

An Empirical Study of Chinese Chunking

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, Japan, 619-0289

{chenwl, yujie, isahara}@nict.go.jp

Abstract

In this paper, we describe an empirical study of Chinese chunking on a corpus, which is extracted from UPENN Chinese Treebank-4 (CTB4). First, we compare the performance of the state-of-the-art machine learning models. Then we propose two approaches in order to improve the performance of Chinese chunking. 1) We propose an approach to resolve the special problems of Chinese chunking. This approach extends the chunk tags for every problem by a tag-extension function. 2) We propose two novel voting methods based on the characteristics of chunking task. Compared with traditional voting methods, the proposed voting methods consider long distance information. The experimental results show that the SVMs model outperforms the other models and that our proposed approaches can improve performance significantly.

1 Introduction

Chunking identifies the non-recursive cores of various types of phrases in text, possibly as a precursor to full parsing or information extraction. Steven P. Abney was the first person to introduce chunks for parsing (Abney, 1991). Ramshaw and Marcus (Ramshaw and Marcus, 1995) first represented base noun phrase recognition as a machine learning problem. In 2000, CoNLL-2000 introduced a shared task to tag many kinds of phrases besides noun phrases in English (Sang and Buchholz, 2000). Additionally, many machine learning approaches, such as Support Vector Machines (SVMs) (Vapnik, 1995),

Conditional Random Fields (CRFs) (Lafferty et al., 2001), Memory-based Learning (MBL) (Park and Zhang, 2003), Transformation-based Learning (TBL) (Brill, 1995), and Hidden Markov Models (HMMs) (Zhou et al., 2000), have been applied to text chunking (Sang and Buchholz, 2000; Hammerston et al., 2002).

Chinese chunking is a difficult task, and much work has been done on this topic (Li et al., 2003a; Tan et al., 2005; Wu et al., 2005; Zhao et al., 2000). However, there are many different Chinese chunk definitions, which are derived from different data sets (Li et al., 2004; Zhang and Zhou, 2002). Therefore, comparing the performance of previous studies in Chinese chunking is very difficult. Furthermore, compared with the other languages, there are some special problems for Chinese chunking (Li et al., 2004).

In this paper, we extracted the chunking corpus from UPENN Chinese Treebank-4 (CTB4). We presented an empirical study of Chinese chunking on this corpus. First, we made an evaluation on the corpus to clarify the performance of state-of-the-art models in Chinese chunking. Then we proposed two approaches in order to improve the performance of Chinese chunking. 1) We proposed an approach to resolve the special problems of Chinese chunking. This approach extended the chunk tags for every problem by a tag-extension function. 2) We proposed two novel voting methods based on the characteristics of chunking task. Compared with traditional voting methods, the proposed voting methods considered long distance information. The experimental results showed the proposed approaches can improve the performance of Chinese chunking significantly.

The rest of this paper is as follows: Section 2 describes the definitions of Chinese chunks. Sec-

tion 3 simply introduces the models and features for Chinese chunking. Section 4 proposes a tag-extension method. Section 5 proposes two new voting approaches. Section 6 explains the experimental results. Finally, in section 7 we draw the conclusions.

2 Definitions of Chinese Chunks

We defined the Chinese chunks based on the CTB4 dataset¹. Many researchers have extracted the chunks from different versions of CTB(Tan et al., 2005; Li et al., 2003b). However, these studies did not provide sufficient detail. We developed a tool² to extract the corpus from CTB4 by modifying the tool Chunklink³.

2.1 Chunk Types

Here we define 12 types of chunks⁴: ADJP, ADVP, CLP, DNP, DP, DVP, LCP, LST, NP, PP, QP, VP(Xue et al., 2000). 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 Chunks

2.2 Data Representation

To represent the chunks clearly, we represent the data with an IOB-based model as the CoNLL00 shared task did, in which every word is to be tagged with a chunk type label extended with I (inside a chunk), O (outside a chunk), and B (inside a chunk, but also the first word of the chunk).

¹More detailed information at <http://www.cis.upenn.edu/~chinese/>.

²Tool is available at <http://www.nlplab.cn/chenwl/tools/chunklinkctb.txt>.

