Computational Linguistics and Chinese Language Processing
Vol. 11, No. 2, June 2006, pp. 115-136

Chinese Chunking Based on Maximum Entropy Markov Models[1]

Guang-Lu Sun[*], Chang-Ning Huang[+], Xiao-Long Wang[*], and Zhi-Ming Xu[*]

Abstract

This paper presents a new Chinese chunking method based on maximum entropy Markov models. We firstly present two types of Chinese chunking specifications and data sets, based on which the chunking models are applied. Then we describe the hidden Markov chunking model and maximum entropy chunking model. Based on our analysis of the two models, we propose a maximum entropy Markov chunking model that combines the transition probabilities and conditional probabilities of states. Experimental results for two types of data sets show that this approach achieves impressive accuracy in terms of the F-score: 91.02% and 92.68%, respectively. Compared with the hidden Markov chunking model and maximum entropy chunking model, based on the same data set, the new chunking model achieves better performance.

Keywords: Chinese Chunking, Maximum Entropy Markov Models, Chunking Specification, Feature Template, Smoothing Algorithm

1. Introduction

Text chunking is a useful step and a relatively tractable median stage in full parsing. Abney [1991] proposed to divide sentences into labeled, non-overlapping sequences of words based on superficial analysis and local information. Ramshaw and Marcus [1995] regarded chunking as a tagging problem and used a machine learning method to resolve it. A uniform standard of English chunking, including the chunking specification, data set, and evaluation method, was developed in the CoNLL-2000 shared task [Kim Sang and Buchholz 2000], which extracted

[1] The work was done while the first author was visiting Microsoft Research Asia.

[*] School of Computer Science and Technology, Harbin Institute of Technology, 150001, Harbin, China
E-mail: {glsun, wangxl, xuzm}@insun.hit.edu.cn
[+] Microsoft Research Asia, Beijing, 100080, China
E-mail: cnhuang@msrchina.research.microsoft.com

[Received May 10, 2005; Revised October 13, 2005; Accepted October 17, 2005]

chunks from the English Penn Treebank [Marcus *et al*. 1993]. Parts of the sparkle project focused on finding various sorts of chunks in English, Italian, French and German texts [Carroll *et al*. 1997]. Chunking is required by many natural language processing applications, such as information retrieval, question and answering, information extraction, and machine translation, and has been one of the most interesting problems in natural language processing.

The Chinese chunking task involves two research issues that we address in this paper. The first is the chunking specification used to define chunk types and to build a data set for supervised learning. Compared with English chunking in the CoNLL-2000 shared task, there are also several types of Chinese chunking specifications and data sets. One is extracting chunks directly from the Chinese Penn Treebank (CPTB) [Xia *et al.* 2000]. Luo [2003] and Fung [2004] regarded chunking as an intermediate step between POS tagging and full parsing, and defined chunks as the lowest non-terminal, that is, a constituent whose children are all preterminals, and they used it in statistical Chinese full parsing [Bikel and Chiang 2000; Xu 2002]. Li [2003] also provided a definition of Chinese chunks and several rules for extracting chunks from CPTB, but she did some manual checking following extraction and pruning. The others types are not based on CPTB. Zhao and Huang [1999] defined Chinese base noun phrases. Based on the inner structure of phrases, Zhou [2002] defined 9 types of Chinese base phrases. At Microsoft Research Asia (MSRA), Li and Huang [2004] defined another chunking specification for annotating all of the chunks in the open Peking University corpus [Yu *et al.* 1996]. In this paper, we select two chunking specifications and the corresponding data sets: the lowest non-terminals corpus extracted from CPTB and the annotated chunking Peking University corpus by MSRA. For the sake of brevity, the former is referred to here as the CPTB chunking specification, and the latter as the MSRA chunking specification. We use them to compare the performance of different chunking models. We select two specifications, not just one, in order to verify that our proposed model is independent of the chunking specifications. We selected these two types of corpus because they are both based on open corpora, but their chunk specifications are quite different: the former consists of rules for extracting from a tree, while the latter is a guide for annotating chunks from a segmented and POS tagged corpus.

The second research issue is chunking algorithms. Many algorithms have been applied to perform chunking. Koeling [2000] and Osborne [2000] utilized the maximum entropy model which was defined 24 feature templates. Kudoh and Matsumoto [2000] applied weighted voting of 8 support vector machines (SVM) systems trained with distinct chunk representations. Park and Zhang [2003] employed a hybrid of hand-drafted rules and a memory-based learning algorithm (MBL). Kinyon [2001] used a rule-based chunking model, which can be used to generate a robust chunking model for any language. Other algorithms have also been utilized, such as the Sparse Network of Winnows (SNoW) [Li and Roth 2001], and MBL [Bosch and Buchholz 2002]. With the CPTB and MSRA Chinese chunking specifications and data sets, we implement a chunking system based on maximum entropy Markov models (MEMM), which combine the transition probabilities and conditional probabilities of states. In open tests, we obtained F-scores of 92.68% with the CPTB data set and 91.02% with the MSRA data set; both results are better than those obtained by Li [2004] with the hidden Markov models (HMM) and maximum entropy model (MEM) under the same training and test data sets.

Section 2 describes two types of chunking specifications that were used in our experiments. Section 3 describes in detail the MEMM chunking model and compares it with the MEM chunking model and HMM chunking model. Section 4 presents experimental results obtained with our system, based on two types of chunking data sets. Finally, we draw some conclusions.

