A Statistics-Based Chinese Parser

Zhou Qiang

The State Key Laboratory of Intelligent Technology and Systems
Dept. of Computer Science, Tsinghua University
Beijing 100084, P. R. China
zhouq@s1000e.cs.tsinghua.edu.cn

Abstract

This paper describes a statistics-based Chinese parser, which parses the Chinese sentences with correct segmentation and POS tagging information through the following processing stages: 1) to predict constituent boundaries, 2) to match open and close brackets and produce syntactic trees, 3) to disambiguate and choose the best parse tree. Evaluating the parser against a smaller Chinese treebank with 5573 sentences, it shows the following encouraging results: 86% precision, 86% recall, 1.1 crossing brackets per sentence and 95% labeled precision.

1 Introduction

Since the large-scale annotated corpora, such as Penn Treebank[MSM93], have been built in English, statistical knowledge extracted from them has been shown to be more and more crucial for natural language parsing and disambiguation. Hindle and Rooth(1993) tried to use word association information to disambiguate the prepositional phrase attachment problem in English. Brill(1993a) proposed a transformation-based error-driven automatic learning method, which has been used in part-of-speech(POS) tagging[Bri92], text chunking[RM95] and sentence bracketing[Bri93b]. Bod's data oriented parsing technique directly used an annotated corpus as a stochastic grammar for parsing[RB93]. Based on the statistical decision-tree models automatically learned from treebank, Magerman's SPATTER parser showed good performance in parsing Wall Street Journal texts[DM95]. Collins(1996) described a statistical parser based on probabilities of dependencies between head-words in treebank, which can perform at least as well as SPATTER.

As a distinctive language, Chinese has many characteristics different from English. Although Chinese information processing techniques have made great progress since 1980, how to use statistical information efficiently in Chinese parser is still a virgin land waiting to explore. This paper describes our preliminary work to build a Chinese parser based on different kinds of statistics extracted from treebank. It tries to parse the Chinese sentences with correct segmentation and POS tagging information through the following processing stages: 1) to predict constituent boundaries using local context statistics, 2) to match the open and close brackets and produce syntactic trees using boundary tag distribution data and syntactic tag reduction rules. 3) to disambiguate parse trees using stochastic

context-free grammar(SCFG) rules. Evaluating the parser against a smaller Chinese treebank with 5573 sentences, it shows the following encouraging results: 86% precision, 86% recall, 1.1 crossing brackets per sentence and 95% labeled precision. This work illustrates that some simple treebank statistics may play an important role in Chinese sentence parsing and disambiguation.

The rest of the paper is organized as follows. Section 2 briefly introduces the statistical data set used in our parser. Section 3 describes the detailed parsing algorithm, including the boundary prediction model, bracket matching model, matching restriction schemes and the statistical disambiguation model. Section 4 gives current experimental results. At last, summary and future work are discussed in section 5.

2 Statistics from treebank

The difficulty to parse natural language sentences is their high ambiguities. Traditionally, disambiguation problems in parsing have been addressed by enumerating possibilities and explicitly declaring knowledge which might aid most interesting natural language processing problems. As the large-scale annotated corpora become available nowadays, automatic knowledge acquisition from them becomes a new efficient approach and has been widely used in many natural language processing systems.

Treebanks are the collections of sentences marked with syntactic constituent structure trees. The statistics extracted from a large scale treebank will show useful syntactic distribution principles and be very helpful for disambiguation in a parser. Some statistical data and rules used in our parser are briefly described as follows:

(1) boundary distribution data(S1)

This group of data shows the different influence of context information on the constituent boundaries in a sentence, counted by the co-occurrence frequencies of different constituent boundary labels(b_i) with the word(w_i) and part-of-speech(POS) tags(t_i), which include: (a) the co-occurrence frequencies with functional words: $f(w_i,b_i)$; (b) the co-occurrence frequencies with a single POS tag: $f(t_i,b_i)$; (c) the co-occurrence frequencies with local POS tags: $f(b_i, t_i, t_{i+1})$ or $f(t_{i-1}, t_i, b_i)$. They play an important role in the prediction of constituent boundary locations.

