Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm

Tung-Hui Chiang, Yi-Chung Lin and Keh-Yih Su

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
E-Mail: thchiang@ee.nthu.edu.tw

Topic Area: Computational Methods (Statistical), Application (NLP)

Abstract

In this paper, a discrimination and robustness oriented adaptive learning procedure is proposed to deal with the task of syntactic ambiguity resolution. Owing to the problem of insufficient training data and approximation error introduced by the language model, traditional statistical approaches, which resolve ambiguities by indirectly and implicitly using maximum likelihood method, fail to achieve high performance in real applications. The proposed method remedies these problems by adjusting the parameters to maximize the accuracy rate directly. To make the proposed algorithm robust, the possible variations between the training corpus and the real tasks are also taken into consideration by enlarging the separation margin between the correct candidate and its competing members. Significant improvement has been observed in the test. The accuracy rate of syntactic disambiguation is raised from 46.0% to 60.62% by using this novel approach.

1. Introduction

Ambiguity resolution has long been the focus in natural language processing. Many rule-based approaches have been proposed in the past. However, when applying such approaches to large scale applications, they usually fail to offer satisfactory performance. As a huge amount of fine-grained knowledge is required to solve the ambiguity problem, it is quite difficult for rule-based approach to acquire the huge and fine-grained knowledge, and maintain consistency among them by human [Su 90a].

Probabilistic approaches attack these problems by providing a more *objective* measure on the preference to a given interpretation. Then, these approaches acquire huge and fine grained knowledge, or parameters in statistic terms from the corpus automatically. The *uncertainty* problem in linguistic phenomena is resolved on a more solid basis if a probabilistic approach is adopted. Moreover, the knowledge acquired by the statistical method is always *consistent* because the knowledge is acquired by jointly considering all the data in the corpus at the same time. Hence, the time for knowledge acquisition and the cost to maintain consistency are significantly reduced by adopting those probabilistic approaches.

To resolve the problems resulting from syntactic ambiguities, a unified statistical approach for ambiguity resolution has been proposed by Su [Su 88, 92b]. In that approach, all knowledge sources, including lexical, syntactic and semantic knowledge, are encoded by a unified *probabilistic score function* with a uniform formulation. This uniform probabilistic score function has been successfully applied in spoken language processing [Su 90b, 91b, 92a] and machine translation systems [Chen 91] to integrate different knowledge sources for ambiguity resolution.

In implementing this unified probabilistic score function, values of score functions are estimated from the data in the training corpus. However, due to the problem of insufficiency of training data and incompleteness of model knowledge, the statistical variations between the training corpus and the real application are usually not covered by this approach. Therefore, the performance in the testing set sometimes gets poor in the real application.

To enhance the capability of discrimination and robustness of those proposed score function, a discrimination-oriented adaptive learning is proposed in this paper. And then, the robustness of this proposed adaptive learning procedure is enhanced by enlarging the margin between the correct candidate and its confusing candidates to achieve maximum separation between different candidates.

Since the implementation of this adaptive learning procedure is based on the uniform probabilistic score function, we will first briefly review the unified probabilistic score function. Readers who are

interested in the details about the uniform probabilistic score function please refer [Chen 91, Su 91b, 92a, 92b].

2. Overview of Uniform Probabilistic Score Function

2.1. General Definition

A *Score Function* for a given syntactic tree, say Syn_j, is defined as follows:

$$Score\,(Syn_j) \equiv P(Syn_j, Lex_j \mid w_1^n), \tag{1}$$

where w_1^n is the input word sequence, $w_1^n = \{w_1, w_2, \cdots, w_n\}$, and Lex_j, the corresponding lexical string, i.e., part of speech sequence $\{c_{j_1}, c_{j_2}, \cdots, c_{j_n}\}$. By applying the *multiplication theorem* of probability, $P(Syn_j, Lex_j \mid w_1^n)$ can be restated as follows.