2. Chinese Chunking Specification

For the sake of comparing the results of different chunking models, two types of chunking specifications and data sets mentioned in Section 1 are defined below.

The following constraints that guarantee feasible consistency and make chunks more applicable are obeyed in both chunking specifications.

1) No chunk can destroy phrase structures. In particular, object-predicate and verb-argument structures cannot be included in one chunk.
2) Any phrase composed of chunks has a flat structure. Neither the relations between chunks nor the words' relations in chunks are divided.

2.1 CPTB Chunking Specification

Guided by Luo's [2003] definition of chunks, we define a chunk as a constituent whose children are all preterminals. Twenty-three types of chunks can be extracted directly from CPTB without performing any pre- and post extraction process. Table 1 shows the tag of each chunk type in the CPTB specification. The tags and tag descriptions are the same as those for CPTB syntactic tags [Xue and Xia 2000].

Table 1. The tag of each chunk type in the CPTB specification

Chunk tag			
ADJP	*ADVP*	*CLP*	*CP*
DNP	*DP*	*DVP*	*FRAG*
IP	*LCP*	*LST*	*NP*
PP	*PRN*	*QP*	*UCP*
VP	*VCD*	*VCP*	*VNV*
VPT	*VRD*	*VSB*	

In order to identify the boundaries of each chunk in sentences, we define two boundary types, which are denoted by *B* and *I*. Let *B* be the beginning of a chunk, and let *I* be the interior of a chunk.

To sum up, combining chunk types with boundary types, the CPTB specification contains forty-six tags. The following is an example tagged based on the CPTB specification:

Example 1

布朗/B-NP (Brown) 表示/B-VP (denoted)，/I-VP 双方/B-NP (two parties) 可以/B-VP (can) 在/B-PP(in) 运输/B-NP(transportation) 、/I-NP 电讯/I-NP (telecommunication) 、/I-NP 发电/I-NP(generate electricity) 、/I-NP 金融/I-NP(finance) 服务业/I-NP(service) 等/I-NP(etc.) 方面/B-NP(aspect) 取得/B-VP(acquire) 进一步/B-ADJP(more) 的/B-DNP(of) 合作/B-NP(cooperation)。/B-IP

(Brown indicated that the two parties can improve cooperation in terms of transportation, telecommunications, electric power, finance, services, etc..)

With this specification, the CPTB chunking data set can be automatically extracted from CPTB.

2.2 MSRA Chunking Specification

Guided by the CoNLL-2000 English chunking specification and the characteristics of Chinese, eleven chunk types are defined in the MSRA chunking specification. Table 2 shows the tag, description and examples for each chunk type.

Table 2. The tag, description and examples for each chunk type in the MSRA chunking specification

Chunk tag	Chunk description	Examples
NP	*Noun chunk*	*[NP 风雨/n (wind and rain) 电闪/n (lightning)], [NP 13 亿/m (1.3 billion) 中国/n (Chinese) 人/n (people)]*
VP	*Verb chunk*	*[VP 迷/v (lose) 了/u 路/n (one's way)], [VP 总/d (always) 也/d (also) 忘/v (forget) 不/d (never) 了/u]*
ADJP	*Adjective chunk*	*[ADJP 最为/d (the most) 出色/a (excellent)], [ADJP 勇敢/a (courageous)]*
ADVP	*Adverb chunk*	*[ADVP 无愧/v (with a clear conscience) 地/u], [ADVP 也/d (also) 早已/d (for a long time)]*
PP	*Prepositional chunk*	*[PP 从/p (from) 柜子/n (cupboard) 里/f (in)], [PP 自/p (since) 1997 年/t (1997) 7 月/t (July) 1 日/t (1st) 以来/f]*

MP	*Numerical chunk*	*[MP 数/m (several) 千/m (thousand) 余/m (about) 件/q (piece)], [MP 十/m (ten) 次/q (time)]*
TP	*Temporal chunk*	*[TP 最近/t (recently)], [TP 1998 年/t (1998) 10 月/t (October) 1 日/t (1st)]*
SP	*Spatial chunk*	*[SP 建国/v (the foundation of the state) 以来/f (after)], [SP 最后/f (finally)]*
CONJP	*Conjunction chunk*	*[CONJP 而是/c (while)], [CONJP 但/c (but) 总的说来/c (generally speaking)]*
INTJP	*Interjection chunk*	*[INTJP 吗/y], [INTJP 了/y 吧/y]*
INDP	*Independent chunk*	*[INDP 新华社/n (Xinhua News Agency) 北京/n (Beijing) 1 月/t (January) 19 日/t (19th) 电/n (dispatch)]*

In order to identify the boundaries of each chunk in sentences, we define four boundary types, which are denoted by *B, I, E, S*. Let *B* be the beginning of a chunk, let *I* be the interior of a chunk, let *E* be the ending of a chunk and let *S* be a single word chunk.

Besides the above types, some special function words (*'的/of', '和/and', '与/and', '或/or'*) in Chinese cannot be divided into any chunk types. We use *O* to tag these words and the punctuations as outside of any chunks.

To sum up, combining chunk types with boundary types, the MSRA specification contains forty-five tags plus *O*. The following is an example tagged based on the MSRA specification:

Example 2

中央/B-NP (central) 电视台/E-NP (television) 得到/S-VP (receive) 一/B-MP (a) 批/E-MP (passel) 思想性/S-NP (ideological nature) 强/S-ADJP (strong) 、/O 艺术性/S-NP (artistic quality) 高/S-ADJP (high) 的/O 好/B-NP (excellent) 作品/E-NP (work) ，/O 其中/S-NP (thereinto) 已/B-VP (already) 有/E-VP (have) 八/B-NP (eight) 部/I-NP (measure word) 作品/E-NP (work) 开始/S-VP (start) 作/S-VP (do) 投拍/S-NP (put to shot) 的/O 准备/S-NP (preparation) 。/O

(Central Television has received a passel of excellent works of strong ideological nature and high artistic quality, of which eight have being prepared to put to shot.)