(2) Syntactic tag reduction data(S2)

This group of data records the possibilities for the constituent structures to be reduced as different syntactic tags, represented by a set of statistical rules:

constituent structure --> {syntactic tag, reduction probability}+.

For example, the rule v+n --> vp 0.93, np 0.07 indicates that a syntactic constituent composed by a verb(v) and a noun(n) can be reduced as a verb phrase(vp) with the probability 0.93, and as a noun phrase(np) only 0.07[1]. Based on them, it is easy to determinate the suitable syntactic tag for a parsed constituent according to its internal structure components.

---

[1] In Chinese, there are a group of verbs with especial syntactic functions. They can directly modify a noun, such as the verb "*xunlian*(train)" in the phrase "*xunlian shouche*(training handbook)". Therefore, we have the noun phrases with constituent structure "v+n" in Chinese treebank.

(3) syntactic tag distribution on a boundary(S3)

This group of data expresses the possibilities for an open or a close bracket to be the boundary of a constituent with certain kind of syntactic tags under different POS context. For example,

n [p --> vp 0.531, pp 0.462, np 0.007,

indicates that the probability for an open bracket under the context of noun(n) and preposition(p) to be the left boundary of a verb phrase(vp) is 0.531, a prepositional phrase(pp) 0.462, and a noun phrase(np) 0.007. This kind of data provides the basis for matching brackets and labeling the matched constituents.

(4) constituent preference data(S4)

This group of data records the preference for a constituent to be combined with its left adjacent constituent or the right adjacent one under local context, counted by the frequencies of different constituent combination cases in treebank(see Figure 1), which are represented as:

{<constituent combination case>, <left combination frequency>, <right combination frequency>}

For example, {p+np+vp, 190, 0} indicates that the combination frequency of the noun phrase(np) with preposition(p) under the local context "p+np+vp" is 190, and with verb phrase(vp) is 0. They will be helpful in preference matching model.

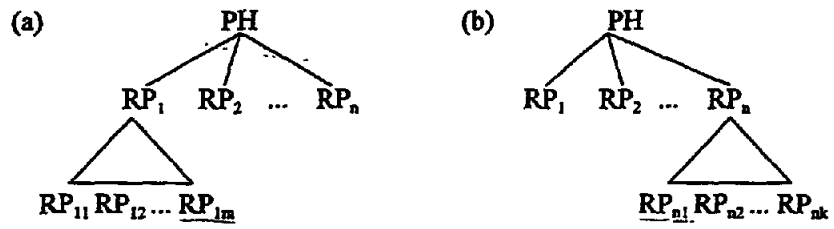

Figure 1. The overview of different constituent combination cases in treebank. (a) The left combination case: $RP_{11}\,RP_{12}\ldots \underline{RP_{1m}}\,RP_2\ldots RP_n$; (b) The right combination case: $RP_1\,RP_2\ldots \underline{RP_{n-1}}\,RP_{n1}\,RP_{n2}\ldots RP_{nk}$.

(5) probabilistic constituent structure rules(S5)

The group of data associates a probability to each constituent structure rule of the grammar, also called as stochastic context-free grammar(SCFG) rules. The probability of a constituent structure rule $A \rightarrow \alpha\beta\gamma$ can be calculated as follows:

$$P(A \rightarrow \alpha\beta\gamma) = \frac{f(A \rightarrow \alpha\beta\gamma)}{\sum_{A \rightarrow \alpha\beta\gamma} f(A \rightarrow \alpha\beta\gamma)}$$

where $f(A \rightarrow \alpha\beta\gamma)$ is the frequency of the constituent $[_A\ \alpha\ \beta\ \gamma\ ]$ in treebank. It provides useful information for syntactic disambiguation.

