Parsing Penn Chinese Treebank Based on Lexicalized Model

Hailong Cao, Yujie Zhang, Hitoshi Isahara
Computational Linguistics Group, National Institute of Information and Communications Technology
{hlcao, yujie, isahara}@nict.go.jp

1 Introduction

Syntactic parsing is one of the most important technologies of natural language processing. The development of Penn Chinese Treebank (CTB) spurred the research of Chinese parsing. This paper describes a lexicalized statistical Chinese parser. First, a lexicalized model based on hidden Markov model is proposed for part of speech tagging. Second, a well-known lexicalized model i.e. the head-driven model is adapted to parse the automatically POS tagged Chinese sentences. The construction of the parser is described, and the effects of details that can make great difference in the parsing performance are analyzed. On sentences of length less than 100 words, the parser performances at 80.08% precision and 78.45% recall on, surpassing the best published results.

2 Lexicalized Model

Our parser takes word segmented sentences as input; formally it is a sequence with n words:

$$W = w_1, w_2, ...w_n$$

Before parsing in the sentence, we will assign each word in the sentence an appropriate part of speech tag by a lexicalized hidden Markov model (HMM).

2.1 Lexicalized Tagging Model Based on HMM

Usually there are more than one POS sequences for a given words sequence *W* since there are more than one POS tags for a single word. The statistical POS tagging method based on Bayesian model is capable of assigning a POS tagging sequence with the greatest conditional probability, which is showed as follows:

$$Tag_{best} = \arg\max_{Tag} P(Tag \mid W)$$
$$= \arg\max_{Tag} \frac{P(Tag, W)}{P(W)} = \arg\max_{Tag} P(Tag, W) \quad (1)$$

Where $Tag = t_1, t_2, ...t_n$ is a candidate POS sequence for *W*.

The classical HMM assumes that the transformation from one state (that means POS here) to another is not affected by the current observation value (that means the current word), and the generation of current observation value is independent of other observation values. That is:

$$P(Tag, W) = P(Tag)P(W \mid Tag) \approx \prod_{i=1}^{n} P(t_i \mid t_1, t_2 ....., t_{i-1}) \prod_{i=1}^{n} P(w_i \mid t_1, t_2 ....., t_n) \quad (2)$$

Furthermore, only N previous states are considered when the current state is generated. And only the current state is involved when the current word is generated:

$$P(Tag, W) = P(Tag)P(W \mid Tag) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-N}, t_{i-N+1} ..... t_{i-1}) \prod_{i=1}^{n} P(w_i \mid t_i) \quad (3)$$

This is the so-called N-order model or the (N+1)-gram model. In practice, bi-gram or tri-gram model is often used to alleviate data sparseness.

In fact, we observed there is tight association between POS tags and words in Chinese text, the above model can not reflect the characteristic of Chinese very well. In order to capture the relation between POS tags and words in Chinese text, we augment HMM by the method below:

$$Tag_{best} = \arg\max P(Tag, W) = \arg\max \prod_{i=1}^{n} P(t_i, w_i \mid t_1, w_1 ..... t_{i-1}, w_{i-1}) \approx \arg\max \prod_{i=1}^{n} P(t_i, w_i \mid t_{i-1}, w_{i-1}) \quad (4)$$

By this transformation, we have broken down the HMM's assumption, and introduced lexical information into POS tagging model to strengthen its discriminative ability.

After we introduce lexical information, data sparseness problem becomes more serious. So it is necessary to utilize some data smoothness method. From equation (4), we can get:

$$P(t_i, w_i | t_{i-1}, w_{i-1}) = P_1(t_i | t_{i-1}, w_{i-1}) P_2(w_i | t_{i-1}, w_{i-1}, t_i) \quad (5)$$