With this specification, all the chunks can be manually annotated in the Peking University corpus which has been segmented and tagged with POS tag manually.

3. Chunking Model[2]

Through the use of the chunk tags described in Section 2, the Chinese chunking problem can be abstracted as a classification problem. Below, we briefly introduce the HMM chunking model and MEM chunking model, and discuss these models' limitations. To overcome these limitations, we propose the MEMM chunking model and describe it in detail.

3.1 HMM for Chunking

HMM is a statistical structure with stochastic transitions and observations [Rabiner 1989]. It can be used to solve classification problems involved in modeling sequential data. Li [2004] proposed the Chinese chunking model based on conventional HMM.

Given a word sequence $W = w_1, w_2, \ldots, w_k$ and its POS sequence $T = t_1, t_2, \ldots, t_k$, where k is the number of words in the sentence, the result of chunking is assumed to be a sequence, in which the words are grouped into chunks as follows:

$$\ldots [w_i\ w_{i+1} \ldots w_{i+m}]\ [w_{i+m+1}\ w_{i+m+2} \ldots w_{i+m+h}] \ldots$$

The corresponding POS tag sequence is grouped as follows:

$$C = \ldots \underbrace{[t_i\ t_{i+1} \ldots t_{i+m}]}_{c_j}\ \underbrace{[t_{i+m+1}\ t_{i+m+2} \ldots t_{i+m+h}]}_{c_{j+1}} \ldots$$

Here c_j corresponds to the POS tag sequence of a chunk. $[t_i\ t_{i+1} \ldots t_{i+m}] \rightarrow c_j$ may also be thought of as a chunk rule. Therefore, C is a sequence of eleven possible chunk rules and some outside words, which we refer to as O. The chunking task is, thus, converted to that of finding a rule sequence. According to Bayes' rule, it can be computed as follows [Xun *et al.* 2000]:

$$\begin{aligned} C^* &= \arg\max_c P(C/W,T) \\ &= \arg\max_c P(W/C,T)P(C,T) \, . \\ &= \arg\max_c P(W/C,T)P(C) \end{aligned} \tag{1}$$

Here, $P(C)$ is the probability of transition. It is seen as the rule's n-gram model. A tri-gram among chunks are used to approximate

---

[2] In Section 3, MSRA chunking specification and tags are used to illustrate in the chunking models.

$$P(C) \approx P(c_1)P(c_2 / c_1)\prod_{i=3}^{k} P(c_i / c_{i-1}, c_{i-2}) . \tag{2}$$

Smoothing follows application of the method proposed by Gao *et al*. [2002].

$P(W / C,T)$ is the probability of emission. The employed independent assumption is that the current word w_i is related to the current POS tag t_i, the current word's boundary type m_i (including *B*, *I*, *E*, *S*, and *O*), and the current word's chunk type x_i (including eleven types of chunks). It is approximated as follows:

$$P(W / C,T) = \prod_{i=1}^{m} P(w_i / t_i, m_i, x_i) . \tag{3}$$

If the triple (w_i, t_i, m_i, x_i) is unseen, formula (4) is used:

$$P(w_i / t_i, m_i, x_i) = \frac{count(t_i, m_i, x_i)}{\max_{j,k}(count(t_i, m_j, x_k))^2} , \tag{4}$$

where $count(t_i, m_i, x_i)$ is the frequency when the triple (t_i, m_i, x_i) occurs.

There are three problems with the HMM chunking model. Firstly, HMM is a generative model focusing on the joint probability of states and observations. But the chunking problem is a conditional probability problem when observations are given. Secondly, independent assumption of HMM makes the current observation relevant to the current state and irrelevant to the context observation; however, context words should have an impact on chunking. Thirdly, many representations give the observation a particular description by means of overlapping features that are not independent of each other. These representations cannot be used in HMM.

3.2 MEM for Chunking

As an alternative to HMM, MEM is proposed to solve the chunking problem. MEM is an exponential model that offers the flexibility of integrating multiple sources of knowledge into a model [Berger 1996]. One of the main advantages of using MEM is the ability to incorporate various features into the conditional probability framework. Furthermore, the conditional probability model focuses on the modeling of tagging sequence, replacing the modeling of observation sequence.

Let *H* denote the histories that consist of *W* and *T*. Given *H*, the goal of MEM is to find the optimal chunk tag sequence $S = s_1, s_2, \ldots , s_k$ that contains forty-five chunk tags. The model decomposes $P(S / H)$ into the product of probabilities of individual chunk actions $P(s_i / H_i)$. H_i represents the histories of s_i.