³Tool is available at <http://ilk.uvt.nl/software.html#chunklink>.

⁴There are 15 types in the Upenn Chinese TreeBank. The other chunk types are FRAG, PRN, and UCP.

Each chunk type could be extended with I or B tags. For instance, NP could be represented as two types of tags, B-NP or I-NP. Therefore, we have 25 types of chunk tags based on the IOB-based model. Every word in a sentence will be tagged with one of these chunk tags. For instance, the sentence (word segmented and Part-of-Speech tagged) "他-NR(He) /到达-VV(reached) /北京-NR(Beijing) /机场-NN(airport) /。 /" will be tagged as follows:

Example 1:

S1: [NP 他][VP 到达][NP 北京/机场][O 。]

S2: 他B-NP /到达B-VP /北京B-NP /机场I-NP /。 O /

Here S1 denotes that the sentence is tagged with chunk types, and S2 denotes that the sentence is tagged with chunk tags based on the IOB-based model.

With data representation, the problem of Chinese chunking can be regarded as a sequence tagging task. That is to say, given a sequence of tokens (words pairing with Part-of-Speech tags), $x = x_1, x_2, \dots, x_n$, we need to generate a sequence of chunk tags, $y = y_1, y_2, \dots, y_n$.

2.3 Data Set

CTB4 dataset consists of 838 files. In the experiments, we used the first 728 files (FID from chtb_001.fid to chtb_899.fid) as training data, and the other 110 files (FID from chtb_900.fid to chtb_1078.fid) as testing data. In the following sections, we use the CTB4 Corpus to refer to the extracted data set. Table 2 lists details on the CTB4 Corpus data used in this study.

	Training	Test
Num of Files	728	110
Num of Sentences	9,878	5,290
Num of Words	238,906	165,862
Num of Phrases	141,426	101,449

Table 2: Information of the CTB4 Corpus

3 Chinese Chunking

3.1 Models for Chinese Chunking

In this paper, we applied four models, including SVMs, CRFs, TBL, and MBL, which have achieved good performance in other languages. We only describe these models briefly since full details are presented elsewhere(Kudo and Matsumoto, 2001; Sha and Pereira, 2003; Ramshaw and Marcus, 1995; Sang, 2002).

3.1.1 SVMs

Support Vector Machines (SVMs) is a powerful supervised learning paradigm based on the Structured Risk Minimization principle from computational learning theory (Vapnik, 1995). Kudo and Matsumoto (Kudo and Matsumoto, 2000) applied SVMs to English chunking and achieved the best performance in the CoNLL00 shared task (Sang and Buchholz, 2000). They created 231 SVMs classifiers to predict the unique pairs of chunk tags. The final decision was given by their weighted voting. Then the label sequence was chosen using a dynamic programming algorithm. Tan et al. (Tan et al., 2004) applied SVMs to Chinese chunking. They used sigmoid functions to extract probabilities from SVMs outputs as the post-processing of classification. In this paper, we used Yamcha (V0.33)⁵ in our experiments.

3.1.2 CRFs

Conditional Random Fields is a powerful sequence labeling model (Lafferty et al., 2001) that combine the advantages of both the generative model and the classification model. Sha and Pereira (Sha and Pereira, 2003) showed that state-of-the-art results can be achieved using CRFs in English chunking. CRFs allow us to utilize a large number of observation features as well as different state sequence based features and other features we want to add. Tan et al. (Tan et al., 2005) applied CRFs to Chinese chunking and their experimental results showed that the CRFs approach provided better performance than HMM. In this paper, we used MALLET (V0.3.2)⁶ (McCallum, 2002) to implement the CRF model.

3.1.3 TBL

Transformation based learning (TBL), first introduced by Eric Brill (Brill, 1995), is mainly based on the idea of successively transforming the data in order to correct the error. The transformation rules obtained are usually few, yet powerful. TBL was applied to Chinese chunking by Li et al. (Li et al., 2004) and TBL provided good performance on their corpus. In this paper, we used fnTBL (V1.0)⁷ to implement the TBL model.