$$\begin{aligned} &P(Syn_j, Lex_j \mid w_1^n) \\ &= P(Syn_j \mid Lex_j, w_1^n) \times P(Lex_j \mid w_1^n) \\ &= S_{syn}(Syn_j) \times S_{lex}(Lex_j). \end{aligned} \tag{2}$$

The two components, $S_{syn}(Syn_j)$ and $S_{lex}(Lex_j)$, in the above formula are called *syntactic Score Function* and *Lexical Score Function*, respectively. The original score function, i.e., $P(Syn_j, Lex_j|w_1^n)$, is then called *Integrated Score Function*.

Next, we assume the information, from the word sequence w_1^n, required for syntactic ambiguity resolution, has percolated to the lexical interpretation Lex_j. Also, only little additional information can be provided from w_1^n for the task of disambiguating syntactic interpretation Syn_j after the lexical interpretation Lex_j is given. Thus, the syntactic score can be approximated as shown in Eq.(3):

$$S_{syn}(Syn_j) = P(Syn_j \mid Lex_j, w_1^n) \approx P(Syn_j \mid Lex_j). \tag{3}$$

The integrated score function $P(Syn_j, Lex_j \mid w_1^n)$ is then approximated as follows.

$$\begin{aligned} &P(Syn_j, Lex_j \mid w_1^n) \\ &\approx P(Syn_j \mid Lex_j) \times P(Lex_j \mid w_1^n). \end{aligned} \tag{4}$$

Such a formulation allows us to use both *lexical and syntactic* knowledge in assigning *preference* measure to a syntactic tree. In the real computation, log operation is used to convert the operations of multiplication to the operations of addition. The following equation shows the final form in the real application.

$$\log P(Syn_j, Lex_j \mid w_1^n) = \log S_{syn}(Syn_j) + \log S_{lex}(Lex_j). \tag{5}$$

2.2. Lexical Score Function

Let $c_{k_1^n}$ denote the k-th sequence of the lexical category, or part of speech, corresponding to the word sequence w_1^n. The *Lexical Score Function* can be expressed as follows [Chen 91, Su 92b]:

$$\begin{aligned} S_{lex}(Lex_k) &= P(Lex_k \mid w_1^n) = P(c_{k_1^n} \mid w_1^n) \\ &= \prod_{i=1}^{n} P\left(c_{k_i} \mid c_{k_1^{i-1}}, w_1^n\right), \end{aligned} \tag{6}$$

where c_{k_i} is the lexical category of w_i. Several forms [Gars 87, Chur 88, Su 92b] for $P\left(c_{k_i} \mid c_{k_1^{i-1}}, w_1^n\right)$ were proposed to simplify the computation. For example, [Chur 88] approximated $P\left(c_{k_i} \mid c_{k_1^{i-1}}, w_1^n\right)$ by $\left[P\left(c_{k_i} \mid c_{k_1^{i-1}}\right) \times P(c_{k_i} \mid w_i)\right]$. A general nonlinear smoothing form [Chen 91] described in Eq.(7) is adopted in this paper:

$$\begin{aligned} &g\left(P\left(c_{k_i} \mid c_{k_{i-1}}, w_i\right)\right) \\ &\approx \lambda g\left(P\left(c_{k_i} \mid w_i\right)\right) + (1-\lambda)\, g\left(c_{k_i} \mid c_{k_{i-1}}\right), \end{aligned} \tag{7}$$

where λ is the lexical weight ($\lambda = 0.6$ is used in the current setup), and g is a transform function ($\log(\cdot)$ is used in this paper). Hence, given both Eq.(6) and (7), the following formula is derived:

$$\begin{aligned} &\log\left(S_{lex}\left(Lex_k\right)\right) \\ &= \sum_{i=1}^{n} \left\{\lambda \log P\left(c_{k_i} \mid w_i\right) + (1-\lambda) \log P\left(c_{k_i} \mid c_{k_{i-1}}\right)\right\}. \end{aligned} \tag{8}$$

It is noted that the above generalized form reduced to the formulation of [Chur 88] when the transform function is log function and λ is 0.5.