The conditional entropy of a distribution $P(s/h)$ is defined as

$$H(p) = -\sum_{s\in S,h\in H} \tilde{p}(h)p(s\,|\,h)\log p(s\,|\,h)\,. \tag{5}$$

By maximizing the conditional entropy subject to certain constraints, we can estimate $P(s/h)$ based on the maximum entropy theory [Ratnaparkhi 1996]. The constraints are defined as follows:

$$P = \{p\,|\,E_p f_j = E_{\tilde{p}} f_j, \forall f_j\}\,, \tag{6}$$

$$\sum_s p(s\,|\,h) = 1\,, \tag{7}$$

where f_j is the feature function of MEM. $E_p f_j$ is the model's expectation of f_j. $E_{\tilde{p}} f_j$ is the empirical expectation of f_j. They are defined as follows:

$$f_j(s,h) = \begin{cases} 1 & if \quad h_j = h^* \quad and \quad s = s^* \\ 0 & otherwise \end{cases}, \tag{8}$$

$$E_p f_j = \sum_{s,h} \tilde{p}(h)p(s\,|\,h)f_j(s,h)\,, \tag{9}$$

$$E_{\tilde{p}} f_j = \sum_{s,h} \tilde{p}(s,h)f_j(s,h)\,. \tag{10}$$

Let s^* be a certain chunk tag, and let h^* be a certain instance of context. The model's distribution $P(s/h)$ can be inferred by means of Lagrange transformation:

$$p(s\,|\,h) = \frac{1}{Z(h)}\exp\left(\sum_j \lambda_j f_j(s,h)\right), \tag{11}$$

$$Z(h) = \sum_s \exp\left(\sum_j \lambda_j f_j(s,h)\right), \tag{12}$$

where $Z(h)$ is the normalization constant. λ_i is the multiplier parameter with respect to each feature function.

Given a set of features and a corpus of training data, the Improved Iterative Scaling algorithm [Della Pietra 1997] can be used to find the optimal parameters { λ_i }.

3.3 MEMM for Chunking

MEM, which combines independent and dependent overlapping features together to predict chunk tags, can overcome the deficiency of HMM mentioned above. However, it does not apply the relations between each tags because MEM labels each word separately without considering the probability of neighboring chunk tag transition. For chunking, the neighboring tags are dependent; for example the chunk tag next to B-NP should be I-NP or E-NP. To overcome this shortcoming, MEMM has been proposed. In it, the current state s_i depends not only on the previous state s_{i-1} but also on the observation sequence O, as shown in Figure 1 [McCallum 2000].

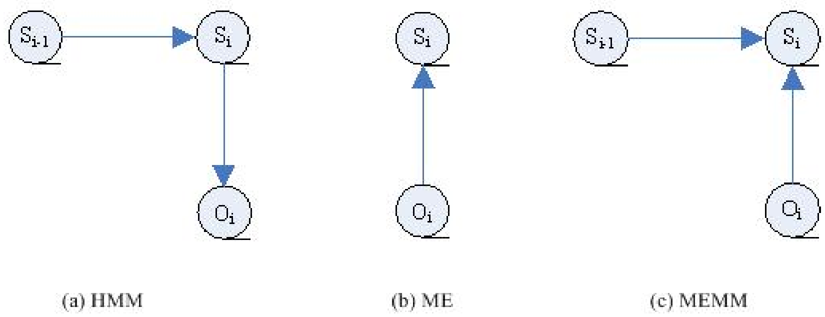

Figure 1. The dependency relation for HMM, MEM, and MEMM

MEMM combines the emission probability and transition probability of HMM into a unified function, $P(s_i \mid s_{i-1}, O)$, where s_i is a chunk tag and O consists of W and T. McCallum [2000] proposed an algorithm to solve the unified function. As the previous state s_{i-1} is assigned to a certain s^*, $P(s_i \mid s_{i-1}, O)$ is divided into $|S|$ separately trained functions, $P_{s*}(s_i \mid O)$, where $|S|$ is the size of the state space. Each separate function is trained using an exponential model. Thus, the number of states increases, and the data sparseness problem becomes more serious. Because there are forty-five types of chunk tags and some tags occur rarely in training data, it is hard to build forty-five separate, conformable exponential models.

As a possible solution, a simplified method can be used to solve the unified function $P(s_i \mid s_{i-1}, O)$. We split $P(s_i \mid s_{i-1}, O)$ into two functions in order to reduce the complexity of the model. $P(s_i \mid s_{i-1}, O)$ is estimated as follows:

$$P(s_i \mid s_{i-1}, O) = P(s_i \mid s_{i-1}) P(s_i \mid H_i), \tag{13}$$

where $P(s_i \mid H_i)$ is the conditional probability of a state. Let H_i be histories of s_i. The previous state s_{i-1} is seen as one of the histories in MEM, just like the representations of the observation sequence O. With this method, forty-five separate exponential models are replaced with one exponential model. Meanwhile, MEM, described in Section 3.2, is used to estimate $P(s_i \mid H_i)$.

$P(s_i \mid s_{i-1})$ is the transition probability of a state. Because only some chunk tag pairs occur in the training data, a smoothing algorithm is needed to solve the data sparseness problem of the tag bi-gram. Since not all chunk tags can be followed between each other, three transition restricted rules are used to reduce the number of tag pairs. This can make smoothing more reliable. Let *X* be a certain chunk type, and let *Y* be a random chunk type. *B*, *I*, *E*, *S*, and *O* were defined in Section 2.2. Thus:

1) *B-X* can be followed by *I-X* or *E-X*;
2) *I-X* can be followed by *I-X* or *E-X*;
3) *E-X*, *S-X*, and *O* can be followed by *B-Y*, *S-Y*, or *O*.

