Improving Chinese Chunking by using a Large Scale Corpus

Wenliang Chen, Yujie Zhang, Hitoshi Isahara
Computational Linguistics Group
National Institute of Information and Communications Technology
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{chenwl, yujie, isahara}@nict.go.jp

Abstract

In this paper, we focus on how to use a large-scale corpus to improve word-based Chinese chunking with input that is word-segmented sentences without manual part-of-speech tags. We investigate the difference of the definitions of word and POS between the large-scale corpus and the chunking corpus. The investigation results show that the definitions do affect the performance of chunking a little. A POS tagger is trained on the large-scale corpus and is used to label the sentences in chunking corpus. Then we present new features based on new POS tags and use them for the SVM model. Trained and evaluated on Penn Chinese Treebank, our approach achieves 81.34% F_1 score, surpassing the other approaches.

1 Introduction

Chunking [1] is to identify the non-recursive cores of various types of phrases in text. CoNLL-2000 introduced a shared task to tag many kinds of phrases other than noun phrases in English [2]. Many machine learning approaches, such as Support Vector Machines (SVM), Conditional Random Fields (CRF), and Hidden Markov Models (HMM), have been applied to chunking [3, 2].

Much work has been done on Chinese chunking [4, 5]. Most previous studies supposed that the chunking systems could obtain perfect input, which has manual word segmentation and part-of-speech (POS) tags, and their systems performed very well [5].

However, the performance with manual POS tags is unrealistic since for novel text no perfect POS tags will be available. In this paper, we focus on *the task: word-based Chinese chunking with input that is word-segmented sentences without manual POS tags.* No perfect POS tags to chunking system may contain errors, which can lower the performance of chunking. For Chinese, the performance of POS tagging is lower than that of English [6], and so would lead to worse performance of Chinese chunking. This causes that it is difficult to obtain a good Chinese chunker on the chunking corpus. On the other hand, many large-scale unlabeled Chinese corpora are available, such as the PFR Corpus[1].

In this paper, we present a simple method that using a large-scale corpus to improve the performance of word-based Chinese chunking. We present the features based on new POS tags, which are different with the tags in original chunking corpus. Our experimental results reveal that our approach performs better than other approaches.

2 Chinese Chunking

2.1 Chunk Definition

We defined the same chunks as [5] and used the tool Chunklinkctb[2] developed by them to extract the Chinese chunking corpus from the Penn Chinese Treebank V5.1 (CTB)[3]. Table 1 provides definitions of these chunks.

Type	Definition
ADJP	Adjective Phrase
ADVP	Adverbial Phrase
CLP	Classifier Phrase
DNP	DEG Phrase
DP	Determiner Phrase
DVP	DEV phrase
LCP	Localizer Phrase
LST	List Marker
NP	Noun Phrase
PP	Prepositional Phrase
QP	Quantifier Phrase
VP	Verb Phrase

Table 1: Definition of Chinese chunks

2.2 Data Representation and Lexical Features

We present the data in the same way as in the CoNLL-2000 shared-task did. With data representation, the problem of Chinese chunking can be regarded as a sequence labeling task. That is to say, given a sequence of tokens (words pairing with the features), $x = \{x_1, x_2, ..., x_n\}$, we need to generate a sequence of chunk tags, $y = \{y_1, y_2, ..., y_n\}$.

We use lexical features as basic features within a fixed window. The features are listed as follows:

[1] More detailed information can be found at http://www.icl.pku.edu.

[2] The tool is available at http://www.nlplab.cn/chenwl/tools/chunklinkctb.txt.

[3] More detailed information can be found at http://www.cis.upenn.edu/ chinese/.

- a) $W_i, (i = -2, -1, 0, 1, 2)$;

Where W refers to a word, while W_0 denotes the current word and $W_n(W_{-n})$ denotes the word n positions to the right (left) of the current word.

2.3 The Chunking Model

In our approach, we apply the SVM model to incorporate our proprosed features and lexical features since the SVM model performed very well in chunking tasks [7, 5]. The SVM is a powerful supervised learning paradigm based on the Structured Risk Minimization principle from computational learning theory. Full details about the SVM model for chunking are presented in the paper [7].