2.3. Syntactic Score Function

To show the computing mechanism for the syntactic score, we take the syntax tree in Fig.1 as an example. The syntax tree is decomposed into a number of phrase levels. Each phrase level (also called a *sentential form*) consists of a set of symbols (terminal or nonterminal) which can derive all the terminal symbols in a sentence. Let label t_i in Fig.1 be the time index for each state transition of a LR parser, and L_i be the i-th phrase level. Thus, a transition from phrase level L_i to phrase level L_{i+1} is equivalent to a **redue** action at time t_i.

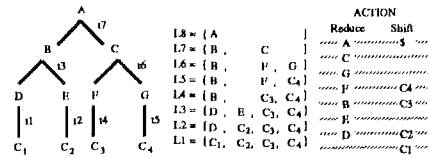


Figure 1 The decomposition of a syntax tree into phrase levels.

The syntactic score of the syntax tree in Fig.1 is then defined as

$$S_{syn}(syn_A) \equiv P(L_8, L_7, \cdots, L_2 \mid L_1) = \prod_{i=2}^{8} P\left(L_i \mid L_1^{i-1}\right) \approx \prod_{i=2}^{8} P(L_i \mid L_{i-1}), \quad (9)$$

where syn_A is the parse tree, and L_1 through L_8 represent different phrase levels. Note that the product terms in the last formula correspond to the rightmost derivation sequence in a general LR parser [Su 91c], with left and right contexts taken into account. Therefore, such a formulation is especially 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 Eq.(9) gives rise to the *normalization problem* for ambiguous syntax trees with different number of nodes. An alternative to relieve this problem is to compact multiple highly correlated phrase levels into one in evaluating the syntactic scores. The formulation is expressed as follows [Su 91c]:

$$\begin{aligned} & S_{syn}(syn_A) \\ & \approx P(L_8, L_7, L_6 \mid L_5) \times P(L_5 \mid L_4) \times P(L_4, L_3 \mid L_2) \times P(L_2 \mid L_1) \\ & \approx P(L_8 \mid L_5) \times P(L_5 \mid L_4) \times P(L_4 \mid L_2) \times P(L_2 \mid L_1). \end{aligned} \quad (10)$$

Because the number of *shifts*, i.e., the number of terms in Eq.(10), is always the same for all ambiguous syntax trees, the normalization problem is then resolved. Moreover, it provides a way to consider both *intra-level context-sensitivity* and *inter-level correlation* of the underlying context-free grammar. With such a score function, the capability of *context-sensitive parsing* (in probability sense) can be achieved with a *context-free grammar*.

3. Discrimination and Robustness Oriented Adaptive Learning

3.1. Concepts of Adaptive Learning

The general idea of adaptive learning is to adjust the model parameters (in this paper, they are lexical scores and syntactic scores) to achieve the desired criterion (in our case, it is to minimize the error rate). To explain clearly how the adaptive learning works, we take the sentence "*I saw a man.*" as an example. The lexical category (i.e., part of speech) and its corresponding log score for each word are listed in Table 1.

Word	Category (part of speech)	log P(c|w)
I	pron (pronoun)	-0.22
	n (noun)	-0.39
saw	vi (intransitive verb)	-0.52
	vt (transitive verb)	-0.16
a	art (article)	-0.02
	prep (preposition)	-1.30
man	n (noun)	0

Table 1 Categories for words and their log word-to-category scores.

