Empirical study on Parsing Chinese Based on Collins' Model

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

Abstract

There is a large set of details in Collins' model which has been proved to be quite important in English parsing. When we apply the model to parse other languages such as Chinese, it is very important to know whether these details will still work. We present a careful assessment of the applicability of a number of detail features in Collins' model when adapted to Chinese. We find that most of them work well for Chinese parsing; some of them do not help or even hurt the performance of Chinese parser. By combining a lexicalized tagging model and Collins' model, we present the syntactic parsing result on the newly available CTB5.1 and achieve 84.45% precision and 84.34% recall on the standard test set.

1 Introduction

The development of Penn Chinese Treebank (abbreviated as CTB, Xue et al., 2005) spurred the research of Chinese syntactic parsing. Many English parsing models have been adapted to parse CTB. Although many improvements have been achieved during the last few years, the label precision and recall of state of art of Chinese parsing is just about 80% and 82% respectively, lower than that of English. There are two obvious reasons accounting for that. First, most previous works trained their parsing models on the first version of CTB that is too small to estimate parameters reliably. Second, the models used in most previous work were originally proposed for English and were likely to be sub-optimal for Chinese.

In this paper, we address the above two issues by presenting an empirical study of adapting Collins' model(Collins, 1997; Collins, 1999) to Chinese based on the newly available CTB5.1 that is more than four times larger than the first version of CTB. There is a large set of details in Collins' model which has been proved to be quite important in English parsing. By taking advantage of the source code released by Collins and the details documented by (Bikel, 2004(a)), we provide a thorough analysis of the effect of some features and details on Chinese parsing. We found that most of them work well for Chinese parsing; some of them do not help or even hurt the performance of Chinese parser. By combining a lexicalized tagging model and Collins' model, we present the syntactic parsing result on the CTB5.1 and achieve 84.45% precision and 84.34% recall on the standard test set.

The rest of this paper is organized as follows. In section 2, we describe the part of speech (POS) tagging model which is a necessary preprocessor of the parsing model. Section 3 briefly introduces Collins' parsing model. Section 4 gives the POS tagging results. In section 5, we analyze the effect of base noun phrase model, coordination model and distance feature, examine the influence of POS tagging errors on parsing performance. Related work on CTB parsing is reviewed in section 6. Finally, conclusion is made in section 7.

2 Lexicalized POS Tagging Model Based on HMM

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

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

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

Usually there are more than one POS tag sequences for a given word sequence since there are usually 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 shown as follows:

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

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 from other observation values. That is:

$$\begin{aligned} & P(Tag,W) \\ & = P(Tag)P(W \mid Tag) \\ & \approx \prod_{i=1}^{n} P(t_i | t_1, t_2......, t_{i-1}) \prod_{i=1}^{n} P(w_i | t_1, t_2......, t_n) \end{aligned} \qquad (2)$$

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:

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

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 that there is close 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:

$$\begin{aligned} & Tag_{best} \\ & = \arg\max P(Tag,W) \\ & = \arg\max \prod_{i=1}^{n} P(t_i, w_i | t_1, w_1......t_{i-1}, w_{i-1}) \\ & \approx \arg\max \prod_{i=1}^{n} P(t_i, w_i | t_{i-1}, w_{i-1}) \end{aligned} \qquad (4)$$

By doing this transformation, we can correct the HMM's unpractical assumption introduce lexical information into POS tagging model to strengthen its discriminative ability.

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

$$\begin{aligned} & P(t_i, w_i | t_{i-1}, w_{i-1}) \\ & = P_1(t_i | t_{i-1}, w_{i-1}) P_2(w_i | t_{i-1}, w_{i-1}, t_i) \end{aligned} \qquad (5)$$

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

$$\begin{aligned} & P_1(t_i | t_{i-1}, w_{i-1}) \\ & \approx \lambda_1 P_{MLE1}(t_i | t_{i-1}, w_{i-1}) + (1-\lambda_1) p_{MLE1}(t_i | t_{i-1}) \end{aligned} \qquad (6)$$

$$\begin{aligned} & P_2(w_i | t_{i-1}, w_{i-1}, t_i) \\ & \approx \lambda_{21} P_{MLE\ 2}(w_i | t_{i-1}, w_{i-1}, t_i) + \\ & \quad (1-\lambda_{21})[\lambda_{22}(P_{MLE\ 2}(w_i | t_{i-1}, t_i) + \\ & \quad (1-\lambda_{22}) P_{MLE\ 2}(w_i | t_i)] \end{aligned} \qquad (7)$$

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

$$P_{MLE}(x \mid y) \equiv \frac{count(x, y)}{count(y)} \qquad (8)$$

3 Parsing based on Collins' Model 2

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

$$T_{best} = argmaxP(T \mid S) = argmax\frac{P(T,S)}{P(S)} = argmaxP(T,S) \qquad (9)$$

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 rules in the parse tree, then

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

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→RHS can be written as:

$$\begin{aligned} & Parent(t,w) \rightarrow L_m(t,w)...... L_1(t,w) \\ & \qquad H(t,w) \\ & \qquad R_1(t,w) ...... R_n(t,w) \end{aligned}$$

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

To overcome the sparseness problem caused by 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 rule is defined as:

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

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.