Our SVM-based chunker has a second-order Markov dependency between chunk tags. In our experiments, we use all features in training and testing without feature selection.

3 New Features based on the PFR Corpus

Our approach requires that the corpus, such as the PFR corpus, has POS tags. Here, we choose the PFR corpus, which is popular in the Chinese information processing community.

3.1 A POS Tagger on the PFR Corpus

The CTB corpus is too small to train a good POS tagger, while the PFR corpus is a large-scale corpus that can be used to train a good POS tagger. This motivates us to use the PFR corpus to provide the features based on more reliable POS tags for improving the performance of chunking.

Many approaches are available for POS tagging, such as Maximum Entropy Markov Model (MEMM), CRF, and HMM. In [8], their experimental results revealed that the models achieved similar performance. To simplify, we here implement an HMM-based POS tagger.

We train an HMM-based POS tagger on the PFR corpus. To test its performance, we used the data from the first eleven months for training and the data from the last one month for testing. Without any parameter tuning, the tagger achieved 94.87% accuracy, which is much better than the accuracy (89.43%) of a tagger trained on CTB.

3.2 The Effect of the Definitions

At first glance, the idea, improving chunking on CTB by using a POS tagger trained on the PFR corpus, is not reasonable since the definitions of word and POS in the two corpora are different. Here, we would like to discuss this in more details.

As previously stated, the definitions of word are different between two corpora. A string is one word in one corpus, but may be segmented into two or more words in another one. For instance, the string “国民经济(national economy)” is one word in PFR, but two words “国民/经济” in CTB. However, this difference does not affect the chunking results very much, because chunk is a larger unit than word. For instance, “国家经济增长速度(national economy’s growth rate)” can be segmented as “国家/经济/增长/速度” or “国家经济/增长/速度”. However, for chunking, it will be tagged as “[国家经济NP][增长速度NP]” in CTB regardless of its word-segmented sequence.

The definitions of POS are also different between two corpora. They have different numbers of tags: CTB has 33 and PFR has 39. However, looking into the details of the definitions of POS, we find that in most cases some categories in one corpus are merged into one category in another corpus and few categories intersect with the others. Table 2 lists four important types of POS: noun, verb, adverb, and adjective. For example, type “noun”: “NR (Proper noun)” in CTB is equal to “nr (Person Name)/ ns (Location)/ nt (Organization)/ nz (Other Proper noun)” in PFR.

Type	CTB	PFR
noun	NR/NT/NN	n/nr/ns/nt/nz/t/s
adverb	AD	d
verb	VA/VC/VE/VV	v/vd/vn
adjective	JJ	a/ad/an

Table 2: Comparison of POS definitions

The investigations indicate that different definitions of word and POS do not affect chunking very much. Thus, it is possible to improve the performance of chunking by training a good POS tagger on a large-scale POS corpus, which is defined under another POS definition without any adaptation.

3.3 The Features based on PFR-POS

We then used this trained tagger to assign POS tags for all sentences in CTB. We call the POS tags in the PFR corpus PFR-POS tags and the POS tags in the CTB corpus CTB-POS tags. The features based on PFR-POS tags are listed as follows:

- a) $POS_i, (i = -2, -1, 0, 1, 2)$;

Where POS refers to a PFR-POS tag of a word, POS_0 denotes the PFR-POS tag of the current word, and $POS_n(POS_{-n})$ denotes the PFR-POS tag of the word n positions to the right (left) of the current word.

4 Experiments

4.1 Experimental Setting

The CTB corpus consists of 890 files. In our experiments, we used the first 838 files (FID from chtb_001.fid to chtb_1078.fid) as training data, and the remaining

System	F_1
BASIC1	75.79
BASIC2	78.18
OURS	81.34

Table 3: The results of proposed approach

52 files (FID from chtb_1100.fid to chtb_1152.fid) as testing data.