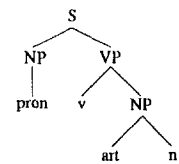
In Table 1, the log word-to-category score, $\log(P(c \mid w))$, for each word is estimated from the training corpus by calculating their relative frequencies. For example, in the training corpus, the word "I" is used as *pronoun* for 60 times, and 40 times as *noun*. Then, the log word-to-category scores can be calculated as follows.

$$\begin{aligned} \log_{10} P(\text{pron} \mid \{\text{I}\}) &= \log_{10}\left(\frac{60}{60+40}\right) = -0.22, \\ \log_{10} P(\text{n} \mid \{\text{I}\}) &= \log_{10}\left(\frac{40}{60+40}\right) = -0.39. \end{aligned} \quad (11)$$

In this example, there are 2*2*2*1=8 possible different ways to assign lexical categories to the input sentence. When these 8 possible lexical sequences are parsed, only four of them are accepted by our parser. They are listed as follows:

1. pron vt art n
2. n vt art n
3. pron vi prep n
4. n vi prep n.

The syntactic scores of different parse trees are then calculated according to Eq.(10). A parse tree corresponding to the lexical sequence "[pron vt art n]" is drawn below as an example.



The log syntactic scores for those four grammatical inputs are computed and listed in Table 2.

Input Lexical Sequence	log syntactic score
[pron vt art n]	(-0.7)+(-0.3)+(-0.3)+(-0.2) = -1.5
[n vt art n]	(-0.2)+(-0.3)+(-0.3)+(-0.2) = -1.0
[pron vi prep n]	(-0.7)+(-0.7)+(-0.4)+(-0.3) = -2.1
[n vi prep n]	(-0.2) +(-0.7)+(-0.4)+(-0.3) = -1.6

Table 2 log syntactic scores of the grammatical input lexical sequences.

According to Eq.(5), the total *log integrated score* (log S_{lex}+logS_{syn}) for each parsed sentence hypothesis is calculated. For example, the log lexical score for "I/[pron] saw/[vt] a/[art] man/[n]" = (-0.22-0.16-0.02-0)= -0.4. Finally, the log integrated scores for the above grammatical inputs are listed as follows:

candidate -1. log integrated score = (-0.40–1.5 = -1.90) : I/[pron] saw/[vt] a/[art] man/[n]

candidate -2. log integrated score = (-0.57–1.0 = -1.57) : I/[n] saw/[vt] a/[art] man/[n]

candidate -3. log integrated score = (-2.04–2.1 = -4.71) : I/[pron] saw/[vi] a/[prep] man/[n]

candidate -4. log integrated score = (-2.21–1.6 = -3.81) : I/[n] saw/[vi] a/[prep] man/[n]

Among these four candidates, the candidate 1 is regarded as the desired selection by linguists. Since our decision criterion will select the candidate which has the highest integrated score, i.e., the second one; I/[n] saw/[vt] a/[art] man/[n], it results in a decision error in this case.

To remedy this error, adaptive learning procedure is adopted to adjust the score values iteratively, including lexical and syntactic scores, until the integrated score of the correct candidate (i.e., candidate 1) raises to the highest rank. In this paper, parameters which are adjusted by adaptive learning procedure are those log scores, including log $P\left(c_{k_i} \mid w_i\right)$, log $P\left(c_{k_i} \mid c_{k_1^{i-1}}\right)$ and log $P\left(\mathrm{L_i} \mid \mathrm{L_1^{i-1}}\right)$. The amount of adjustment in each iteration depends on the *misclassification distance*. Misclassification distance is defined as the difference between the score of the top candidate and that of the correct one. (In the above example, distance = (score of correct candidate)-(score of top candidate) = (-1.90)-(-1.57)= –0.33). From iteration to iteration, the parameters (both lexical and syntactic scores) are adjusted so that the integrated score of the correct candidate is increased, and the integrated score of the wrong candidate is decreased at the same time. The learning procedure for a sentence is stopped when the candidate of this sentence is correctly selected. To make the explanation of this adaptive learning procedure clear, we assume lexical scores are unchanged during learning. That is, only the parameters of the syntactic scores are adjusted. The details of adaptive learning for adjusting syntactic scores are listed as follows:

