Study on Architectures for Chinese POS Tagging and Parsing

Hailong Cao, Yujie Zhang and Hitoshi Isahara

3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{hlcao, yujie, isahara}@nict.go.jp

Abstract. How to deal with part of speech (POS) tagging is a very important problem when we build a syntactic parsing system. We could preprocess the text with a POS tagger before perform parsing in a pipelined approach. Alternatively, we could perform POS tagging and parsing simultaneously in an integrated approach. Few, if any, comparisons have been made on such architecture issues for Chinese parsing. This paper presents an in-depth study on this problem. According to comparison experiments, we find that integrated approach can make significantly better performance both on Chinese parsing and unknown words POS tagging than the pipelined approach. As for known words POS tagging, we find that the two approaches get similar tagging accuracy, but the tagging results of integrated approach do lead to much better parsing performance. We also analyze the reasons account for the performance difference.

1 Introduction

POS tag is an important feature in most of the parsing models as having a word's POS tag can help us determine what kind of syntactic constituent the word can compose. So usually it is necessary to assign a proper POS tag to each word in a sentence which is to be parsed. We could adopt the pipelined approach which performs parsing strictly after POS tagging, or performs POS tagging and parsing simultaneously in an integrated approach. The pipelined approach is simple and fast but is subject to error propagation. Though integrated approach can make decision from global view in theory, whether it can get better accuracy in practice is still an open question since little detailed comparison has been made between pipelined and integrated approaches for Chinese parsing.

This paper presents an in-depth study on such issues for Chinese parsing. We compare the performances of the pipelined approach, the integrated approach and two kinds of compromise strategies. There are three findings in our experiments. First, integrated approach can improve parsing performance by considering POS tag of known word globally though it can not enhance the known words tagging accuracy. Second, integrated approach can get better tagging accuracy on unknown words and therefore get better parsing result. Third, better tagging results do not always lead to better parsing results. Our comparison experiments suggest that fully integrated approach is the best strategy for Chinese parsing if complexity is not a major concern. We also analyze the reasons that account for the performance difference.

© A. Gelbukh (Ed.)
Advances in Natural Language Processing and Applications
Research in Computing Science 33, 2008, pp. 55-66

Received 11/10/07
Accepted 07/12/07
Final Version 17/01/08

2 Lexicalized POS Tagging Model Based on HMM

As the first step of our investigation, we build a separate POS tagger without considering syntactic information. Our tagger takes segmented sentences as input; formally it is a sequence with n words:

$$W = w_1, w_2, \ldots\ldots w_n$$

We assign each word in the sentence an appropriate POS 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 tag sequence with the greatest conditional probability, which is shown as follows:

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

Where $Tag = t_1, t_2, \ldots\ldots t_n$ is a candidate POS tag 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:

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

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:

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

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

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

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

2.1 Data Smoothing

Data sparseness problem is more serious after we introduce lexical information. This makes it necessary to utilize some data smoothing method. From equation (4), we get:

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

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

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

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

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:

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

2.2 Unknown Words Processing

Unknown words[1] processing is an important issue when we build POS tagger. Much work has been done on guessing the POS tag of unknown words. For convenience, the simplest algorithm is applied in our tagger. The tagger pretends that every POS tag in the tag set is a candidate for each unknown word, with equal probability. This means that the POS tag of an unknown word is predicted using lexical forms and POS tags of surrounding words without considering the lexical form of the unknown word. For more details of this algorithm, please refer to (Jurafsky and Martin, 2000).

We found that many unknown words are number words consist of Arabic numeral such as "1, 2, 3" or Chinese numeral such as "一, 二, 三". So other than pretending that every POS tag is possible, we simply tag an unknown word as CD(cardinal number) if it includes any Arabic or Chinese numeral.

2.3 Tagging Results

In our experiments, both the tagging model and the parsing model are trained and tested on the Penn Chinese Treebank 5.1(abbreviated as CTB, Xue et al., 2005) which contains 507,216 words, 18,782 sentences. We use the article 301-325 for testing. Article 001-270 and article 400-1151 are used for training. For comparison, we use the bi-gram HMM as a baseline for the lexicalized HMM tagging model. Table 1 shows the evaluation results. In all this paper, evaluation is implemented on

[1] In this paper, we define the word that does not appear in training set as unknown word.

sentences which have no more than 40 words. Among all the words in test sentences, 8.17% are unknown words.