3 The parsing algorithm

The aim of the parser is to take a correctly segmented and POS tagged Chinese sentence as input(for example Figure 2(a)) and produce a phrase structure tree as output(Figure 2(b)). A parsing algorithm to this problem must deal with two important issues: (1) how to produce the suitable syntactic trees from a

tagged word sequence, (2) how to select the best tree from all of the possible parse trees.

The key of our approach is to simplify the parsing problem as two processing stages. First, the statistical prediction model assigns a suitable constituent boundary tag to every word in the sentence and produce a partially bracketed sentence(Figure 2(c)). Second, the preference matching model constructs the syntactic trees through bracket matching operations and select a preference matched tree using probability score scheme as output(Figure 2(d)).

(a) 我(my)/r 弟弟(brother)/n 要(want)/v 买(buy)/v 两(two)/m 个(-classifier)/q 足球(football)/n 。(period)/w [2]
My brother wants to buy two footballs.

(b) [zj [dj [np 我/r 弟弟/n] [vp 要/v [vp 买/v [np [mp 两/m 个/q] 足球/n]]]] 。/w]

(c) [我/r 弟弟/n] [要/v [买/v [两/m 个/q] 足球/n] 。/w]

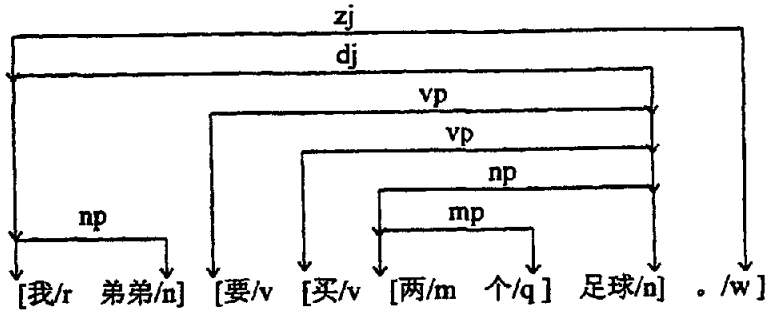

Figure 2. An overview of the representation used by the parser. (a) The segmented and tagged sentence; (b) A candidate parse-tree(the correct one), represented by its bracketed and labeled form; (c) A constituent boundary prediction representation of (a); (d) A preference matched tree of (c). Arrows show the bracket matching operations.

3.1 The boundary prediction model

A constituent boundary parse of a sentence can be represented by a sequence of boundary tags. Each tag corresponds to one word in the sentence, and can value *L*, *M* or *R*, respectively meaning the beginning, continuation or termination of a constituent in the syntactic tree. A constituent boundary parse *B* is therefore given by $B = (b_1, b_2, ..., b_n)$, where b_i is the boundary tag of the *i*th word and *n* is the number of

[2] The POS and syntactic tags used in this sentence are briefly describes as follows. Some detailed information about our POS and syntactic tagsets can be found in [ZQd96].

[POS tags]: r--pronoun, n--noun, v--verb, m--numeral, q--classifier, w--punctuation.

[Syn tags]: np--noun phrase, mp--numeral-classifier phrase, vp--verb phrase, dj--simple sentence pattern, zj--complete sentence.

words in the sentence.

Let $S=<W,T>$ be the input sentence for syntactic analyzing, where $W=w_1,w_2,...,w_n$ is the word sequence in the sentence, and $T=t_1,t_2,...,t_n$ is the corresponding POS tag sequence, i.e., t_i is the POS tag of w_i. Just like the statistical approaches in many automatic POS tagging programs, our job is to select a constituent boundary sequence B' with the highest score, $P(B|S)$, from all possible sequences.

$$B' = \arg\max P(B|S) = \arg\max P(S|B)P(B) \tag{1}$$

Assume the effects of word information and POS information are independent, we get

$$P(S|B) = P(W|B)P(T|B) \tag{2}$$

Furthermore, replace $P(W|B)$ and $P(T|B)$ by the approximation that each constituent boundary is determined only by a functional word(w_i) or local POS context(C_i).

$$P(S|B) = \prod_{i=1}^{n} P(w_i|b_i)P(C_i|b_i) \tag{3}$$

In addition, for $P(B)$, it is possible to use simple bigram approximation:

$$P(B) = \prod_{i=1}^{n} P(b_i|b_i-1) \tag{4}$$

where, $P(b_1|b_0) = P(b_1)$.

