3 (66.2)	0.36 (0.27)

approach fails to achieve satisfactory performance because the discrimination and robustness issues are not considered in the estimation process. To improve performance, a discrimination- and robustness-oriented method is adopted to directly pursue the correct ranking orders of possible alternative syntactic structures. In addition, this learning procedure is able to resolve problems resulting from statistical variations between the training corpus and real tasks.

The effects of parameter smoothing for null events with Turing's formula and the Back-Off method are investigated in this paper. A better initial estimate of the parameters makes the robust learning procedure achieve better performance when many local optima exist in the parameter space. Significant improvement of 34.3% error reduction rate is attained when we apply the robust learning procedure on the smoothed parameters.

Finally, a parameter tying scheme for rare events is proposed so that the unreliably estimated parameters are tied and trained together through the robust learning procedure. Thus, this approach makes it possible to tune all the parameters through the learning process. In addition, the number of parameters is significantly reduced with the tying process. The reduction of the number of parameters is over 99% for each language model. Moreover, the accuracy rate of 70.3% for parse tree selection, or 36.7% error reduction rate, is obtained by using this novel approach.

To explore the areas for further improving the system, the remaining errors have been examined. It was found that a very large portion of errors result from attachment problems, including prepositional phrase (PP) attachment and modification scope for adverbial phrases, adjective phrases, and relative clauses, while less than 10% of the errors arise because of incorrect part-of-speech tagging. To further improve the lexical scoring module, some refinement mechanisms developed for our part-of-speech tagger (Lin, Chiang, and Su 1994) will be incorporated into this system. As for the attachment problems, we found that the system appears to have a preference for local attachment, which is not always inappropriate. The current model fails to deal with such problems because only syntactic information from two left contextual nonterminal symbols is consulted for computation. To resolve the attachment problems, integrating seman-

tic information, such as word sense collocations, would be required. In addition, to enable the system to take into account information associated with long-distance dependency, we plan to modify the syntactic model so that it can evaluate *structural dependency* across various subtrees in the parse history. A large number of parameters will inevitably be required for such a formulation, and a large training corpus is thus needed for training. A bootstrapping procedure for parameter estimation with respect to a very large corpus, therefore, will be applied in future research.

Acknowledgments

This research is supported by the R.O.C. National Science Council under NSC 82-0408-E-007-059 project. We would like to thank the Behavior Design Corporation (BDC) for providing us with the parsed corpus. Jing-Shin Chang has given valuable suggestions for writing this paper, in particular for the comparison with Briscoe and Carroll's approach. Also, four anonymous reviewers' comments on earlier drafts were very helpful to us in preparing the final version.

References

Amari, Shunichi (1967). "A theory of adaptive pattern classifiers." *IEEE Trans. on Electronic Computers* EC-16, 299–307.

Bahl, Lalit R.; Brown, Peter F.; deSouza, Peter V.; and Mercer, Robert L. (1988). "A new algorithm for the estimation of hidden Markov model parameters." In *Proceedings, IEEE 1988 International Conference on Acoustics, Speech, and Signal Processing*. New York, 493–496.

Bahl, Lalit R.; Jelinek, Frederick; and Mercer, Robert (1983). "A maximum likelihood approach to continuous speech recognition." *IEEE Trans. on Pattern Analysis and Machine Intelligence* PAMI-5(2), 179–190.

Briscoe, Ted, and Carroll, John (1993). "Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars." *Computational Linguistics*, 19(1), 25–59.

Brown, Peter F.; Della Pietra, Vincert J.; deSouza, Peter V.; Lai, Jenifer C.; and Mercer, Robert L. (1992). "Class-based n-gram models of natural language." *Computational Linguistics*, 18(4), 467–479.

Chang, Jing-Shin; Luo, Yi-Fen; and Su, Keh-Yih (1992). "GPSM: A generalized probabilistic semantic model for ambiguity resolution." In *Proceedings, 30th Annual Meeting of the Association for Computational Linguistics*. University of Delaware, Newark, 177–184.

Chen, Shu-Chuan; Chang, Jing-Shin; Wang, Jong-Nae; and Su, Keh-Yih (1991). "ArchTran: A corpus-based statistics-oriented English–Chinese machine translation system." In *Proceedings, Machine Translation Summit III*. Washington, D.C., 33–40.