Initialization

candidate -1. $^{\Delta}$ syntactic score = [-0.7 -0.3 -0.3 -0.2] = -1.5, log integrated score = -1.9;

candidate -2. * syntactic score = [-0.2 -0.3 -0.3 -0.2] = -1.0, log integrated score = -1.57;

candidate -3. syntactic score = [-0.7 -0.7 -0.4 -0.3] = -2.1, log integrated score = -4.71;

candidate -4. syntactic score = [-0.2 -0.7 -0.4 -0.3] = -1.6, log integrated score = -3.81;

Iteration 1

candidate -1. $^{\Delta}$ syntactic score = [-0.5 -0.3 -0.3 -0.2] = -1.3, log integrated score = -1.7;

candidate -2. * syntactic score = [-0.3 -0.3 -0.3 -0.2] = -1.1, log integrated score = -1.67;

candidate -3. syntactic score = [-0.5 -0.7 -0.4 -0.3] = -1.9, log integrated score = -3.94;

candidate -4. syntactic score = [-0.3 -0.7 -0.4 -0.3] = -1.7, log integrated score = -3.91;

Iteration 2

candidate -1. $^{\Delta *}$ syntactic score = [-0.2 -0.3 -0.3 -0.2] = -1.0, log integrated score = -1.4; (stop learning)

candidate -2. syntactic score = [-0.6 -0.3 -0.3 -0.2] = -1.4, log integrated score = -1.97;

candidate -3. syntactic score = [-0.2 -0.7 -0.4 -0.3] = -1.6, log integrated score = -3.64;

candidate -4. syntactic score = [-0.6 -0.7 -0.4 -0.3] = -2.0, log integrated score = -4.21;

(where * denotes the top candidate, and Δ denotes the desired candidate)

It is clear that after the second iteration, parameters have been adjusted so that the desired candidate (i.e., candidate 1) would be selected.

3.2. Procedure of Discrimination Learning

Since correct decision only depends upon correct rank ordering of the integrated scores for all ambiguities, not their real value, a discrimination-oriented approach should directly pursue correct rank ordering. To derive the discrimination function, the probability scores mentioned above are first jointly considered. Then, a discrimination-oriented function, namely $g(\cdot)$, is defined as a measurement of above mentioned score functions, so that it can well preserve the correct rank ordering [Su 91a]. Here, $g(\cdot)$ is chosen as the weighted sum of log

lexical and log syntactic scores, i.e.,

$$
\begin{aligned}
& g(Syn_k) \\
& = w_{lex} \cdot \log S_{lex}(Lex_k) + w_{syn} \cdot \log S_{syn}(Syn_k) \\
& = w_{lex} \cdot \sum_{i=1}^{n} \log P\left(c_{k_i} | c_{k_1}^{i-1}, w_1^n\right) + w_{syn} \cdot \sum_{i=1}^{n} \log P\left(I_{ri} | I_1^{i-1}\right) \\
& = w_{lex} \cdot \sum_{i=1}^{n} \lambda_{lex}(i) + w_{syn} \cdot \sum_{i=1}^{n} \lambda_{syn}(i),
\end{aligned}
\tag{12}
$$

where $\lambda_{lex}(i) = \log P\left(c_{k_i} | c_{k_1}^{i-1}, w_1^n\right)$, and $\lambda_{syn}(i) = \log P\left(I_{ri} | I_1^{i-1}\right)$. Both stand for the log lexical score and the log syntactic score of the i-th word for the k-th syntactic ambiguity, respectively. In addition, w_{lex} and w_{syn} correspond to the weights of lexical and syntactic scores, respectively.