Collins' model 2 also includes subcategorization frames. So it is necessary to make complement/adjunct distinction in training data. We label the following three types of non-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 non-terminal will not be labeled as complement if it is the head child of its parent. For more details such as parameter estimation and special preprocessing of punctuations, we refer the reader to (Collins, 1999).

4 Experiments on tagging

4.1 Data

In our experiments, both the tagging model and the parsing model are trained and tested on the Penn Chinese Treebank 5.1 which contains 507,216 words, 18,782 sentences. Following the previous researches, we use the article 271-300 for testing and 301-325 for developing. All the other articles, i.e. article 001-270 and article 400-1151, are used for training.

The POS tagset of CTB has 33 tags, fewer than that of English Penn Treebank (abbreviated as ETB) which has 45 tags. This is consonant with the fact that Chinese make less use of morphology than English. For example, ETB uses 6 verb tags, namely VB, VBD, VBG, VBN, VBP and VBZ, to express the distinctions of tense and person. However, such distinctions are not necessary for Chinese.

The definitions of most of the CTB tags such as NN, CD and CC are similar to that of ETB. CTB also has several language special tags such as DEC, DEG and BA.

4.2 Result and Analysis

When we train the tagging model, all syntactic labels in CTB are removed and only words and POS tags are kept.

Table1: Tagging results

Model	Accuracy on development set	Accuracy on test set
Bi-gram HMM	91.29%	91.95%
Lexicalized model	93.27%	94.65%

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. Table 2 gives the top 10 frequent error types.

Table 2: High frequency tagging error types on development set

Error type		Count	Percentage
Gold tag	Error tag		
NN	M	11	2.40%
VA	NN	13	2.83%
JJ	NN	16	3.49%
CD	NN	20	4.36%
DEG	DEC	20	4.36%
NR	VV	20	4.36%
DEC	DEG	30	6.54%
NN	VV	31	6.75%
VV	NN	40	8.71%
NR	NN	156	33.99%

The Gold tag and Error tag in table 2 is used to express that Gold tag is mistagged as Error tag. The most frequent error occurs on NR(name entity) which is mistagged as NN(noun) for 156 times. NR words are difficult to recognize because many of them are never seen in the training data. How to deal with unknown words is a direction of future work.

NN/VV ambiguity is widespread in Chinese and is very difficult to resolve due to the lack of morphology in Chinese.

5 Experiments on parsing

5.1 Setting

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 from (Xia, 1999) is used to find the head of constituent in CTB. For convenience, all the following experiments are implemented on sentences which have no more than 40 words.

CYK parsing algorithm is used to decode the model in a bottom-up process. Beam search is used to prune search space. We set the beam width to 50,000. We use colon, comma and quotation mark as clues for comma-pruning.

From a more general view, there is another pruning strategy. Collins set a limitation for the maximum number of edges in the chart table. For a sentence to be parsed, if the number of edges generated by the decoder is more than two third of the limitation, the parsing algorithm will exit. This limitation was originally set to 200 thousand. However, if we still use this limitation in Chinese parsing, 10 out of 348 sentences from test set will be failed to get any result because of too much parse ambiguity in Chinese. So we raise the limitation from 200 thousand to 2 million to guarantee that the decoder can find a parsing tree for every sentence in the development set and test set.

5.2 The effect of BNP & coordination model

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

1) One is the special processing of base noun phrase (BNP), i.e. the non-recursive noun phrase. Because the annotation of Penn Treebank is different from what Collins model has assumed, Collins re-annotated the non-recursive noun phrase as BNP and inserted an additional noun phrase node above the BNP. For example, the parse tree shown in figure 1 will be transformed into the style illustrated in figure 2. The English translation of the example sentence would be: “In Ning-bo(宁波) bonded-area(保税区), the construction(建设) achievement(成就) is remarkable(显著) ”.

2) The other detail is the coordination model. There is a coordinator in the coordination construction. The model described in section 3 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.