Through three rules, five hundred and seventy-three types of tag pairs can be enumerated. Interpolation smoothing is used, and $P(s_i \mid s_{i-1})$ is estimated as follows:

$$P(s_i \mid s_{i-1}) = \lambda * P'(s_i \mid s_{i-1}) + (1-\lambda) * P(s_i)\,. \tag{14}$$

Maximum Likelihood Estimation (MLE) is used to estimate the empirical probability $P'(s_i \mid s_{i-1})$ and the tag unigram $P(s_i)$. We set the empirical value λ to 0.7 in the MSRA data set.

Finally, $P(s_i \mid s_{i-1}, O)$ can be estimated by means of $P(s_i \mid H_i)$ and $P(s_i \mid s_{i-1})$. If H_i includes the previous state s_{i-1}, then $P(s_i \mid H_i)$ and $Z(h)$ vary as the previous state s_{i-1} changes in $P(s_i \mid s_{i-1})$. By means of this method, $P(s_i \mid H_i)$ and $P(s_i \mid s_{i-1})$ can be combined dynamically. The Viterbi algorithm is used to search for the optimal sequence of states. Figure 2 shows the structure of the Chinese chunking model based on MEMM.

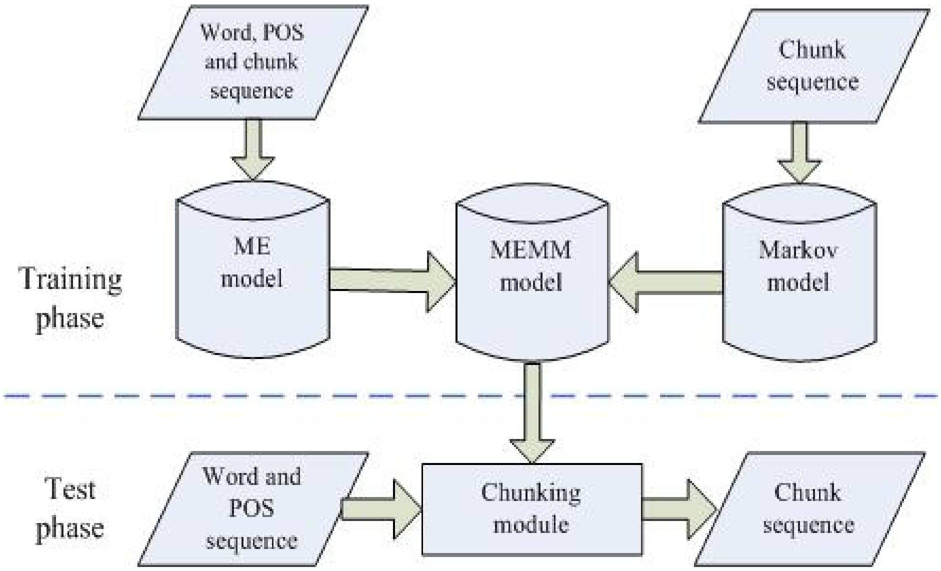

Figure 2. The structure of the MEMM Chinese chunking model

3.4 Features in MEMM and MEM

MEM and MEMM are both highly dependent on feature templates. For the sake of making a fair comparison between MEM and MEMM, both MEM and MEMM use the same feature template. The histories of the current state are a source for feature collection. The lexical and POS information of the current word, the left context consisting of two words, and the right context consisting of two words are regarded as histories. In addition, the affix information of the current word and the chunk tag of the previous word are atomic features [Ratnaparkhi 1996; Koeling 2000]. Table 3 shows the atomic features.

Table 3. Atomic features in MEMM and MEM

Feature tag	Feature explanation
W_i	*Current word*
W_{i-1}	*The previous word*
W_{i-2}	*The previous but one word*
W_{i+1}	*The next word*
W_{i+2}	*The next but one word*
P_i	*Current POS tag*
P_{i-1}	*POS tag of the previous word*
P_{i-2}	*POS tag of the previous but one word*
P_{i+1}	*POS tag of the next word*
P_{i+2}	*POS tag of the next but one word*
S_{i-1}	*Chunk tag of the previous word*
PF_i	*Two-character prefix of the current word*
AF_i	*Two-character suffix of the current word*

In order to compare the effectiveness of different types of features, we selected three types of feature templates. Table 4 shows the template based on lexical information only. Table 5 shows the template based on POS information only. Table 6 shows the template based on both lexical and POS information. Results obtained using different feature templates will be given in Section 4.

The heuristic that low frequency features are not reliable was used to cut off the features that occurred less than three times. Through feature selection, more reliable features could be used.

Table 4. Feature template based on lexical information

Feature type	Features
Atomic features	W_i, W_{i-1}, W_{i-2}, W_{i+1}, W_{i+2}, S_{i-1}, PF_i, AF_i
Combined features	$W_{i-1}W_i$, $W_{i-2}W_{i-1}$, W_iW_{i+1}, $W_{i+1}W_{i+2}$, $W_{i-1}W_{i+1}$, $W_{i-1}W_iW_{i+1}$, $W_{i-2}W_{i-1}W_i$, $W_iW_{i+1}W_{i+2}$,

Table 5. Feature template based on POS information

Feature type	Features
Atomic features	P_i, P_{i-1}, P_{i-2}, P_{i+1}, P_{i+2}, S_{i-1}
Combined features	$P_{i-1}P_i$, $P_{i-2}P_{i-1}$, P_iP_{i+1}, $P_{i+1}P_{i+2}$, $P_{i-1}P_{i+1}$, $P_{i-1}P_iP_{i+1}$, $P_{i-2}P_{i-1}P_i$, $P_iP_{i+1}P_{i+2}$,