Table 1. Evaluation of tagging results

	Accuracy on all words	Accuracy on known words	Accuracy on unknown words
Bi-gram HMM	91.79%	95.93%	45.32%
Lexicalized HMM	93.84%	97.09%	57.25%

We can see that the lexicalized HMM outperforms bi-gram HMM significantly both on known words and unknown words tagging.

In order to further confirm the performance of lexicalized HMM, we test it on English Penn tree bank which is much bigger than CTB. We use 39832 sentences from section 02-21 as training data and 2416 sentences from section 23 as test data. We get an accuracy of 96.77% which is close to state of the art of English POS tagging. So we can think 93.84% tagging accuracy on CTB is capable as a baseline in our following comparison experiments.

3 Parsing based on Collins' Model 2

The parsing model we start with is the well-known head-lexicalized model proposed by Collins (Collins, 1999). 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:

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

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

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

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 to each non-terminal node in the parse tree. Formally, a grammar rule LHS → RHS can be written as:

$$Parent(t,w) \rightarrow L_m(t,w) \ldots\ldots L_1(t,w) H(t,w) R_1(t,w) \ldots\ldots R_n(t,w)$$

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 introduction 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:

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

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.

When we adopt Collins' model to parse Chinese, the head percolation table from (Xia, 1999) is used to find the head of constituent in CTB. Collins' model 2 also includes sub-categorization frame. So it is necessary to 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 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) and (Bikel, 2004(a)).

4 Parsing Experiments on Different POS Tagging Strategies

It is necessary to assign a proper POS tag to each word in a sentence which is to be parsed. We could perform parsing strictly after POS tagging, or perform POS tagging and parsing in an integrated approach simultaneously. So in order to test which architecture is best for Chinese parsing, we perform four experiments with different setting. The first and the fourth experiments are based on pipelined approach and integrated approach respectively. The second and third are based on compromise strategy by which we mean the input sentence is tagged by a front-end tagger first and then some words will be re-tagged by the parser.

4.1 Parsing Strictly after POS Tagging

In this approach, we just input the output of our POS tagger into the parser based on Collins' model. The task of parser is to find the maximum probability tree whose terminal/non-terminal nodes are just W/Tag. Table 2 shows the experimental results which can be used as a baseline for the following experiments. For completeness, table 2 also includes the tagging performance which has been shown in table 1.

4.2 Compromise Strategy 1: Re-tag Known Word by Parser

The main question of the compromise strategy is to determine which words should be re-tagged by the parser. In Collins English parser, the input sentence is tagged by a front-end tagger.

Table 2. Tagging and parsing results

	Tagging performance	
Accuracy on all words	Accuracy on known words	Accuracy on unknown words
93.84%	97.09%	57.25%
	Parsing performance	
Precision	Recall	F1
81.84%	82.14%	81.99%

Table 3. Tagging and parsing results

	Tagging performance	
Accuracy on all words	Accuracy on known words	Accuracy on unknown words
93.94%	97.20%	57.25%
	Parsing performance	
Precision	Recall	F1
83.22%	83.11%	83.16%

Table 4. High frequency tagging error types on known words

Lexicalized HMM			Compromise strategy		
Error type		Count	Error type		Count
Gold tag	Error tag		Gold tag	Error tag	
CC	AD	5	M	NN	4
M	NN	5	NR	NN	5
JJ	NN	6	VA	NN	5
VA	NN	6	DEC	DEG	9
NN	M	7	JJ	NN	10
NN	JJ	8	NN	JJ	10
DEG	DEC	17	NN	M	10
VV	NN	17	DEG	DEC	11
NN	VV	18	VV	NN	15
DEC	DEG	27	NN	VV	18

Among the outputs of the tagger, only unknown words' POS tags are kept and known words are re-tagged by the parser. So in this section, we simply follow this strategy. For a known word, the possible POS tags are limited to those which have been seen in training data for that word. Table 3 shows the experimental results.

Comparing table 2 and table 3, we can see that tagging accuracy on unknown words is still 57.25% since the parser does not change them at all. As for known words, tagging accuracy increased from 97.09% to 97.20%. Although the increase in POS tagging is very small, a significant improvement is achieved in parsing performance. In order to find the reason why the compromise strategy obtains improvements on parsing, we analyze the tagging results in detail. We find that differences do exist between the two tagging results in section 4.1 and section 4.2. The two tagging strategies give different POS tag on 2.33% of all the known words. Table 4 shows top 10 frequent error types of two tagging results.