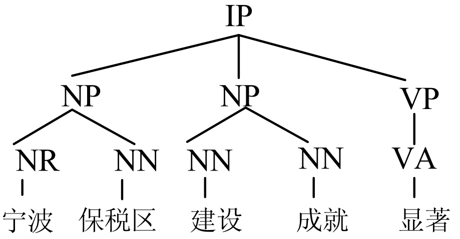

Figure 1: A sample parse tree from CTB.

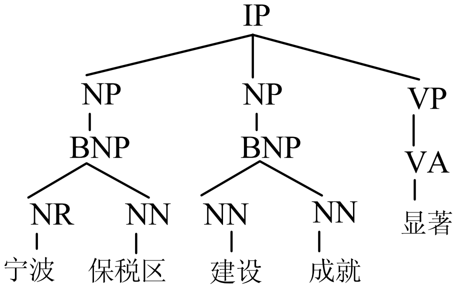

Figure 2: A sample parse tree after re-annotation.

BNP model and coordination model are originally proposed to deal with the language phenomena of English and the annotation standard in the Penn English tree bank. So it is still an open question whether the special details are also effective on Chinese which is different from English in many aspects. In order to test that, we performed four experiments on the development set. Table 3 shows the results.

Table 3: Results on development set (“Yes” means the detail is used, and “No” means it is not used.)

BNP	Coordination	Precision	Recall	F1
No	No	81.66%	81.21%	81.43%
Yes	No	82.57%	82.48%	82.53%
No	Yes	81.90%	81.27%	81.58%
Yes	Yes	82.34%	82.37%	82.36%

It is clear that BNP model can make significant improvement. It raises the F1 from 81.43% to 82.53%. Coordination model can also make a little improvement. However, if BNP model and coordination model are utilized together, the performance is worse than that when we only use BNP model.

We think one possible reason is the head modification in coordination phrase. In Collins’ model, the head of coordination is determined by two steps. First, it is picked out by the head percolation table. Second, if the head is on the right of the conjunction which is non-initial, the non-terminal on the left of the conjunction will be chosen as the new head. For example, in a grammar rule such as NP → NP_1 CC NP_2, NP2 is the initial head child of NP. But when training the coordination model, the head is modified to be NP_1. However, according to the head percolation table, Chinese noun phrase and verb phrase are usually right-headed. So the head of coordination and non-coordination is not determined in a consistent way. We suspect this is the reason why coordination model hurts the parsing performance. So rather than follow the original model, we always select the non-terminal on the right of CC as the head both on training and decoding process. The experiment results are shown in table 4.

Table 4: Results on development set with new coordination model.

BNP	Coordination	Precision	Recall	F1
No	Yes	82.07%	81.81%	81.94%
Yes	Yes	82.23%	82.39%	82.31%

In this way, we can raise the F1 from 81.58% to 81.94% when only coordination model is used. But coordination model still hurts performance when being used together with BNP model. We leave the analysis to the future work and tentatively conclude that the coordination model is not necessary when we build Chinese parser. In all the following experiments, BNP model is kept and coordination model is eliminated.

5.3 The Impact of the Distance Measure on Accuracy

It has been proved that distance measure is very important in English parsing. There are two features in the distance measure. One is the adjacency condition which indicates whether the modifier is adjacent to the head or not. The other is verb condition which indicates whether there is a verb between the modifier and the head or not. For more strict definition of distance, we refer the reader to (Collins, 1999). One important motivation of adding distance measure into the model is to capture the right-branching structures in English. However, Chinese is not a right-branching language. So it is necessary to check the impact of distance measure on Chinese parsing. Table 5 gives the results with and without the adjacency and verb distance measure.

Table 5: Results on development set with and without distance measure.

A	V	Precision	Recall	F1
No	No	78.36%	75.59%	76.95%
Yes	No	82.06%	82.17%	82.12%
Yes	Yes	82.57%	82.48%	82.53%

It is clear that adjacency condition is very effective. It raises the F1 from 76.95% to 82.12%. However, verb condition only adds 0.41% improvement. What we found in table 5 is quite different from that in English parsing. For a comparison, we list the effect of distance measure on English parsing in table 6.

Table 6: Results on section 0 of WSJ Treebank. (From (Collins, 1999))

A	V	Precision	Recall
No	No	85.1%	86.8%
Yes	No	87.7%	87.8%
Yes	Yes	88.7%	89.0%

We can get two possible clues to improve Chinese parsing by comparing the results.

First, verb condition works much more effective at English parsing than at Chinese parsing. We think the current definition of verb condition is sub-optimal for Chinese and it is quite possible to improve Chinese parsing by further tuning it.