In this way, we can smooth the P_1 and P_2 in equation (5) by the following method:

$$P_1(t_i | t_{i-1}, w_{i-1}) = \lambda_1 P_{ML1}(t_i | t_{i-1}, w_{i-1}) + (1 - \lambda_1) p_{ML1}(t_i | t_{i-1}) \quad (6)$$

$$\begin{aligned} P_2(w_i | t_{i-1}, w_{i-1}, t_i) &= \lambda_{21} P_{ML2}(w_i | t_{i-1}, w_{i-1}, t_i) + \\ &(1 - \lambda_{21})[\lambda_{22}(P_{ML2}(w_i | t_{i-1}, t_i) + \\ &(1 - \lambda_{22}) P_{ML2}(w_i | t_i)] \end{aligned} \quad (7)$$

λ_1, λ_{21} and λ_{22} are smoothing parameters and $P_{ML}(x|y)$ is the empirical probability estimated from the data in the training set by using maximal likelihood estimation method:

$$P_{ML}(x | y) \equiv \frac{count(x, y)}{count(y)} \quad (8)$$

2.2 Parsing based on Collins' Model 2

The parsing model we start with is the well-known head-lexicalized model proposed by Collins[1]. Given an input sentence $S=(w_1/t_1, \ldots\ldots w_n/t_n)$ the most likely parse tree defined by a statistical generative model is:

$$T_{best} = argmax P(T | S) = argmax \frac{P(T,S)}{P(S)} = argmax P(T,S) \quad (9)$$

Probabilistic context-free grammar (PCFG) is one of the simple methods that are used to model distributions over sentence/parse-tree pairs. If there are k context free grammar ules in the parse tree, then

$$P(T,S) = \prod_{i=1..k} P(RHS_i | LHS_i) \quad (10)$$

Where LHS /RHS standards for the left/right hand side of the grammar rule.

Based on PCFG, Collins proposed a lexicalized model by associating a word w and a part of speech tag t with each non-terminal node in the parse tree. Formally, a grammar rule LHS $\rightarrow$ RHS can be written as:

$$\begin{aligned} Parent(t,w) \rightarrow\ & Lm(t,w) \ldots\ldots Lm1(t,w) \\ & H(t,w) \\ & R_1(t,w) \ldots\ldots R_n(t,w) \end{aligned}$$

Where *Parent* is the father and H is the head child, $L_m \ldots\ldots L_1$ and $R_1 \ldots\ldots R_n$ are left and right modifiers of H.

To overcome the sparseness problem due to the addition of lexical items, the generation of RHS is broken down into a Markov process that makes certain independence assumptions, and the probability of a grammar ule is defined as:

$$\begin{aligned} P(RHS | LHS) = & P_h(H | Parent(t,w)) \cdot \\ & \prod_{i=1}^{m+1} P_l(L_i(t,w) | Parent(t,w), H) \\ & \cdot \prod_{i=1}^{n+1} P_r(R_i(t,w) | Parent(t,w), H) \end{aligned} \quad (11)$$

Where L_{m+1} and R_{n+1} are stop categories. The probability P_h, P_l and P_r are estimated by maximum likelihood estimation method.

In adapting Collis' model 2 to Chinese; we also make complement/adjunct distinction in training data. We label the following three types of no-terminal as complement:

(1) NP, CP (Sub clause) or IP (simple clause) whose parent is IP.
(2) NP, CP, VP or IP whose parent is VP.
(3) IP whose parent is CP.

In addition, the no-terminal will not be labeled as complement if it is the head child of its parent. For more details such as distance measure and special preprocessing of punctuations, we refer the reader to the paper by Collins[1].

3 Experiments and Results

3.1 Data

In our experiments, both the tagging model and the parsing model are trained and tested on the Penn Chinese Treebank[2]. Following previous researches, we use article 001-270 for training, 271-300 for open testing and 301-325 for developing.

3.2 Tagging Result and Analysis

When we train the tagging model, all syntactic labels in CTB are removed and only every word and its POS tag are kept.

Table1: Evaluation tagging results

Model	Accuracy on close test set	Accuracy on open test set
Bi-gram HMM	95.98%	90.83%
Lexicalized model	99.59%	92.19%

For comparison, we use the bi-gram HMM as a baseline for the lexicalized tagging model Table 1 shows the evaluation results.