If the parse tree of a sentence is misselected, the parameters (i.e., the lexical and the syntactic scores) are adjusted via the proposed adaptive learning procedure. Otherwise, no parameters would be adjusted. When misselection occurs, the misclassification distance, d_{sc}, is less than zero. This misclassification distance is defined as the difference between the log integrated score of the correct candidate and that of the top one. A specific term of the syntactic score components in the *(t+1)*-th iteration of the *correct candidate*, say $\lambda_{syn}^{(t+1)}(j)$, would be adjusted as follows:

$$
\begin{cases}
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j) + \Delta\lambda_{syn}^{(t)}(j), & d_{sc} \le 0, \\
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j), & otherwise.
\end{cases}
\tag{13}
$$

At the same time, the term of the syntactic score components of the *top candidate* would be adjusted according to the following formulas:

$$
\begin{cases}
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j) - \Delta\lambda_{syn}^{(t)}(j), & d_{sc} \le 0, \\
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j), & otherwise,
\end{cases}
\tag{14}
$$

where $\Delta\lambda_{syn}^{(t)}(j)$ is the amount of adjustment. This value is represented as

$$
\Delta\lambda_{syn}^{(t)}(j) = \frac{\epsilon \cdot d_0}{d_{sc}^2 + d_0^2} \cdot \frac{w_{syn}}{\sum_{i=1}^{n} \left[w_{lex} \cdot \lambda_{lex}^{(t)}(i) + w_{syn} \cdot \lambda_{syn}^{(t)}(i) \right]},
\tag{15}
$$

where d_0 is a constant which stands for a window size, and ϵ is the learning constant for controlling the speed of convergence. The learning rule for adjusting the lexical scores can be represented in a similar manner. Notice that only the parameters of the top candidate and those of the correct candidate would be adjusted when misselections occur. Those parameters of other wrong candidates would not be adjusted in this adaptive learning procedure. From Eq.(13), (14) and (15), it is clear that the score of the correct candidate will increase and that of wrong candidate will decrease from iteration to iteration until the correct candidate is selected. For the purpose of clarity, the detailed derivations of the above adaptive learning procedure will not be given here. Interested readers can contact the authors for details.

3.3. Robustness Issues

Since it is easy to improve the performance in a training set by adopting a model with a large number of parameters, the error rate measured in the training set frequently turns out to be over-optimistic. Moreover, the parameters estimated from the training corpus may be quite differ from that obtained from the real applications. These phenomena may occur due to the factors of finite sampling size, style mismatch, or domain mismatch, etc. To achieve a better performance in the real application, one must deal with the possible mismatch of parameters, or statistical variation between the training corpus and the real application. One way to achieve this goal is to enlarge the inter-class distance to achieve maximum separation [Su 91a] between the correct candidate and the other candidates. That is, this approach provides a tolerance zone between different candidates for allowing possible data scattering in the real application.

Traditional adaptive learning methods [Amar 67, Kata 90] stop adjusting parameters once the input pattern has been correctly classified. However, if we stop adjusting parameters under the condition that the observations are correctly classified in the training corpus, the distance between the correct candidate and other ambiguities may still be too small. Thus, it is vulnerable to deal with possible modeling errors and statistical variations between the training corpus and the real application. Su [Su 91a] has proposed a robust learning procedure which continues to enlarge the margin between the correct candidate and the top one, even if the syntax tree of the sentence has been correctly selected. That is, the parameters will not be adjusted only if the distance between the correct candidate and the others has exceeded a given threshold. The learning rules in Eq.(13), (14) are then modified as follows.