⁵Yamcha is available at <http://chasen.org/taku/software/yamcha/>

⁶MALLET is available at http://mallet.cs.umass.edu/index.php/Main_Page

⁷fnTBL is available at <http://nlp.cs.jhu.edu/rflorian/fntbl/index.html>

3.1.4 MBL

Memory-based Learning (also called instance based learning) is a non-parametric inductive learning paradigm that stores training instances in a memory structure on which predictions of new instances are based (Walter et al., 1999). The similarity between the new instance X and example Y in memory is computed using a distance metric. Tjong Kim Sang (Sang, 2002) applied memory-based learning (MBL) to English chunking. MBL performs well for a variety of shallow parsing tasks, often yielding good results. In this paper, we used TiMBL⁸ (Daelemans et al., 2004) to implement the MBL model.

3.2 Features

The observations are based on features that are able to represent the difference between the two events. We utilize both lexical and Part-Of-Speech (POS) information as the features.

We use the lexical and POS information within a fixed window. We also consider different combinations of them. The features are listed as follows:

- WORD: uni-gram and bi-grams of words in an n window.
- POS: uni-gram and bi-grams of POS in an n window.
- WORD+POS: Both the features of WORD and POS.

where n is a predefined number to denote window size.

For instance, the WORD features at the 3rd position (北京-NR) in Example 1 (set n as 2): "他_L2 到达_L1 北京_0 机场_R1 。_R2"(uni-gram) and "他_到达_LB1 到达_北京_B0 北京_机场_RB1 机场_。_RB2"(bi-gram). Thus features of WORD have 9 items (5 from uni-gram and 4 from bi-grams). In the similar way, features of POS also have 9 items and features of WORD+POS have 18 items (9+9).

4 Tag-Extension

In Chinese chunking, there are some difficult problems, which are related to Special Terms, Noun-Noun Compounds, Named Entities Tagging and Coordination. In this section, we propose an approach to resolve these problems by extending the chunk tags.

⁸TiMBL is available at <http://ilk.uvt.nl/timbl/>

In the current data representation, the chunk tags are too generic to construct accurate models. Therefore, we define a tag-extension function f_s in order to extend the chunk tags as follows:

$$T_e = f_s(T, Q) = T \cdot Q \quad (1)$$

where, T denotes the original tag set, Q denotes the problem set, and T_e denotes the extended tag set. For instance, we have an q problem($q \in Q$). Then we extend the chunk tags with q . For NP Recognition, we have two new tags: B-NP- q and I-NP- q . Here we name this approach as Tag-Extension.

In the following three cases study, we demonstrate that how to use Tag-Extension to resolve the difficult problems in NP Recognition.

1) Special Terms: this kind of noun phrases is special terms such as ” \llbracket / 生命(Life)/ 禁区(Forbidden Zone)/ \rrbracket /”, which are bracketed with the punctuation ” \llbracket , \rrbracket , \lbracket , \rbracket , \langle , \rangle ”. They are divided into two types: chunks with these punctuation and chunks without these punctuation. For instance, ” \llbracket / 生命/ 禁区/ \rrbracket /” is an NP chunk (\llbracket B-NP/ 生命I-NP/ 禁区I-NP/ \rrbracket I-NP/) while ” \llbracket /永远(forever)/ 盛开(full-blown)/ 的(DE)/ 紫荆花(Chinese Redbud)/ \rrbracket /” is tagged as (\llbracket O/ 永远O /盛开O/ 的O/ 紫荆花B-NP/ \rrbracket O/). We extend the tags with SPE for Special Terms: B-NP-SPE and I-NP-SPE.

2) Coordination: These problems are related to the conjunctions ”和(and), 与(and), 或(or), 暨(and)”. They can be divided into two types: chunks with conjunctions and chunks without conjunctions. For instance, ”香港(HongKong)/ 和(and)/ 澳门(Macau)” is an NP chunk (香港B-NP/ 和I-NP/ 澳门I-NP/), while in ”最低(least)/ 工资(salary)/ 和(and)/ 生活费(living maintenance)” it is difficult to tell whether ”最低” is a shared modifier or not, even for people. We extend the tags with COO for Coordination: B-NP-COO and I-NP-COO.