Second, there is a more dramatic performance drop on Chinese than English when we turn off the distance measure. This difference deserves our attention because the model still includes subcategorization frame which has been shown to have similar effect with distance measure at English. So the difference implies that we have not made full use of subcategorization in Chinese parsing. In fact, the complement/adjunct distinction we make in section 3 is very simple and it is quite possible to improve Chinese parsing by further tuning it.

5.4 The influence of POS tagging errors on parsing

Because of the shortness of training data, the accuracy of our tagger is rather lower than the state-of-art English POS tagging. In order to know how much parsing errors is caused by POS tagging error, we rerun the parser on the sentences with gold standard POS tags.

In this way, we can raise the F1 from 82.53% to 86.37%. However, this result does not tell the whole story. In fact, tagging is integrated in Collins' model and every possible POS tag is considered if the word has been seen in training data. As for the output of POS tagger, the parser only accepts the POS tags of unseen words and tags other words by itself. So in the above experiment, there are still many parsing errors due to the parser fails to select the correct POS tags for some words.

To eliminate the influence of tagging errors, we change the setting of Collins' model to make sure that only the gold standard POS tag is considered. Then we have a parsing performance of 89.10%. This result is very promising. If more training data is available, we think there is large room to improve the tagging performance and therefore the parsing performance.

Finally, we run the parser on the automatically tagged sentences from the standard test set. We achieve 84.45% precision and 84.34% recall.

6 Related Work on Parsing CTB

Much work has been done on parsing CTB. Table 7 gives some previous results.

Table 7: Results from related work on the standard test set.

	Precision	Recall	F1
Bikel & Chiang 2000	77.8%	76.8%	77.3%
Chiang & Bikel 2002	81.1%	78.8%	79.9%
Levy & Manning 2003	78.4%	79.2%	78.8%
Hearne & Way, 2004	77.92%	74.46%	76.15%
Jiang's Thesis 2004	82.0%	80.1%	81.1%
Bikel's Thesis 2004(b)	81.2%	78.0%	79.6%
Xiong et al. 2005	80.1%	78.7%	79.4%
Wang et al. 2006	81.1%	79.2%	80.1%

(Bikel and Chang, 2000) presented the first result of CTB parsing based on BBN model and TIG model. (Chang and Bike, 2002) proposed an automatic method to determine the head child based on the EM algorithm. (Levy and Manning, 2003) adapted factored model to Chinese parsing and achieved much improvement by grammar transformation inspired by error analysis. (Hearne and Way, 2004) applied Data-Oriented Parsing approach to CTB. (Jiang, 2004) adapted Collins' model to Chinese parsing and made significant improvement by proposing language specific enhancements. (Bike, 2004(b)) built a multi-language parsing engine which can be extended to Chinese by the addition of tweaked head-finding rules and argument-finding heuristics. (Xiong et al., 2005) proposed a semantic-class based method by combining factored model and Collins' model. They used thesaurus to overcome the data sparseness problem. (Wang et al., 2006) presented a deterministic Chinese parser. They transform parsing into a succession of classification problem and make it convenient to apply machine learning method. In addition, (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.

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

It should be noticed that the recall is much lower than the precision in most of results shown in table in 7. We think one possible reason for that is the shortness of data since all of those parsers are trained on the first version of CTB that consists of only 4185 sentences.

7 Conclusion

There is a large set of details in Collins' model which has been proved to be quite important in English parsing. We present a thorough analysis of the effect of some important details on Chinese parsing. We find that BNP model works well for Chinese parsing while coordination model does not help or even hurts performance of Chinese parser. As for the effect of distance feature, we find that the adjacency condition is very effective in Chinese parsing while the effect of verb condition on Chinese is not as great as that on English. We also evaluate the influence of POS tagging errors on parsing and find much parsing errors are aroused by POS tagging errors. By combining a lexicalized tagging model and Collins' model, we present the syntactic parsing result on the newly available CTB5.1 and achieve 84.45% precision and 84.34% recall on standard test set.

The current model is deficient in analyzing coordination structure. In future work, more effective model should be designed.

References

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

Daniel M. Bikel. 2004(a). Intricacies of Collins' Parsing Model. *In Computational Linguistics*, 30(4): 479-511.

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

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

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

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

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

Mengqiu Wang, Kenji Sagae and Teruko Mitamura. 2006. A Fast, Accurate Deterministic Parser for Chinese. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics.*

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Conference of the Association for Computational Linguistics.*

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

Nianwen Xue, Fei Xia, Fu-Dong Chiou and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2): 207-238.

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

Roger Levy and C. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Conference of the Association for Computational Linguistics.*

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

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