Chiang, Tung-Hui; Lin, Yi-Chung; and Su, Keh-Yih (1992). "Syntactic ambiguity resolution using a discrimination and robustness oriented adaptive learning algorithm." In *Proceedings, Fifteenth International Conference on Computational Linguistics*, Nantes, 352–358.

Church, Kenneth (1989). "A stochastic parts program and noun phrase for unrestricted text." In *Proceedings, IEEE 1989 International Conference on Acoustics, Speech, and Signal Processing*. Glasgow, 695–698.

Efron, Bradley, and Gong, Gail (1983). "A leisurely look at the bootstrap, the jackknife, and cross-validation." *The American Statistics*, 37(1), 36–48.

Garside, Roger; Leech, Geoffrey; and Sampson, Geoffrey (1987). *The Computational Analysis of English: A corpus-based approach*. Longman.

Gillick, L. and Cox, S. J. (1989). "Some statistical issues in the comparison of speech recognition algorithm." In *Proceedings, IEEE 1989 International Conference on Acoustics, Speech, and Signal Processing*. Glasgow, 532–535.

Good, I. J. (1953). "The population frequencies of species and the estimation of population parameters." *Biometrika*, 40, 237–264.

Hopcroft, John E., and Ullman, Jeffrey D. (1974). *Formal Languages and Their Relation to Automata*. Addison-Wesley.

Katagiri, Shigeru; Lee, Chin-Hui and Juang, Biing-Hwang (1991). "New discriminative training algorithm based on the generalized probabilistic descent method." In *Proceedings, 1991 IEEE Workshop Neural Networks for Signal Processing*. Piscataway, New Jersey, 299–308.

Katz, Slava M. (1987). "Estimation of probabilities from sparse data for the language model component of a speech recognizer." *IEEE Transactions on Acoustics, Speech and Signal Processing*. ASSP-35,

400–401.

Kendall, Maurice, and Stuart, Alan (1979). *The Advanced Theory of Statistics.* Macmillan.

Lin, Yi-Chung; Chiang, Tung-Hui; and Su, Keh-Yih (1994). "Automatic model refinement—with an application to tagging." In *Proceedings, 15th International Conference on Computational Linguistics.* Kyoto, 148–153.

Merialdo, Bernard (1991). "Tagging text with a probabilistic model." In *Proceedings, the IEEE 1991 International Conference on Acoustic, Speech, and Signal Processing.* Toronto, 809–812.

Su, Keh-Yih, and Chang, Jing-Shin (1988). "Semantic and syntactic aspects of score function." In *Proceedings, 12th International Conference on Computational Linguistics.* Budapest, 22–27.

Su, Keh-Yih, and Chang, Jing-Shin (1990). "Some key issues in designing MT systems." *Machine Translation,* 5(4), 265–300.

Su, Keh-Yih, and Lee, Chin-Hui (1991). "Robustness and discrimination oriented speech recognition using weighted HMM and subspace projection approaches." In *Proceedings, IEEE 1991 International Conference on Acoustic, Speech, and Signal Processing.* Toronto, 541–544.

Su, Keh-Yih, and Lee, Chin-Hui (1994). "Speech recognition using weighted HMM and subspace projection approaches." *IEEE Trans. on Speech and Audio Processing,* 2(1), 69–79.

Su, Keh-Yih; Chang, Jing-Shin; and Lin, Yi-Chung (1992). "A discriminative approach for ambiguity resolution based on a semantic score function." In *Proceedings, 1992 International Conference on Spoken Language Processing.* Banff, 149–152.

Su, Keh-Yih; Chiang, Tung-Hui; and Lin, Yi-Chung (1991). "A robustness and discrimination oriented score function for integrating speech and language processing." In *Proceedings, 2nd European Conference on Speech Communication and Technology.* Genova, 207–210.

Su, Keh-Yih; Wang, Jong-Nae; Su, Mei-Hui; and Chang, Jing-Shin (1991). "GLR parsing with scoring." In *Generalized LR Parsing,* edited by Masaru Tomita, 93–112. Kluwer Academic Publisher.

Su, Keh-Yih; Wang, Jong-Nae; Su, Mei-Hui; and Chang, Jing-Shin (1989). "A sequential truncation parsing algorithm based on the score function." In *Proceedings of 1989 International Workshop on Parsing Technologies (IWPT-89).* Pittsburgh, 95–104.

Wright, J. H., and Wrigley, E. N. (1991). "GLR parsing with probability." In *Generalized LR Parsing,* edited by Masaru Tomita, 113–128. Kluwer Academic Publisher.