3) Named Entities Tagging: Named Entities(NE)(Sang and Meulder, 2003) are not distinguished in CTB4, and they are all tagged as ”NR”. However, they play different roles in chunks, especial in noun phrases. For instance, ”澳门-NR(Macau)/ 机场-NN(Airport)” and ”香港-NR(Hong Kong)/ 机场-NN(Airport)” vs ”邓小平-NR(Deng Xiaoping)/ 先生-NN(Mr.)” and ”宋卫平-NR(Song Weiping) 主席-NN(President)”. Here ”澳门” and ”香港” are LOCATION, while

”邓小平” and ”宋卫平” are PERSON. To investigate the effect of Named Entities, we use a LOCATION dictionary, which is generated from the PFR corpus⁹ of ICL, Peking University, to tag location words in the CTB4 Corpus. Then we extend the tags with LOC for this problem: B-NP-LOC and I-NP-LOC.

From the above cases study, we know the steps of Tag-Extension. Firstly, identifying a special problem of chunking. Secondly, extending the chunk tags via Equation (1). Finally, replacing the tags of related tokens with new chunk tags. After Tag-Extension, we use new added chunk tags to describe some special problems.

5 Voting Methods

Kudo and Matsumoto(Kudo and Matsumoto, 2001) reported that they achieved higher accuracy by applying voting of systems that were trained using different data representations. Tjong Kim Sang et al.(Sang and Buchholz, 2000) reported similar results by combining different systems.

In order to provide better results, we also apply the voting of basic systems, including SVMs, CRFs, MBL and TBL. Depending on the characteristics in the chunking task, we propose two new voting methods. In these two voting methods, we consider long distance information.

In the weighted voting method, we can assign different weights to the results of the individual system(van Halteren et al., 1998). However, it requires a larger amount of computational capacity as the training data is divided and is repeatedly used to obtain the voting weights. In this paper, we give the same weight to all basic systems in our voting methods. Suppose, we have K basic systems, the input sentence is $x = x_1, x_2, \dots, x_n$, and the results of K basic systems are $t_j = t_{1j}, t_{2j}, \dots, t_{nj}, 1 \leq j \leq K$. Then our goal is to gain a new result $y = y_1, y_2, \dots, y_n$ by voting.

5.1 Basic Voting

This is traditional voting method, which is the same as Uniform Weight in (Kudo and Matsumoto, 2001). Here we name it as Basic Voting. For each position, we have K candidates from K basic systems. After voting, we choose the candidate with the most votes as the final result for each position.

⁹More information at <http://www.icl.pku.edu>

5.2 Sent-based Voting

In this paper, we treat chunking as a sequence labeling task. Here we apply this idea in computing the votes of one sentence instead of one word. We name it as Sent-based Voting. For one sentence, we have K candidates, which are the tagged sequences produced by K basic systems. First, we vote on each position, as done in Basic Voting. Then we compute the votes of every candidate by accumulating the votes of each position. Finally, we choose the candidate with the most votes as the final result for the sentence. That is to say, we make a decision based on the votes of the whole sentence instead of each position.

5.3 Phrase-based Voting

In chunking, one phrase includes one or more words, and the word tags in one phrase depend on each other. Therefore, we propose a novel voting method based on phrases, and we compute the votes of one phrase instead of one word or one sentence. Here we name it as Phrase-based Voting.

There are two steps in the Phrase-based Voting procedure. First, we segment one sentence into pieces. Then we calculate the votes of the pieces. Table 3 is the algorithm of Phrase-based Voting, where $F(t_{ij}, t_{ik})$ is a binary function:

$$F(t_{ij}, t_{ik}) = \begin{cases} 1 & : t_{ij} = t_{ik} \\ 0 & : t_{ij} \neq t_{ik} \end{cases} \quad (2)$$

In the segmenting step, we seek the "O" or "B-XP" (XP can be replaced by any type of phrase) tags, in the results of basic systems. Then we get a new piece if all K results have the "O" or "B-XP" tags at the same position.