From table 1, we can see that the performance of lexicalized model outperforms bi-gram HMM significantly. We think there is still large room for improvement if more training data is available.

3.3 Parsing Result and Analysis

Before we train the parsing model, we also do the standard tree transformation such as the removal of empty nodes and semantic information in the tree bank. The head percolation table form Xia[3] is used to find heads of constituent in CTB. CYK parsing algorithm is used to decode the model in a bottom-up process. In practice, every node in the chart table is ranked according to the product of inside probability and outside probability. We explore the Viterb algorithm and the beam search strategy to improve parsing efficiency.

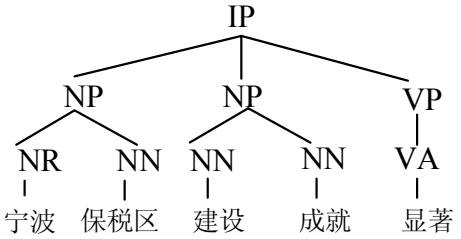

Figure 1: A sample parse tree from CTB.

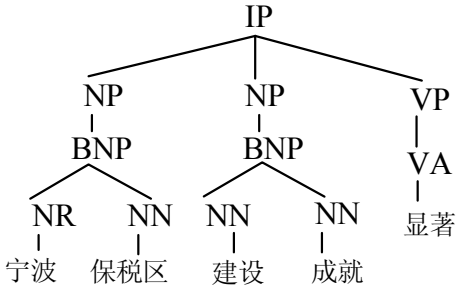

Figure 2: A sample parse tree after re-annotation.

There are two important details Collins used in his English parser.

1) One of them is the special processing of base noun phrase (BNP), i.e. the non-recursive noun phrase. Because the internal structure of base noun phrase is quite different from other kinds of noun phrase, Collins re-annotated the non-recursive noun phrase as NPB and inserted an additional noun phrase node above the NPB. For example, the parse tree shown in figure 1 will be transformed into the style illustrated in figure 2.

2) The other important detail is coordination model. There is a coordinator in the coordination construction. The model described in section 2.2 fails to learn that there is always one phrase following the coordinator. For this reason, instead of generating the coordinator and the following phrase one by one independently, they are generated together in one step.

BNP model and coordination model was originally proposed to deal with the annotation standard in the Penn English tree bank (ETB). For example, the main reason to treat BNP specially is the internal structure is left underspecified in ETB. However, the annotation standard of CTB is different from that of ETB. So it is an open question weather the special details are also effective on Chinese. In order to confirm that, we perform four experiments on development set. Table 2 shows the result.

Table 2: Result on development set with gold-standard POS tag. "Yes" means the detail is used, and "No" means it is not used.

BNP	Coordination	Precision	Recall	F1
No	No	86.41%	83.14%	84.74%
Yes	No	86.40%	85.67%	**86.04%**
No	Yes	86.12%	83.56%	84.82%
Yes	Yes	85.16%	85.34%	85.25%

It is clear that BNP model can make significant improvement. It improves the F1 from 84.74% to 86.04%. Coordination model can also make a little improvement. However, if BNP model and coordination model are utilized together, the performance is much worse than that when we only use BNP model. We leave the analysis to the future work and tentatively conclude that coordination model is not necessary when we build Chinese parser.

We then run the parser on the test set sentences that automatically tagged by our POS tagger. table 3 shows that it performances at 80.08% precision and 78.45% recall for sentences ≤ 100 words, surpassing the best published results. On sentences ≤ 40 words, we also achieve competitive parse accuracy.

4 Related Work on Parsing CTB

Much work has been done on parsing CTB and many models and approaches have been applied to CTB parsing such as BBN model, TIG, factored model, DOP method and semantic-based method. Table 3 gives some previous results. In addition[4-12], [Luo, 2003] and [Fung et.al, 2004] constructed character based parser. So their work is not directly comparable with the other parsers that operate at word-level.

Table 3: Comparison with related work on the standard test set.