Therefore, a statistical model for the automatic prediction of constituent boundary is set up.

$$B' = \arg\max \prod_{i=1}^{n} P(w_i|b_i)P(C_i|b_i)P(bi|b_i-1) \tag{5}$$

The probability estimates of the model are based on the boundary distribution data(S1) described in section 2, and can be calculated through maximum likelihood estimation(MLE) method. For example,

$$\begin{aligned} P(C_i|b_i) &= \max[P(t_i,t_i+1|bi), P(t_i-1,ti|b_i)] \\ &= \max[f(bi,t_i,t_i+1)/f(bi), f(t_i-1,ti,bi)/f(bi)] \end{aligned} \tag{6}$$

There are two directions to improve the prediction model. First, many post-editing rules that are manually developed or automatically learned by an error-driven learning method can be used to refine the automatic prediction outputs[ZQ96]. Second, a new statistical model based on forward-backward algorithm will produce multiple boundary predictions for a word in the sentence[ZZ96].

3.2 Basic matching model

In order to build a complete syntactic tree based on the boundary prediction information, two basic problems must be resolved. The first one is how to find the reasonable constituents among the partially bracketed sentence. The second one is how to label the found constituents with suitable syntactic tags. This section will propose some basic concepts and operations of the matching model to deal with the first problem, and section 3.3.1 will give methods to resolve the second one. The formal description of the bracket matching model can be found in [ZQd96].

(1) Simple matching operation

The simple matching SM(i,j) is the matching of the open bracket ($b_i = L$) and the close bracket ($b_j = R$) under the condition: $\forall\ b_k = M$, $k \in (i,j)$.

(2) Expanded matching operation

The expanded matching EM(i,j) is the matching of the open bracket ($b_i = L$) and the close bracket ($b_j = R$) under one of the following conditions:

(a) $\exists$ {SM(i,k), i<k<j} and $\forall\ b_p = M$, p∈(k,j).

(b) $\exists$ {SM(k,j), i<k<j} and $\forall\ b_p = M$, p∈(i,k).

(c) $\exists$ {SM(i,k) & SM(p,j), i<k<p<j} and $\forall\ b_q = M$, q∈(k,p).

(3) Matched constituent

A matched constituent MC(i,j) is a syntactic constituent constructed by the simple matching operation SM(i,j) or the expanded matching operation EM(i,j).

Therefore, a basic matching algorithm can be built as follows: Starting from the preprocessed sentence S=<W,T,B>, we first use the simple matching operation, then the expanded matching operation, so as to find every possible matched constituent in the sentence. The complete matching principle will guarantee that this algorithm will produce all matched constituents in the sentence. See [ZQd96] for more detailed information of this principle and its formal proof.

3.3 Matching restriction schemes

The basic matching algorithm based on the complete matching principle is inefficient, because many ungrammatical or unnecessary constituents can be produced by two matching operations. In order to improve the efficiency of the algorithm, some matching restriction schemes are needed, which include, (1) to label the matched constituents with reasonable syntactic tags, (2) to set the matching restriction regions, (3) to discard unnecessary matching operations according to local preference information.

3.3.1 Constituent labeling

The aim of labeling approach is to eliminate the ungrammatical matched constituents and label the suitable syntactic tags for the reasonable constituents, according to their internal structure and external context information.

First, some common erroneous constituent structures can be enumerated under current POS tagset and syntactic tagset. Moreover, many heuristic rules to find ungrammatical constituents can also be summarized according to constituent combination principles. Based on them, most ungrammatical constituents can be eliminated.