In the voting step, the goal is to choose a result for each piece. For each piece, we have K candidates. First, we vote on each position within the piece, as done in Basic Voting. Then we accumulate the votes of each position for every candidate. Finally, we pick the one, which has the most votes, as the final result for the piece.

The difference in these three voting methods is that we make the decisions in different ranges: Basic Voting is at one word; Phrase-based Voting is in one piece; and Sent-based Voting is in one sentence.

6 Experiments

In this section, we investigated the performance of Chinese chunking on the CTB4 Corpus.

Input: Sequence: $x = x_1, \dots, x_n$; K results: $t_j = t_{1j}, \dots, t_{nj}, 1 \leq j \leq K$. Output: Voted results: $y = y_1, y_2, \dots, y_n$
Segmenting: Segment the sentence into pieces. <pre style="margin: 0;"> Pieces[]=null; begin = 1 For each i in (2, n){ For each j in (1,K) if(t_{ij} is not "O" and "B-XP") break; if($j > K$){ add new piece: $p = x_{begin}, \dots, x_{i-1}$ into Pieces; begin = i; } }</pre>
Voting: Choose the result with the most votes for each piece: $p = x_{begin}, \dots, x_{end}$. Votes[K] = 0; For each k in (1,K)
$Votes[k] = \sum_{begin \leq i \leq end, 1 \leq j \leq K} F(t_{ij}, t_{ik}) \quad (3)$
$k_{max} = \operatorname{argmax}_{1 \leq k \leq K} (Votes[k]);$ Choose $t_{begin, k_{max}}, \dots, t_{end, k_{max}}$ as the result for piece p.

Table 3: Algorithm of Phrase-based Voting

6.1 Experimental Setting

To investigate the chunker sensitivity to the size of the training set, we generated different sizes of training sets, including 1%, 2%, 5%, 10%, 20%, 50%, and 100% of the total training data.

In our experiments, we used all the default parameter settings of the packages. Our SVMs and CRFs chunkers have a first-order Markov dependency between chunk tags.

We evaluated the results as CONLL2000 share-task 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:

$$F_1 = \frac{2 \times P \times R}{R + P} \quad (4)$$

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

6.2 Experimental Results

6.2.1 POS vs. WORD+POS

In this experiment, we compared the performance of different feature representations, in-

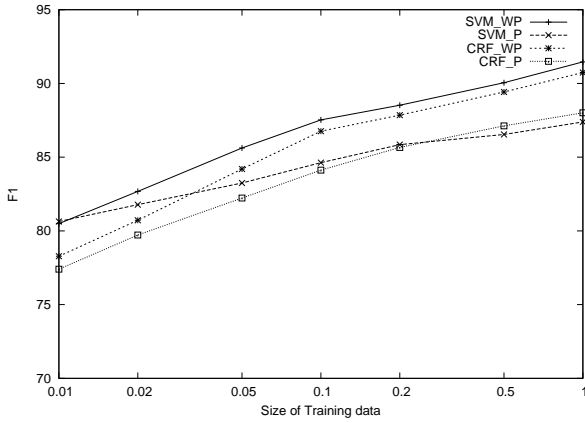


Figure 1: Results of different features

cluding POS and WORD+ POS(See section 3.2), and set the window size as 2. We also investigated the effects of different sizes of training data. The SVMs and CRFs approaches were used in the experiments because they provided good performance in chunking(Kudo and Matsumoto, 2001)(Sha and Pereira, 2003).

Figure 1 shows the experimental results, where x denotes the size of the training data, "WP" refers to WORD+POS, "P" refers to POS. We can see from the figure that WORD+POS yielded better performance than POS in the most cases. However, when the size of training data was small, the performance was similar. With WORD+POS, SVMs provided higher accuracy than CRFs in all training sizes. However, with POS, CRFs yielded better performance than SVMs in large scale training sizes. Furthermore, we found SVMs with WORD+POS provided 4.07% higher accuracy than with POS, while CRFs provided 2.73% higher accuracy.