From table 4 we can see that the main advantage of compromise strategy is to disambiguate DEC and DEG. DEC and DEG are two possible POS tag of the auxiliary word "的" which is used frequently in Chinese text. DEC means that there will be a clause named CP before it, such as:

```
[NP [CP [IP [NP [PN 他们]]
                 [VP [VV 工作]]]
             [DEC 的]]
        [NP [NN 单位]]]
   (The place where they work.)
```

DEG means that there will be a simple phrase named DNP before it such as:

```
[NP [DNP [NP [NR 约翰]]
               [DEG 的]]
      [NP [NN 书]]]
          (John's book.)
```

We can see that the POS tag of "的" is very important to determine the syntactic structure of the words before it. So mis-tagging the word "的" will trigger much more parsing errors.

On the other hand, table 4 shows that compromise strategy makes more errors in disambiguating NN-M and NN-JJ than lexicalized HMM. But the parsing errors aroused by such tagging errors are usually limited to the local part of the mis-tagged words. As a result, compromise strategy can improve parsing performance though it can not enhance the overall known words tagging accuracy.

Inspired by the above analysis, a straightforward idea is just re-tag the word "的" with the parser and keep the POS tags of other words unchanged. Table 5 shows the experimental results. As we expected, the tagging accuracy is increased from 97.20% to 97.55%. However, the F1 score of parsing is decreased from 83.16%to 82.94%. It seems that better tagging results do not always lead to better parsing results. The reason for this interesting phenomenon is currently beyond our knowledge and further investigation is necessary.

Table 5. Tagging and parsing results

Tagging performance		
Accuracy on all words	Accuracy on known words	Accuracy on unknown words
94.26%	97.55%	57.25%
Parsing performance		
Precision	Recall	F1
82.93%	82.94%	82.94%

4.3 Compromise Strategy 2: Re-tag Unknown Words by Parser

There is no reason to believe that unknown word must be tagged by a front-end tagger. In contrast of original Collins' model, we try to re-tag unknown words by the parser and do not make any change on known word's tag assigned by lexicalized HMM. When parsing, the parser enumerates every POS tag in tag set for each unknown word. Table 6 shows the experimental results.

Table 6. Tagging and parsing results

Tagging performance		
Accuracy on all words	Accuracy on known words	Accuracy on unknown words
94.20%	97.09%	61.63%
Parsing performance		
Precision	Recall	F1
82.28%	82.33%	82.30%

Comparing table 2 with table 6, we can see that Collins' model is superior to lexicalized HMM on unknown words POS tagging because the accuracy is increased to 61.63% from 57.25%. There are two differences on tagging between Collins' model and the lexicalized HMM. First, HMM generates a word's tag conditioning on information of its previous word, while Collins' model generates a word's tag conditioning on information of the head word which the word depends on. Second, Collins' model uses higher level syntax structure information additionally. We think these two differences account for the improvement on unknown words POS tagging.

However, it should be noted that 61.63% unknown words tagging accuracy is still very low because the approach is too simple. We think there is much room for enhancing the parsing accuracy by applying more effective unknown words tagging algorithm.

4.4 Integrated approach: Tag all Words by Parser

Since both known words re-tagging and unknown words re-tagging can improve parsing performance, we continue to test how much gains we can get when all words are tagged by the parser. In this integrated approach, the input of the parser is a sequence of words without any POS information. Table 7 shows the experimental results.

Comparing table 2 and table 7, we can see the integrated approach can make significantly better performance both on POS tagging and parsing. In detail, the performance gain on tagging is mainly from unknown words while the improvement on known word tagging is trivial. With the effect of both known word re-tagging and unknown words re-tagging, the F1 score is raised from 81.99% to 83.49%.

Furthermore, we can see that known words tagging accuracy is higher than what we get in section 4.2; unknown words tagging accuracy is higher than what we get in section 4.3. This suggests that fully integrated approach is the best strategy for Chinese parsing if complexity is not a major concern.

Table 7. Tagging and parsing results

	Tagging performance	
Accuracy on all words	Accuracy on known words	Accuracy on unknown words
94.34%	97.22%	62.02%
	Parsing performance	
Precision	Recall	F1
83.64%	83.34%	83.49%

5 Related Work

Much work has been done on parsing CTB. (Bikel and Chiang, 2000) presented the first result of CTB parsing based on BBN model and TIG model. (Chiang and Bikel, 2002) proposed an automatic method to determine the head child based on the EM algorithm. (Hearne and Way, 2004) applied Data-Oriented Parsing approach to CTB. (Xiong et al., 2005) proposed a semantic-class based parsing method. (Wang et al., 2006) presented a deterministic Chinese parser. In all these work, tagging and parsing is performed in the pipelined approach.