Then, we can assign a suitable syntactic tag to each matched constituent through the following sequential processing steps:

(a) Set the syntactic tags according to the statistical reduction rule, if it can be searched in syntactic tag reduction data(S2) using the constituent structure string as a keyword.

(b) Determine the syntactic tags according to the intersection of the tag distribution sets of the open and close bracket on the constituent boundary, if they can be found in statistical data(S3).

(c) Assign an especial tag that is not in the current syntactic set to every unlabeled constituent after above two processing steps.

3.3.2 Restriction regions for matching

There are many regional restricted constituents in natural language, such as reference constituents in the pair of quotation marks: " ... ", and the regular collocation phrase: "*zai ... de shihou*(when ...)" in Chinese. The constituents inside them can not have syntactic relationship with the outside ones.

In bracket matching model, these cases can be generalized as a matching restriction region (MRR), which is informally represented as the region <RL,RR> in Figure 3.

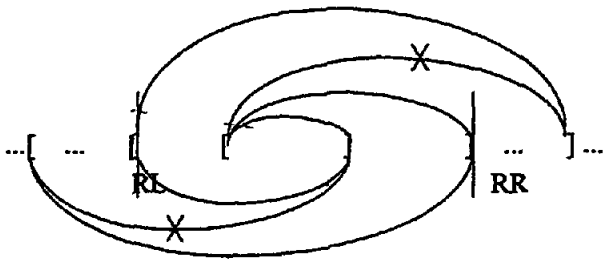

Figure 3: Informal description of a MRR <RL,RR>. The arcs show bracket matching operations, and the arcs marked with 'X' indicate that such matching operations are forbidden.

Therefore, the basic matching algorithm can be improved by adding the following restrictions:

(a) To restrict the matching operations inside MRR and guarantee them can't cross the boundary of the MRR.

(b) To reduce the MRR as a constituent MC(RL,RR) after all matching operations inside MRR have been finished, so as to make it as a whole during the following matching operations.

The key to use MRR efficiently is to correctly identify the possible restriction regions in the sentences. Reference [ZQd96] describes the automatic identification methods for some Chinese MRRs.

3.3.3 Local preference matching

Consider such a parsing state after the simple matching operation SM(i,j):

$$[t_{i-1} \quad MC(i,j) \quad t_{j+1}]$$

Starting from it, there are two possible expanded matching operations: EM(i-1,j) or EM(i,j+1). All of them must be processed according to basic matching algorithm, and two candidate matched constituents: MC(i-1,j) and MC(i,j+1), will be produced. But in many cases, one of these operations is unnecessary because only one candidate constituent may be included in the best parse tree. These superfluous matching operations reduces the parsing efficiency of the basic matching algorithm.

Let "A B C" to be the local matching context (For the above example, we have: A=[t_{i-1}, B= MC(i,j), and C= t_{j+1}]). $P(B,C)$ is the right combination probability for constituent 'B' and $P(A,B)$ is its left combination probability, which can be easily computed using the constituent preference data (S4) described in section 2. Set α=0.5 as the difference threshold. Then, a simple preference-based approach can be added into the basic matching algorithm to improve the parsing efficiency:

if $P(B,C)-P(A,B)>\alpha$, then the matching operation [A,B] will be discarded.

if $P(A,B)$-$P(B,C)>\alpha$, then the matching operation [B,C] will be discarded.

3.4 Statistical disambiguation model

This section describes the way the best syntactic tree is selected. A statistical approach to this problem is to use SCFG rules extracted from treebank and set a probability score scheme for disambiguation.

Assume a constituent labeled with syntactic tag PH is composed by the syntactic components RP_1, RP_2, ... , RP_n. Its parsing probability $P(PH)$ can be calculated through the following formula:

$$P(PH) = \prod_{i=1}^{n} P(RP_i) \cdot P(PH \rightarrow RP_1 RP_2 \ldots RP_n) \tag{7}$$

where the probability $P(PH \rightarrow RP_1\, RP_2 \ldots\, RP_n)$ comes from statistical data(S5) defined in section 2. In addition, if RP_i is a word component, then set $P(RP_i) = 1$.