6.2.2 Comparison of Models

In this experiment, we compared the performance of the models, including SVMs, CRFs, MBL, and TBL, in Chinese chunking. In the experiments, we used the feature WORD+POS and set the window size as 2 for the first two models. For MBL, WORD features were within a one-window size, and POS features were within a two-window size. We used the original data for TBL without any reformatting.

Table 4 shows the comparative results of the models. We found that the SVMs approach was superior to the other ones. It yielded results that were 0.72%, 1.51%, and 3.58% higher accuracy than respective CRFs, TBL, and MBL approaches.

	SVMs	CRFs	TBL	MBL
ADJP	84.45	84.55	85.95	80.48
ADV	83.12	82.74	81.98	77.95
CLP	5.26	0.00	0.00	3.70
DNP	99.65	99.64	99.65	99.61
DP	99.70	99.40	99.70	99.46
DVP	96.77	92.89	99.61	99.41
LCP	99.85	99.85	99.74	99.82
LST	68.75	68.25	56.72	64.75
NP	90.54	89.79	89.82	87.90
PP	99.67	99.66	99.67	99.59
QP	96.73	96.53	96.60	96.40
VP	89.74	88.50	85.75	82.51
+	91.46	90.74	89.95	87.88

Table 4: Comparative Results of Models

Method	Precision	Recall	F_1
CRFs	91.47	90.01	90.74
SVMs	92.03	90.91	91.46
V1	91.97	90.66	91.31
V2	92.32	90.93	91.62
V3	92.40	90.97	91.68

Table 5: Voting Results

Giving more details for each category, the SVMs approach provided the best results in ten categories, the CRFs in one category, and the TBL in five categories.

6.2.3 Comparison of Voting Methods

In this section, we compared the performance of the voting methods of four basic systems, which were used in Section 6.2.2. Table 5 shows the results of the voting systems, where V1 refers to Basic Voting, V2 refers to Sent-based Voting, and V3 refers to Phrase-based Voting. We found that Basic Voting provided slightly worse results than SVMs. However, by applying the Sent-based Voting method, we achieved higher accuracy than any single system. Furthermore, we were able to achieve more higher accuracy by applying Phrase-based Voting. Phrase-based Voting provided 0.22% and 0.94% higher accuracy than respective SVMs, CRFs approaches, the best two single systems.

The results suggested that the Phrase-based Voting method is quite suitable for chunking task. The Phrase-based Voting method considers one chunk as a voting unit instead of one word or one sentence.

	SVMs	CRFs	TBL	MBL	V3
NPR	90.62	89.72	89.89	87.77	90.92
COO	90.61	89.78	90.05	87.80	91.03
SPE	90.65	90.14	90.31	87.77	91.00
LOC	90.53	89.83	89.69	87.78	90.86
NPR*	-	-	-	-	91.13

Table 6: Results of Tag-Extension in NP Recognition

6.2.4 Tag-Extension

NP is the most important phrase in Chinese chunking and about 47% phrases in the CTB4 Corpus are NPs. In this experiment, we presented the results of Tag-Extension in NP Recognition.

Table 6 shows the experimental results of Tag-Extension, where "NPR" refers to chunking without any extension, "SPE" refers to chunking with Special Terms Tag-Extension, "COO" refers to chunking with Coordination Tag-Extension, "LOC" refers to chunking with LOCATION Tag-Extension, "NPR*" refers to voting of eight systems (four of SPE and four of COO), and "V3" refers to Phrase-based Voting method.

For NP Recognition, SVMs also yielded the best results. But it was surprised that TBL provided 0.17% higher accuracy than CRFs. By applying Phrase-based Voting, we achieved better results, 0.30% higher accuracy than SVMs.

From the table, we can see that the Tag-Extension approach can provide better results. In COO, TBL got the most improvement with 0.16%. And in SPE, TBL and CRFs got the same improvement with 0.42%. We also found that Phrase-based Voting can improve the performance significantly. NPR* provided 0.51% higher than SVMs, the best single system.

For LOC, the voting method helped to improve the performance, provided at least 0.33% higher accuracy than any single system. But we also found that CRFs and MBL provided better results while SVMs and TBL yielded worse results. The reason was that our NE tagging method was very simple. We believe NE tagging can be effective in Chinese chunking, if we use a highly accurate Named Entity Recognition system.