	≤40 word				≤100 word			
	Recall	Precision	F1	POS	Recall	Precision	F1	POS
Bikel & Chiang 2000	76.8	77.8	77.3	--				
Chiang & Bikel 2002	78.8%	81.1%	79.9%	--	75.2%	78.0%	76.6%	--
Levy & Manning 2003	79.2%	78.4%	78.8%	--				
Bikel's Thesis 2004	78.0%	81.2%	79.6%	--	74.4%	78.5%	76.4%	--
Jiang's Thesis 2004	80.1%	82.0%	81.1%	92.4%				
Sun & Jurafsky 2004	**85.5%**	**86.4%**	**85.9%**	--				
Xiong et al. 2005	78.7%	80.1%	79.4%	--				
Wang et al. 2006	79.2%	81.1%	80.1%	92.5%	76.7%	78.4%	77.5%	92.2%
This work	79.02%	80.85%	79.93%	**92.78%**	**78.45%**	**80.08%**	**79.26%**	**92.36%**

We also note that Collins' model have already been applied to parse Chinese in several work prior to this paper. However, given the same training data and test data, the obtained results are different from each other greatly. We think the different way of utilizing the large sets of details account for most of the difference. Different POS taggers used in each work also result in different parsing accuracy to some extent.

5 Conclusion and Future Work

This article describes a lexicalized statistical Chinese parser. First, a lexicalized model based on hidden Markov model is proposed for part of speech tagging. We get a tagging accuracy of 92.19% that is significantly higher than that of HMM. Second, we adapt a well-known lexicalized model i.e. the head-driven model to parse the automatically POS tagged Chinese sentences. The construction of the parser is described, and the effects of details that can make great difference in the parsing performance are analyzed. We evaluate the parser on the standard test set, it performances at 80.08% precision and 78.45% recall on sentences of length less than 100 words, surpassing the best published results.

As for the future work, an error analysis in CTB parsing should be conducted to improve the performance of Chinese parsing system.

Reference

1 Nianwen Xue, Fei Xia, Fudong Chiou, Martha Palmer. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering, 2004.

2 Michael Collins. Head-driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania, 1999.

3 Fei Xia. Automatic Grammar Generation from Two Different Perspectives. PhD thesis, University of Pennsylvania, 1999.

4 Daniel Bikel and David Chiang. Two Statistical Parsing Models Applied to Chinese Treebank. In Proceedings of the 2nd Chinese language processing workshop, 2000.

5 David Chiang and Daniel Bikel. Recovering Latent Information in Treebanks. In Proceedings of the 19th International Conference on Computational Linguistics, 2002.

6 Roger Levy and Christopher Manning. Is it Harder to Parse Chinese, or the Chinese Treebank? In Proceedings of ACL, 2003.

7 Honglin Sun and Daniel Jurafsky. 2004. Shallow semantic parsing of Chinese. In Proceedings of the HLT/NAACL '04.

8 Zhengping Jiang. 2004. Statistical Chinese parsing. Honours thesis, National University of Singapore.

9 Daniel M. Bikel. 2004. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. thesis, University of Pennsylvania.

10 Deyi Xiong, Shuanglong Li, Qun Liu et al. Parsing the Penn Chinese Treebank with Semantic Knowledge. In Proceedings of the Second International Joint Conference Natural language processing, 2005.

11 Mengqiu Wang, Kenji Sagae and Teruko Mitamura, A Fast, Accurate Deterministic Parser for Chinese, In Proceedings of COLING/ACL '06

12 Mary Hearne and Andy Way. Data-Oriented Parsing and the Penn Chinese Treebank. In Proceedings of the First International Joint Conference Natural language processing, 2004.

13 Xiaoqiang Luo. A Maximum Entropy Chinese Character-Based Parser. In Proceedings of the conference on Empirical methods in Natural Language Processing, 2003.

14 Pascale Fung, Grace Ngai, Yongsheng Yang and Benfeng Chen. A Maximum-Entropy Chinese Parser. Augmented by Transformation-Based. Learning. ACM Transactions on Asian Language Processing, Volume 3, Issue 2, pp 159 - 168, 2004.