We used the package TNT [9], a very efficient statistical part-of-speech tagger, for POS tagging. We used the package YamCha (V0.33)[4] to implement the SVM model. We used all the default parameter settings of these packages.

We evaluated the results as CoNLL-2000 sharetask did. The performance of the algorithm was measured with two scores: precision P and recall R. Precision measures how many chunks found by the algorithm are correct and the recall rate contains the percentage of chunks defined in the corpus that were found by the chunking program. The two rates can be combined in one measure:

$$F_1 = \frac{2 \times P \times R}{R + P} \quad (1)$$

In this paper, we report the results with F_1 score.

4.2 Experimental Results

In the following experiments, "BASIC1" refers to the SVM model with lexical features, "BASIC2" refers to the SVM model with lexical features and CTB-POS features, and "OURS" refers to our proposed approach.

4.2.1 The Effect of new Features

Table 3 shows the experimental results. Totally, the final system provided 5.55% improvement more than BASIC1 and 3.16% more than BASIC2.

We also attempted to discover the effect of the size of unlabeled corpus. Figure 1 shows the experimental results, where x tics refers to what percentage of the PFR corpus that we used. When the percentage of the corpus we used was smaller than 0.5%, PFR-POS made a negative contribution because the performance of POS tagging was quite low. If we used more data, PFR-POS provided more reliable information for the SVM model.

4.2.2 Comparison with other Systems

There are many different Chinese chunk definitions, which are derived from different data sets [4, 10, 5]. Therefore, comparing the performance of previous studies in Chinese chunking is very difficult.

Here, we implemented other systems by ourself. The SVM and CRF have achieved state-of-the-art performance in English and Chinese chunking [7, 3, 5]. Table 4 shows the comparative results of our proposed approach with BASIC2 and CRF* (a CRF-based approach) [5]. The CRF* was implemented by the package CRF++ (V0.42)[5] and used lexical features and CTB-POS features, the same as BASIC2 used.

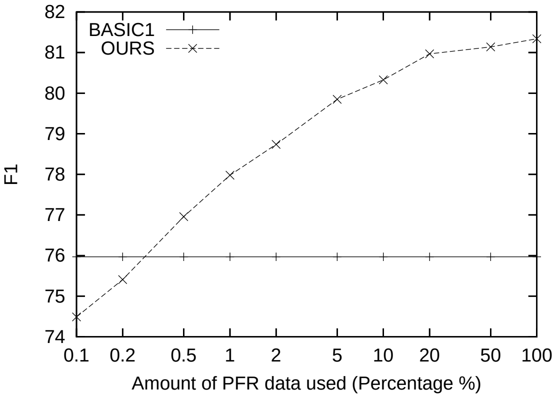

Figure 1: The effect of corpus size

We found that our proposed approach achieved the best performance among all the approaches. OURS produced 3.16% higher than BASIC2 and 4.04% higher than CRF*.

	F_1	description
OURS	81.34	Our proposed approach
BASIC2	78.18	SVM-based with CTB-POS
CRF*	77.30	CRF-based with CTB-POS

Table 4: Comparative results.

Using manual POS, we also implemented a SVM-based system, which was described in [5]. The system provided 91.64% F1 score. This suggested that we still can improve the performance of chunking by providing better POS tagging results.

4.3 Discussions

To better understand why the proposed approach performed better, we conducted the analysis by looking at the effect of POS tagging.

POS tags are very important information for chunking. However, in the CTB corpus, if we trained a POS tagger (TNT-based) on training data, the accuracy was 89.43%. This led to worse performance of chunking. To know whether our POS tagger is good, we also tested the TNT package on the standard training and testing sets for full parsing [11]. The TNT-based tagger provided 91.52% accuracy, comparative result with [11].