If $d_{sc} \le \delta$, where δ is a preset margin, the syntactic score in the (t+1) iteration for the *correct candidate* is adjusted according to the following formulas:

$$
\begin{cases}
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j) + \Delta\lambda_{syn}^{(t)}(j), & d_{sc} \le \delta, \\
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j), & otherwise.
\end{cases}
\tag{16}
$$

And, the syntactic score of the *top candidate* is adjusted as follows:

$$
\begin{cases}
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j) - \Delta\lambda_{syn}^{(t)}(j), & d_{sc} \le \delta, \\
\lambda_{syn}^{(t+1)}(j) = \lambda_{syn}^{(t)}(j), & otherwise.
\end{cases}
\tag{17}
$$

4. Simulations

The following experiments are conducted to investigate the advantage of the proposed discrimination and robustness oriented adaptive learning procedure. In the experiments, 4,000 sentences, which are extracted from IBM technical manuals are first associated with their corresponding correct category sequences and correct parsed trees by linguists. The corpus are then partitioned into a training corpus of 3,200 sentences and a test set of 800 sentences. Next, the lexical and syntactic probabilities are estimated from the data in the training corpus. Afterwards, the sentences in the test set are used to evaluate the performance of the proposed algorithm using the estimated lexical and syntactic probabilities. This integrated score function approach using the estimated probabilities is considered as the *baseline* system. Performances of discrimination oriented adaptive learnings with and without robustness enhancement are then evaluated. The accuracy rate of the syntactic ambiguity resolution for the training corpus and the test set are summarized in Table 3. (Note that the top candidate is selected from all possible parses allowed by the grammars of the system; therefore, the baseline performance is evaluated under a highly ambiguous environment.)

	Training Corpus	Test Set
Baseline	79.75	46.00
+ Basic version of learning	95.50	56.88
+ Robust version of Learning	96.03	60.62

Table 3 Accuracy rate (in %) of syntactic disambiguation

Table 3 shows that syntax tree accuracy rate is improved from 46% to 56.88% using the basic version of discrimination oriented adaptive learning procedure. This significant improvement shows the superiority of the adaptive learning procedure for dealing with the disambiguation task. Furthermore, when the robust version of learning procedure is adopted, the performance is improved further (from 56.88% to 60.62%). It means that the robustness of the learning procedure is indeed enhanced by enlarging the distance between the correct candidate and other candidates. Moreover, not only the accuracy rate of syntax tree is improved using adaptive learning, but also that of lexical sequence is improved. In this paper, a lexical sequence is regarded as "*correct*" only if all the lexical categories in a sentence perfectly match those selected by linguists. In other words, we are measuring "*sentence accuracy rate*" in contract to "*word accuracy rate*" as adopted in [Chur 88, Gars 87]. Table 4 shows that the basic version of adaptive learning procedure improves the sentence accuracy rate of lexical sequences about 5% (from 77.12% to 82.38%). Again, with the robust version of learning, the accurate rate of lexical sequences is greatly enhanced.

	Training Corpus	Test Set
Baseline	91.41	77.12
+ Basic version of learning	98.91	82.38
+ Robust version of Learning	98.53	87.88

Table 4 Sentence accuracy rate (in %) of lexical sequences

The behavior of each iteration of the adaptive learning process is shown in Figure 2. Through observing this figure, we can conclude that if the robustness issues are not considered during learning, the performance of the test set would decrease as the training process goes on. This is the phenomena of over-tuning. However, by forcing the learning procedure to continue until the separation between the correct candidate and the top one exceeds the desired margin, the performance of the test set can be further improved, and no degradation phenomenon is observed.

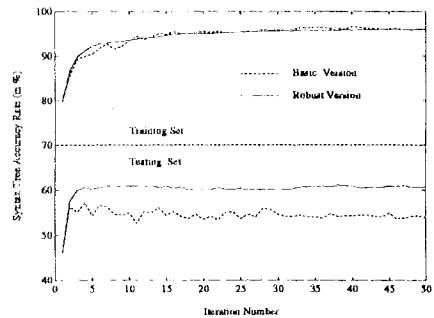


Figure 2 Syntax tree accuracy rate versus iterations for basic and robust version of adaptive learning