(Bikel, 2004(b)) built a multi-language parsing engine which can be extended to Chinese. In this work, tagging and parsing is performed in a compromise approach. (Levy and Manning, 2003) applied factored model to Chinese parsing and achieved much improvement by grammar transformation. In this work, tagging and parsing is performed in an integrated approach.

The research most similar to ours is (Jiang, 2004). However, on the issue of POS tagging, Jiang's findings are quite different from ours. Jiang found if the parser retags the known words, tagging accuracy can be increased from 90.42% to 92.42%. When Jiang inputted untagged sentences into the parser, the parsing F1 score dropped from 81.1% to 78.07%.

In addition, (Luo, 2003) and (Fung et al., 2004) constructed character based parser. Luo's results showed that higher-level syntactic structures are of little use to word segmentation.

As for English parsing, how to deal with POS tagging is also an open question. (Charniak et al. 1996) investigated the use of POS taggers that output multiple tags for parsing and concluded that single taggers are preferable. However, (Watson, 2006) found that multiple-tag per word can improve on parser accuracy at the cost of efficiency. (Yoshida et al., 2007) also showed that the accuracy of parsers could be improved by allowing POS taggers to output multiple answers for some words.

6 Conclusion and Future Work

It is necessary to assign a proper POS tag to each word in a sentence which is to be parsed. We could adopt the pipelined approach which performs parsing strictly after POS tagging, or perform POS tagging and parsing simultaneously in an integrated approach. This paper presents an in-depth study on such architecture issues for Chinese parsing. There are three findings in our experiments. First, integrated approach can improve parsing performance by considering POS tag of known word globally though it can not enhance the known words tagging accuracy. Second, integrated approach can get better tagging accuracy on unknown words and therefore get better parsing result. Third, better tagging results do not always lead to better parsing results. Our comparisons suggest that fully integrated approach is the best strategy for Chinese parsing if complexity is not a major concern.

There are at least two directions for the future work. First, now the unknown words tagging accuracy in both pipelined approach and integrated approach are very low, therefore more effective unknown words tagging algorithm should be applied in future. Second, lexicalized HMM predicts a word's tag based on local information while parsing model predicts that based on long distance dependency, so simultaneously use both local and long distance feature for tagging is another direction of future work.

References

1. Daniel M. Bikel and David Chiang. 2000. Two Statistical Parsing Models Applied to Chinese Treebank. In Proceedings of the 2nd Chinese language processing workshop.
2. Daniel M. Bikel. 2004(a). Intricacies of Collins' Parsing Model. In Computational Linguistics, 30(4): 479-511.
3. Daniel M. Bikel. 2004(b). On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. thesis, University of Pennsylvania.
4. David Chiang and Daniel Bikel. 2002. Recovering Latent Information in Treebanks. In Proceedings of the 19th International Conference on Computational Linguistics.
5. 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.
6. Daniel Jurafsky, James H. Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice-Hall.
7. Eugene Charniak, Glenn Carroll, John Adcock et al. 1996. Taggers for Parsers. Artificial Intelligence, 85(1-2):45–57.
8. Fei Xia. 1999. Automatic Grammar Generation from Two Different Perspectives. PhD thesis, University of Pennsylvania.
9. Kazuhiro Yoshida, Yoshimasa Tsuruoka, Yusuke Miyao, et al. 2007. Ambiguous Part-of-Speech Tagging for Improving Accuracy and Domain Portability of Syntactic Parsers. In Proceedings of the Twentieth International Joint Conference on Artificial Intelligence.
10. 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.

11. Mengqiu Wang, Kenji Sagae and Teruko Mitamura. 2006. A Fast, Accurate Deterministic Parser for Chinese. In Proceedings of COLING/ACL.
12. Michael Collins. 1999. Head-driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.
13. Nianwen Xue, Fei Xia, Fu-Dong Chiou et al. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering, 11(2):207-238.
14. Pascale Fung, Grace Ngai, Yongsheng Yang et al. 2004. A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning. ACM Transactions on Asian Language Processing, 3(2):159-168.
15. Rebecca Watson. 2006. Part-of-speech Tagging Models for Parsing. In Proc. of CLUK.
16. Roger Levy and Christopher Manning. 2003. Is it Harder to Parse Chinese, or the Chinese Treebank? In Proceedings of ACL.
17. Xiaoqiang Luo. 2003. A Maximum Entropy Chinese Character-Based Parser. In Proceedings of the conference on Empirical methods in Natural Language Processing.
18. Zhengping Jiang. 2004. Statistical Chinese parsing. Honours thesis, National University of Singapore.