We investigated the performance of POS tagging on the PFR corpus with different percentages. Figure 2 shows the experimental results. If we used 1% training data of the PFR corpus, the accuracy of

[4] YamCha is available at http://chasen.org/ taku/software/yamcha/

[5] CRF++ is available at http://chasen.org/ taku/software/CRF++/

POS tagging was 89.75%, similar to the results of the POS tagger trained on CTB. When we looked at Figure 1, the performance of chunking on 1% of the PFR corpus was 78.64%, similar to the results of BASIC2. We also found that the performance of chunking increased while the accuracy of POS tagging became better. This indicated that we should pay more attention to the accuracy of POS tagging.

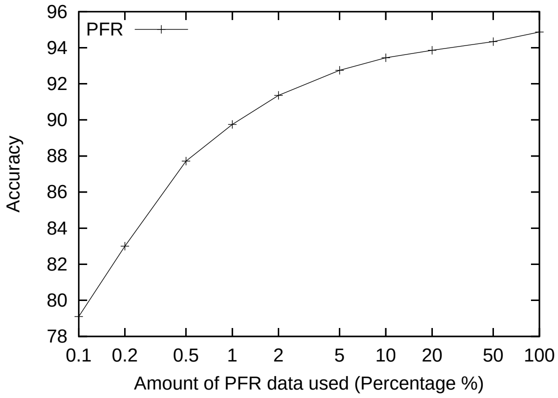

Figure 2: The results of POS tagging

5 Related Work

In the CoNLL-2000 shared-task, POS tags of the data were derived by the Brill tagger, which provided 96.6% accuracy on Penn Treebank [2]. Kudo and Matsumoto [7] applied SVM to English chunking and performed best in the shared task. Sha and Pereira [3] showed that state-of-the-art results can be achieved using CRF in English chunking.

Much work has been done on Chinese chunking [4, 10, 12, 5]. Tan et al.[12] applied SVM to Chinese chunking. They used sigmoid functions to extract probabilities from SVMs outputs as the post-processing of classification. Chen et al.[5] applied SVM, CRF, Memory-based Learning (MBL), and Transformation-based Learning (TBL) to Chinese chunking. Their experimental results revealed that the SVM model performed best among four models. In these previous studies, they supposed the input is a word-based sentence with manual POS tags.

6 Conclusions

This paper presented an approach to improve Chinese chunking by using the PFR corpus. We present the features based on the POS tags defined in the PFR corpus. And then we used new features and lexical features for the SVM model. In particular, we have achieved an absolute improvement of 5.55% over the baseline performance in F_1 score. Our experimental results also revealed that our proposed approach outperformed the other systems.

In this paper, we focus on Chinese word-based chunking. However, in real applications, the input for a Chinese chunking system is a character-based sentence. In our future work, we will try to use a large-scale corpus to improve character-based chunking.

References

[1] Steven P. Abney. Parsing by chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer, Dordrecht, 1991.

[2] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL2000*, pages 127–132, Lisbin, Portugal, 2000.

[3] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL03*, 2003.

[4] Hongqiao Li, Changning Huang, Jianfeng Gao, and Xiaozhong Fan. Chinese chunking with another type of spec. In *SIGHAN*, 2004.

[5] Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. An empirical study of chinese chunking. In *COLING/ACL 2006(Poster Sessions)*, Sydney, Australia, July 2006.

[6] Tetsuji Nakagawa and Yuji Matsumoto. Guessing parts-of-speech of unknown words using global information. In *ACL*, 2006.

[7] Taku Kudo and Yuji Matsumoto. Use of support vector learning for chunk identification. In *In Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144, 2000.

[8] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML01)*, 2001.

[9] T. Brants. TnT–a statistical part-of-speech tagger. *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231, 2000.

[10] Yuqi Zhang and Qiang Zhou. Chinese base-phrases chunking. In *Proceedings of The First SIGHAN Workshop on Chinese Language Processing*, 2002.

[11] Mengqiu Wang, Kenji Sagae, and Teruko Mitamura. A Fast, Accurate Deterministic Parser for Chinese. In *Coling-ACL2006*, 2006.

[12] Yongmei Tan, Tianshun Yao, Qing Chen, and Jingbo Zhu. Chinese chunk identification using svms plus sigmoid. In *IJCNLP*, pages 527–536, 2004.