5. Summary

Because of insufficient training data, and approximation error introduced by the language model, traditional statistical approaches, which resolve ambiguities by indirectly and implicitly using maximum likelihood method, fail to achieve high performance in real applications. To overcome these problems, adaptive learning is proposed to pursue the goal of minimizing discrimination error directly. The performance of syntactic ambiguity resolution is significantly improved using the discrimination oriented analysis. In addition, the sentence accuracy rate of the lexical sequences is also improved. Moreover, the performance is further enhanced by using the robust version of learning procedure, which enlargeds the margin between the correct candidate and its candidates. The final results show that using the basic version of learning, the syntax tree selection accuracy rate is improved about 10% (from 46% to 56.88%), and the total improvement is over 14% using robust version learning. Also, the sentence accuracy rate for lexical sequences is improved from 77.12% to 82.38 and 87.88% using the basic and robust version of learning procedure, respectively.

Reference

[Amar 67] Amari S., "A theory of adaptive pattern classifiers," IEEE Trans. on Electronic Computers, Vol. EC-16, pp. 299–307, June 1967.

[Chen 91] Chen S.-C., J.-S. Chang, J.-N. Wang, and K.-Y. Su, "ArchTran: A Corpus-based Statistics-oriented English-Chinese Machine Translation System," *Proc. of Machine Translation Summit III*, Washington, D. C., U.S.A., July 1–3, 1991.

[Chur 88] Church, K., "A Stochastic Parts Program and Noun Phrase for Unrestricted Text," ACL Proceedings of 2nd Conference on Applied Natural Language Processing, pp.136–143, Austin, Texas, U.S.A., 9–12 Feb. 1988.

[Gars 87] Garside, R., G., Leech, and G., Sampson, "The Computational Analysis of English: A Corpus-Based Approach," London: Longman.

[Kata 90] S.Katagiri. C.H. Lee, "A Generalized Probabilistic Decent Method," *Proc. Acous. Sco. of Japan*, 2-p-6, pp. 141–142, Nagoya, Sept 1990.

[Su 88] Su K.-Y., and J.-S. Chang, "Semantic and Syntactic Aspects of Score Function," Proc. COLING-88, Vol.2, pp. 642–644, 12th Int. Conf. on Comput. Linguistic, Budapest, Hurgay, 22–27, Aug. 1988.

[Su 90a] Su, K.-Y., and J.-S Chang, 1990. "Some Key Issues in Designing MT Systems," *Machine Translation*, vol. 5, no. 4, pp. 265-300, 1990.

[Su 90b] Su K.-Y., T.-H. Chiang and Y.-C. Lin, "A Unified Probabilistic Score Function for Integrating Speech and Language Information in Spoken Language Processing," Proceeding of 1990 International Conference on Spoken Language Processing, pp.901–904, Kobe, Japan, 19–22 Nov. 1990.

[Su 91b] Keh-Yih Su, Tung-Hui Chiang and Yi-Chung Lin, "A Robustness and Discrimination Oriented Score Function for Integrating Speech and Language Processing," *Proceeding of the 2nd European Conference on Speech Communication and Technology*, Genova, Italy, pp. 207–210, Sep. 24–26 1991.

[Su 91a] Su K.-Y., and C.-H. Lee, "Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approaches," Proc. ICASSP-91, pp.541–544, Toranto, Canada, 14–17 May, 1991.

[Su 91c] Su K.-Y., J.-N. Wang, M.-H. Su, and J.-S. Chang, "GLR Parsing with Scoring," in Tomita, Masaru (ed.), Generalized LR Parsing, Chapter 7, pp.93–112, Kluwer Academic Publisher 1991..

[Su 92a] Keh-Yih Su, Tung-Hui Chiang and Yi-Chung Lin, "An Unified Framework to Incorporate Speech and Language Information in Spoken Language Processing" to appear in the Proceeding of *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-92*, San Francisco, California, U.S.A., March 23–26 1992.

[Su 92b] Su K.-Y., J.-S. Chang and Y.-C. Lin, "A Unified Approach to Disambiguation Using A Uniform Formulation of Probabilistic Score Function," in preparation.