By computing logarithm on both sides of equation (7), we will get the probability score $Score(PH)$:

$$\begin{aligned} Score(PH) &= \log P(PH) = \log\left[\prod_{i=1}^{n} P(RP_i) \cdot P(PH \rightarrow RP_1 \ldots RP_n)\right] \\ &= \sum_{i=1}^{n} \log P(RP_i) + \log P(PH \rightarrow RP_1 \ldots RP_n) \\ &= \sum_{i=1}^{n} Score(RP_i) + \log P(PH \rightarrow RP_1 \ldots RP_n) \end{aligned} \tag{8}$$

Formally, a labeled constituent MC(1,n) may be looked as a syntactic tree. Therefore, the most likely parse tree under this score model is then this kind of matched constituent with the maximum probability score, i.e. T_{best} = argmax $Score$(MC(1,n)).

4 Experimental results

In the absence of an available annotated Chinese corpus, we had to build a small Chinese treebank for training and evaluating the parser, which consists of the sentences extracted from two parts of Chinese texts: (1) test set for Chinese-English machine translation systems (Text A), (2) Singapore primary school textbooks on Chinese language (Text B). Table 1 shows the basic statistics of these two parts in the treebank.

Table 1: Basic statistics for the Chinese treebank.

	Character Number	Word Number	Sentence Number	Mean Sentence Length(words/sent.)
Text A	1434	11821	17058	8.243
Text B	4139	52606	72434	12.71

Then, the treebank is divided as a training set with 4777 sentences and a test set with 796 sentences based on balanced sampling principle. Figure 4 shows the distributions of sentence length in the training and test sets. In addition, according to the difference of word(including punctuation) number in the

sentence, all sentences in the treebank can be further classified as two sets. One is simple sentence set, in which every sentence has no more than 20 words. The other is complex sentence set, in which every sentence has more than 20 words. Therefore, we will obtain complete knowledge about the performance of the parser by the comparison of it on these two types of sentences. Table 2 shows the distribution data of simple and complex sentences in the training and test sets.