7 Conclusions

In this paper, we conducted an empirical study of Chinese chunking. We compared the performance of four models, SVMs, CRFs, MBL, and TBL.

We also investigated the effects of using different sizes of training data. In order to provide higher accuracy, we proposed two new voting methods according to the characteristics of the chunking task. We proposed the Tag-Extension approach to resolve the special problems of Chinese chunking by extending the chunk tags.

The experimental results showed that the SVMs model was superior to the other three models. We also found that part-of-speech tags played an important role in Chinese chunking because the gap of the performance between WORD+POS and POS was very small.

We found that the proposed voting approaches can provide higher accuracy than any single system can. In particular, the Phrase-based Voting approach is more suitable for chunking task than the other two voting approaches. Our experimental results also indicated that the Tag-Extension approach can improve the performance significantly.

References

- Steven P. Abney. 1991. 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.
- Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.
- Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. Timbl: Tilburg memory-based learner v5.1.
- James Hammerton, Miles Osborne, Susan Armstrong, and Walter Daelemans. 2002. Introduction to special issue on machine learning approaches to shallow parsing. *JMLR*, 2(3):551–558.
- Taku Kudo and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. In *In Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144.
- Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *In Proceedings of NAACL01*.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML01)*.

- Heng Li, Jonathan J. Webster, Chunyu Kit, and Tianshun Yao. 2003a. Transductive hmm based chinese text chunking. In *Proceedings of IEEE NLP-KE2003*, pages 257–262, Beijing, China.
- Sujian Li, Qun Liu, and Zhifeng Yang. 2003b. Chunking parsing with maximum entropy principle (in chinese). *Chinese Journal of Computers*, 26(12):1722–1727.
- Hongqiao Li, Changning Huang, Jianfeng Gao, and Xiaozhong Fan. 2004. Chinese chunking with another type of spec. In *The Third SIGHAN Workshop on Chinese Language Processing*.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Seong-Bae Park and Byoung-Tak Zhang. 2003. Text chunking by combining hand-crafted rules and memory-based learning. In *ACL*, pages 497–504.
- Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL2000*, pages 127–132, Lisbon, Portugal.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*.
- Erik F. Tjong Kim Sang. 2002. Memory-based shallow parsing. *JMLR*, 2(3):559–594.
- Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL03*.
- Yongmei Tan, Tianshun Yao, Qing Chen, and Jingbo Zhu. 2004. Chinese chunk identification using svms plus sigmoid. In *IJCNLP*, pages 527–536.
- Yongmei Tan, Tianshun Yao, Qing Chen, and Jingbo Zhu. 2005. Applying conditional random fields to chinese shallow parsing. In *Proceedings of CICLing-2005*, pages 167–176, Mexico City, Mexico. Springer.
- Hans van Halteren, Jakub Zavrel, and Walter Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *COLING-ACL*, pages 491–497.
- V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
- Daelemans Walter, Sabine Buchholz, and Jorn Veenstra. 1999. Memory-based shallow parsing.
- Shih-Hung Wu, Cheng-Wei Shih, Chia-Wei Wu, Tzong-Han Tsai, and Wen-Lian Hsu. 2005. Applying maximum entropy to robust chinese shallow parsing. In *Proceedings of ROCLING2005*.
- Nianwen Xue, Fei Xia, Shizhe Huang, and Anthony Kroch. 2000. The bracketing guidelines for the penn chinese treebank. Technical report, University of Pennsylvania.
- Yuqi Zhang and Qiang Zhou. 2002. Chinese base-phrases chunking. In *Proceedings of The First SIGHAN Workshop on Chinese Language Processing*.
- Tiejun Zhao, Muyun Yang, Fang Liu, Jianmin Yao, and Hao Yu. 2000. Statistics based hybrid approach to chinese base phrase identification. In *Proceedings of Second Chinese Language Processing Workshop*.
- GuoDong Zhou, Jian Su, and TongGuan Tey. 2000. Hybrid text chunking. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of the CoNLL00, Lisbon, 2000*, pages 163–165. Association for Computational Linguistics, Somerset, New Jersey.