Table 6. Feature template based on both lexical and POS information

Feature type	Features
Atomic features	W_i, W_{i-1}, W_{i-2}, W_{i+1}, W_{i+2}, P_i, P_{i-1}, P_{i-2}, P_{i+1}, P_{i+2}, S_{i-1}, PF_i, AF_i
Combined features	$W_{i-1}W_i$, W_iW_{i+1}, $W_{i-1}W_{i+1}$, $P_{i-1}P_i$, $P_{i-2}P_{i-1}$, P_iP_{i+1}, $P_{i-1}P_{i+1}$, $P_{i-1}P_iP_{i+1}$, $P_{i-2}P_{i-1}P_i$, $P_iP_{i+1}P_{i+2}$, W_iP_{i+1}, W_iP_{i+2}, P_iW_{i-1}, W_{i-2} $P_{i-1}P_i$, $P_iW_{i+1}P_{i+1}$, $P_{i-1}W_iP_i$, $S_{i-1}P_iP_{i+1}$, $S_{i-1}P_i$, $S_{i-1}P_{i-1}P_i$, P_iW_{i+1},

4. Evaluation and Discussion

We will firstly describe in detail our Chinese chunking data set. Then we will present the chunking performance and discuss it.

4.1 Data Set

The CPTB chunking data set is based on data automatically extracted from CPTB, which has a total of around 100,000 word tokens. Following Bikel's [2000] division, sections 001-270 (approximately 90% of the CPTB) were used for training, and sections 271-300 (approximately 10%) for testing. The remaining sections (301-325) were held for later development/tuning purposes. The CPTB chunking data set consisted of 3,822 sentences with 74,587 chunks and 92,729 word tokens. Thirty-one types of POS tags and forty-one types of chunk tags occurred in the data set. The average length (AL) of the chunks is 1.243 word tokens. Table 7 shows details of the training and test data sets.

Table 7. CPTB chunking training and test data sets

Data set	Number of sentences	Number of chunks	Number of word tokens
Training	*3474*	*68162*	*84749*
Test	*348*	*6425*	*7980*

The MSRA chunking data set is based on the Peking University corpus, which has been segmented, POS tagged, and chunk annotated manually. The data set consisted of 18,239 sentences with 243,868 chunks and 473,179 word tokens. The vocabulary size was 34,793. Forty-two types of POS tags and forty-three types of chunk tags occurred in the data set. The AL of the chunks is 1.377 word tokens[3]. Table 8 shows details of the training and test data sets. Table 9 shows the distribution of each type of chunk in the data set.

Table 8. MSRA chunking training and test data sets

Data set	Number of sentences	Number of chunks	Number of word tokens	Number of *O*
Training	*17,253*	*229,989*	*444,777*	*92,839*
Test	*986*	*13,879*	*28,382*	*5,493*

Table 9. The distribution of each type of MSRA chunk

Chunk type	AL	Percentage (%)
NP	*1.649*	*45.94*
VP	*1.416*	*29.82*
PP	*1.221*	*6.59*
MP	*1.818*	*3.69*
ADJP	*1.308*	*3.77*
SP	*1.167*	*2.71*
TP	*1.251*	*2.59*
CONJP	*1.000*	*2.22*
INDP	*4.297*	*1.41*
ADVP	*1.117*	*1.06*
INTJP	*1.016*	*0.23*
ALL	*1.507*	*100*

4.2 Experimental Results

Following the measurement approach adopted in CoNLL-2000, we measured the performance of Chinese chunking in terms of the precision (P), recall (R), and F-score (F). All the results were obtained in open tests.

---

[3] The AL of chunks includes the length of *O*. Without *O*, the AL is 1.507 word tokens.

For the CPTB chunking data set, the results are listed in Table 10. The results for HMM [Li 2004] are listed in the first row of Table 10. The second and third rows list the results for MEM and MEMM, respectively, where the same feature template defined in Table 6 was used. The empirical value λ mentioned in Section 3.3 was set to 0.65, based on the training data. It can be seen that, MEMM achieved the best results on the CPTB chunking data set.

Table 10. Chunking performance achieved by applying different systems to the CPTB data set

Model	P(%)	R(%)	F (%)
HMM	*89.07*	*90.82*	*89.94*
MEM	*92.33*	*90.93*	*91.62*
MEMM Lexical and POS features	*93.20*	*92.17*	*92.68*

In order to test the feature impact on MEMM, we tested MEMM chunking on the CPTB data set with the different types of feature templates described in Section 3.4. Table 11 shows the results. The chunk tag that had maximum occurrence probability for each word token was used to chunk its corresponding token. With this method, we got the baseline results listed in the first row of Table 11. The results obtained using the feature template in Table 4 are listed in the second row of Table 11, and then the third and fourth row is for Table 5 and Table 6. It can be seen that, the performance achieved using POS information only is much better than the performance achieved using lexical information only. The performance achieved using lexical and POS information is much better than the performance achieved using POS information only.

Table 11. MEMM chunking performance achieved by applying different feature templates to the CPTB data set

Model	P(%)	R(%)	F (%)
Baseline	*59.22*	*65.76*	*62.32*
MEMM Lexical features	*74.45*	*72.05*	*73.23*
MEMM POS features	*88.92*	*87.80*	*88.35*
MEMM Lexical and POS features	*93.20*	*92.17*	*92.68*

Table 12 shows the performance of different chunk types for the CPTB chunking data set when the total MEMM F-score in total was 92.68%. As shown, some chunk types achieved much poorer performance, such as *PRN*, *UCP*, *VNV*, and *VSB*. The reason was that they rarely occurred in the training data set, so it was difficult to tag them correctly. NP was the most frequent chunk type, but its performance was much poorer than the average performance. The reason is that the boundary of NP is difficult to distinguish.