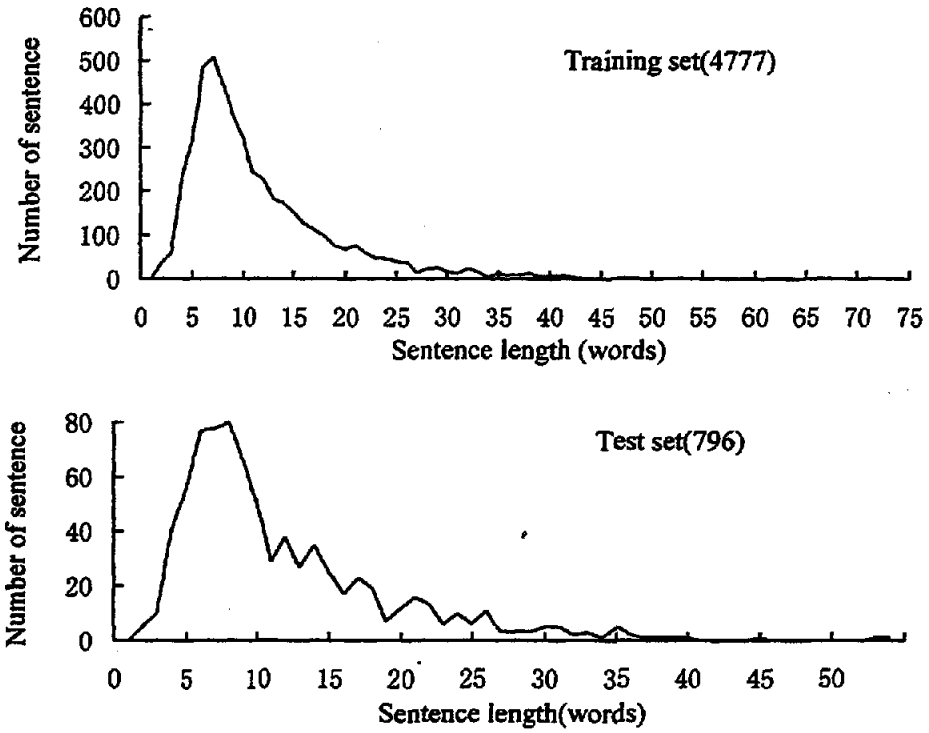

Figure 4. Distribution of sentence length in training and test sets.

Table 2: Distribution of the simple and complex sentences in the training and test sets.

	Simple Sentences		Complex Sentences		Mean Sent. Length
	Sent. Number	% in Set	Sent. Number	% in Set	
Training Set	4176	87.419	601	12.581	11.533
Test Set	683	85.804	113	16.477	14.196

In order to evaluate the performance of the current Chinese parser, we are using the following measures:

1) Matched precision(MP) =

$$\frac{\text{number of correct matched constituents in proposed parse}}{\text{number of matched constituent in proposed parse}}$$

2) Matched recall(MR) =

$$\frac{\text{number of correct matched constituents in proposed parse}}{\text{number of constituents in treebank parse}}$$

3) Crossing Brackets(CBs) = number of constituents which violate constituent boundaries with a constituent in the treebank parse.

The above measures are similar with the PARSEVAL measures defined in [Bla91]. Here, for a matched constituent to be 'correct' it must have the same boundary location with a constituent in the treebank parse.

4) Boundary prediction precision(BPP) =

$$\frac{\text{number of words with correct constituent boundary prediction}}{\text{number of words in the sentence}}$$

5) Labeled precision(LP) =

$$\frac{\text{number of correct labeled constituents in proposed parse}}{\text{number of correct matched constituent in proposed parse}}$$

6) Sentence parsing ratio(SPR) =

$$\frac{\text{number of sentences having a proposed parse by parser}}{\text{number of input sentences}}$$

Table 3 shows the experiment results. On a 80Mhz 486 personal computer with 16 megabytes RAM, the parser can parse about 1.38 sentences per second.

Table 3: Results on the training set and test set. 0 CBs, ≤ 1 CBs, ≤ 2 CBs are the percentage of sentences with 0, ≤ 1 or ≤ 2 crossing brackets respectively.

	Training Set			Test Set		
	Simple Sent.	Complex Sent.	Overall	Simple Sent.	Complex Sent.	Overall
BPP(%)		/	97.09	/	/	96.96
CBs	0.72	3.44	1.06	0.71	3.71	1.14
0 CBs(%)	67.04	12.81	60.23	69.25	13.27	61.30
≤ 1 CBs(%)	79.16	26.95	72.60	79.06	22.12	70.98
≤ 2 CBs(%)	89.56	43.76	83.81	88.72	38.05	81.53
MR(%)	89.45	82.51	87.43	89.60	80.81	86.79
MP(%)	89.42	82.40	87.38	89.28	80.71	86.54
LP(%)	95.79	93.88	95.26	95.61	93.53	95.00
SPP(%)	/	/	99.98	/	/	100.00

5 Conclusion

In this paper, we propose a statistics-based Chinese parsing algorithm. It has the following characteristics:

(1) The idea to separate constituent boundary prediction as a preprocessing stage from parser, just as the widely accepted POS tagging, is based on the following premises: (a) Most constituent boundaries in a Chinese sentence can be predicted according to their local word and POS information, (b) The parsing complexity can be reduced based on constituent boundary prediction.

(2) The proof of complete matching principle and the application of matching restriction schemes guarantee the soundness and efficiency of the matching algorithm.