Table 12. The performance of each chunk type for the CPTB data set

Chunk type	P (%)	R (%)	F (%)
ADJP	*97.03*	*98.86*	*97.94*
ADVP	*99.40*	*99.70*	*99.55*
CLP	*99.26*	*99.26*	*99.26*
CP	*98.05*	*98.53*	*98.29*
DNP	*100*	*100*	*100*
DP	*100*	*100*	*100*
FRAG	*98.31*	*100*	*99.15*
IP	*92.19*	*90.17*	*91.17*
LCP	*98.08*	*100*	*99.03*
NP	*88.72*	*85.97*	*87.32*
PP	*99.11*	*100*	*99.55*
PRN	*0.00*	*0.00*	*0.00*
QP	*100*	*98.88*	*99.44*
UCP	*0.00*	*0.00*	*0.00*
VCD	*50.00*	*33.33*	*40.00*
VNV	*0.00*	*0.00*	*0.00*
VP	*93.97*	*96.11*	*95.03*
VRD	*80.00*	*40.00*	*53.33*
VSB	*0.00*	*0.00*	*0.00*
ALL	*93.20*	*92.17*	*92.68*

For the MSRA chunking data set, Table 13 shows the chunking results. As before, MEMM and MEM used the same feature template, defined in Table 6. The experimental results show that the MEMM chunking model was more efficient for resolving the Chinese chunking problem. The reason is that MEMM chunking model uses sufficient context information that can describe actual language phenomena effectively, as explained in Section 3.3.

Table 14 shows the MEMM chunking results for the MSRA data set with different types of feature templates. The baseline and feature templates were defined the same as in Table 11. The performance achieved using POS information only was again much better than the performance achieved using lexical information only. One reason is that the model using lexical features has a more serious data sparseness problem than the model using POS features does. The other reason is that POS tags have a stronger ability to predict chunk tags and that POS tag are the gold standard (because they are manually annotated). The performance achieved using lexical and POS information was again better than the performance achieved using POS information only. This means that lexical information can improve chunking accuracy because it provides sufficient context information for predicting the current chunk tag.

Table 13. Chunking performance achieved by applying different systems to the MSRA data set

Model	P(%)	R(%)	F (%)
HMM	*87.47*	*89.61*	*88.53*
MEM	*90.95*	*88.74*	*89.83*
MEMM Lexical and POS features	*91.36*	*90.68*	*91.02*

Table 14. MEMM chunking performance achieved by applying different feature templates to the MSRA data set

Model	P(%)	R(%)	F (%)
Baseline	*64.27*	*72.12*	*67.97*
MEMM Lexical features	*74.91*	*75.37*	*75.14*
MEMM POS features	*85.47*	*85.28*	*85.38*
MEMM Lexical and POS features	*91.36*	*90.68*	*91.02*

Table 15 shows the performance of different chunk types for HMM and MEM when the total MEMM F-score in total was 91.02% on the MSRA data set. Because *NP* and *VP* chunks accounted for 75.76% of all chunks, their performance dominated the overall chunking performance. As shown, the performance of *VP* was somewhat better, while the performance of *NP* was much lower than average, just as in the experimental results for the CPTB data set (shown in Table 12). The performance of *PP*, *CONJP*, and *INTJP* was somewhat better because most of them are single words. For almost all the chunk types, the performance of MEMM is the best. HMM was better for the *INDP* chunk type because the AL of *INDP* was 4.297 and the HMM method can classify chunk types that have longer AL.

In order to show the relationship between MEMM and the data set size, we split the MSRA training data set into parts with different sizes. Figure 3 shows the results for different sizes of training data sets with the feature template shown in Table 6. When the size of the training data set increased to 6,900 sentences, that is, forty percent of the whole training data set, the F-score was 90%. However, when the size of the training data set increased to 17,253 sentences, the F-score only increased by one percent. Thus, it can be seen that expanding the scale of the training data set helps the chunking performance very little after the data set reaches a certain scale.

Table 15. The performance of each chunk type for the MSRA data set

Chunk type	MEMM P (%)	MEMM R (%)	MEMM F (%)	HMM F (%)	MEM F (%)
NP	*88.64*	*87.48*	*88.06*	*85.95*	*87.59*
VP	*95.25*	*96.81*	*96.03*	*92.60*	*94.96*
PP	*93.98*	*93.88*	*93.93*	*92.86*	*94.27*
MP	*88.69*	*83.71*	*86.13*	*88.35*	*84.84*
ADJP	*92.26*	*84.76*	*88.35*	*84.17*	*86.03*
SP	*82.99*	*85.60*	*84.28*	*77.93*	*83.51*
TP	*92.02*	*92.02*	*92.02*	*89.91*	*84.57*
CONJP	*99.34*	*94.62*	*96.92*	*97.65*	*89.35*
INDP	*78.76*	*83.96*	*81.28*	*91.28*	*54.82*
ADVP	*91.98*	*79.68*	*85.39*	*76.84*	*83.73*
INTJP	*95.65*	*95.65*	*95.65*	*79.31*	*86.25*
ALL	*91.36*	*90.68*	*91.02*	*88.53*	*89.93*