(3) To use SCFG rules as a main disambiguation knowledge will cut down the hard work to manually develop a complex and detailed disambiguation rule base.

Although the experimental results are encouraging, there are many possibilities for improvement of the algorithm. Some unsupervised training methods for SCFG rules, such as inside-outside algorithm[LY90] and its improved approaches([PS92],[SYW95]), should be tried in the absence of large-scale Chinese treebanks. The disambiguation model could be extended to capture context-sensitive statistics[CC94] and word statistics([EC95],[Col96]).

Acknowledgments

The author would like to thank Prof. Yao Tianshun. Prof. Yu Shiwen and Prof. Huang Changning for their kind advice and support, and many colleagues and students in Institute of Computational Linguistics, Peking University for proofreading the treebank. The research was supported by national natural science foundation Grant 69483003.

References

[Bla91] E. Black et al. (1991). "A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars." In *Proceedings of the February 1991 DARPA Speech and Natural language Workshop*, 306-311.

[Bri92] Eric Brill (1992). "A simple rule-based part of speech tagger". In *Proceedings, Third Conference on Applied Natural Language Processing.* Trento, Italy, 152-155.

[Bri93a] Eric Brill (1993). *A Corpus-Based Approach to Language Learning.* Ph.D. thesis, University of Pennsylvania.

[Bri93b] Eric Brill. (1993). "Automatic Grammar Induction and Parsing Free Text : A Transformation-Based Approach." In *Proc. of ACL-31*, 259-265.

[CC94] E. Charniak & G. Carroll. (1994). "Context-Sensitive Statistics For Improved Grammatical Language Models." In *Proc. of AAAI-94*, 728-733.

[Col96] Michael John Collins (1996). "A New Statistical Parser Based on Bigram Lexical Dependencies." In *Proc. of ACL-34*, 184-191.

[DM95] David M. Magerman. (1995). "Statistical Decision-Tree Models for Parsing", In *Proc. of ACL-*

95, 276-303.

[EC95] Eugene Charniak (1995). "Parsing with context-free grammars and word statistics", *Technical report CS-95-28, Department of Computer Science, Brown University.*

[HR93] D. Hindle & M. Rooth. (1993). "Structural Ambiguity and Lexical Relations", *Computational Linguistics*, 19(1), 103-120.

[LY90] K.Lari, and S.J.Young. (1990). "The estimation of stochastic context-free grammars using the Inside-Outside algorithm." *Compute Speech and Language*, 4(1), 35-56.

[MSM93] Mitchell P.Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993). "Building a Large Annotated Corpus of English: The Penn Treebank", *Computational Linguistics*, 19(2), 313-330.

[PS92] F. Pereira, and Y.Schabes. (1992). "Inside-Outside reestimation from partially bracketed Corpora." In *Proc. of ACL-30*, 128-135.

[RB93] Rens Bod. (1993). "Using an Annotated Language Corpus as a Virtual Stochastic Grammar", In *Proc. of AAAA-93, 778-783.*

[RM95] Lance A. Ramshaw & Mitchell P. Marcus (1995). "Text Chunking using Transformation-Based Learning", In *Proceedings of the third workshop on very large corpora*, 82-94.

[SYW95] H-H. Shih, S. J. Young, N.P. Waegner. (1995). "An inference approach to grammar construction", *Computer Speech and Language*, 9(3), 235-256.

[ZQ96] Zhou Qiang (1996). "A Model for Automatic Prediction of Chinese Phrase Boundary Location", *Journal of Software, Vol 7 Supplement*, 315-322.

[ZQd96] Zhou Qiang (1996). *Phrase Bracketing and Annotating on Chinese Language Corpus*, Ph.D. dissertation, Dept. of Computer Science and Technology, Peking University, June 1996.

[ZZ96] Zhou Qiang, Zhang Wei (1996). "An Improved Model for Automatic Prediction of Chinese Phrase Boundary Location", In *Proc. of ICCC'96, Singapore, June 4-7*, 75-81.