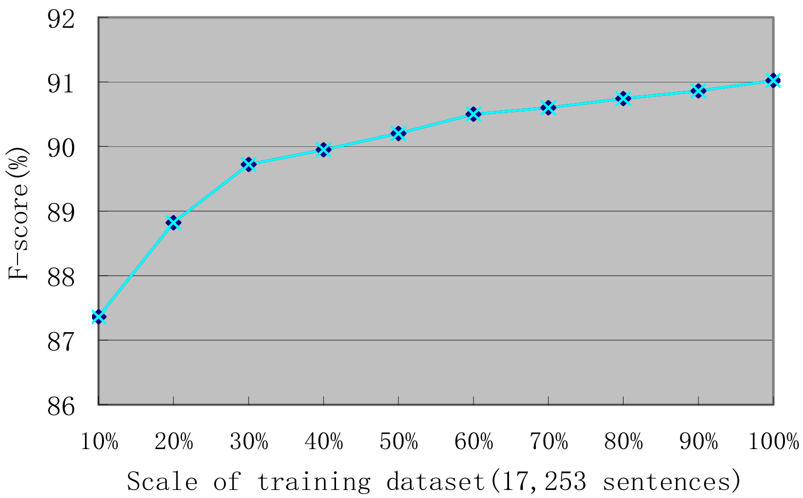

Figure 3. The results for MSRA training data sets of different sizes using the feature template shown in Table 6

Figure 4 shows the results for training data sets of different sizes using the feature template shown in Table 4, which only has lexical information. When the entire training data set was used, the F-score was 74.27%. But the curve shows that the F-score could still improve significantly if the scale of the training data set were increased. This means that there is much room to improve the accuracy if we enlarge the training corpus further.

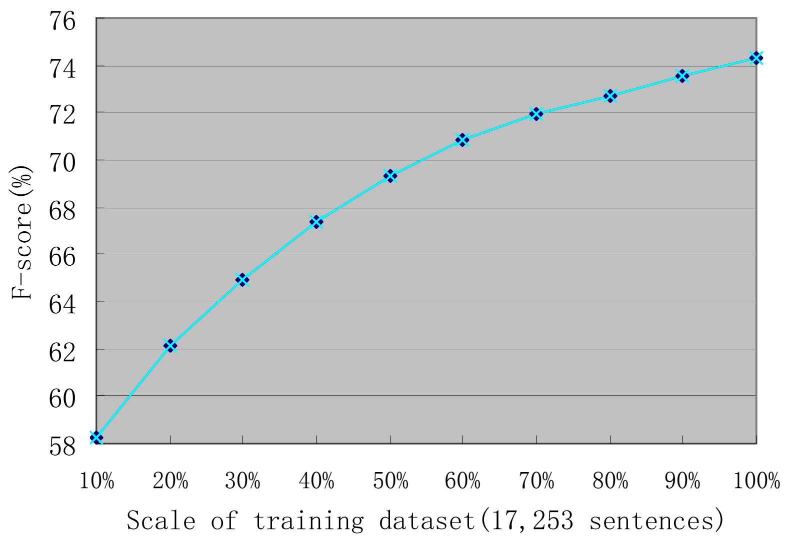

Figure 4. The results for MSRA training data sets of different sizes using the feature template shown in Table 4

Table 16. The distribution of each type of error in the MSRA data set

Error type		Wrong labeling	Under-combining	Over-combining	Overlapping
HMM	No. of the Errors	*55*	*591*	*316*	*70*
	Percentage (%)	*5.3*	*57.3*	*30.6*	*6.9*
MEM	No. of the Errors	*32*	*530*	*305*	*69*
	Percentage (%)	*3.4*	*56.6*	*32.6*	*7.4*
MEMM	No. of the Errors	*25*	*431*	*330*	*66*
	Percentage (%)	*2.9*	*50.6*	*38.7*	*7.7*

Table 16 shows the number and percentage of each type of error in the MEMM results, compared with those in the HMM and MEM results. Four types of Chinese chunking errors are defined: wrong labeling, under-combining, over-combining, and overlapping. Since one chunking error can possibly result in two chunk tagging errors, there were 852 chunking errors. Under-combining and over-combining errors amounted to almost 90% in all the errors for all three models, so identifying the boundaries of chunks is important to get better performance. The reason why MEMM has the best performance is that the numbers of the two types of errors decrease when the sequential relations of the chunk tags are considered.

5. Conclusion

In this paper we have proposed a new method of Chinese chunking based on MEMM. The transition probabilities of chunk tags are estimated using the Markov model. A smoothing algorithm is applied to deal with the data sparseness problem of the chunk tag bi-gram. The conditional probabilities of chunk tags along with histories are estimated through MEM. The two probabilities are combined dynamically in MEMM.

For the purpose of comparing the performance of different models, chunking models were applied to both the CPTB chunking data set and MSRA chunking data set. The experiments on the PTCB data set showed that the new model achieved an F-score of 92.68%, which was better than the F-scores of HMM and MEM in Chinese chunking. The improvement was 2.74% and 1.06%, respectively. The experiments on the MSRA data set showed that the new model had an F-score of 91.02%, which was also better than the F-scores of HMM and MEM. The improvement in this case was 2.49% and 1.19%, respectively. The reasons for the improvement have been analyzed through error analysis. We have also discussed the effects of different feature types and different sizes of training data sets on the performance of MEMM.

Acknowledgements

We thank Dr. Ming Zhou, Dr. Jianfeng Gao, Dr. Mu Li, Dr. Yajuan Lv, Dr. John Chen, and Dr. Hongqiao Li for their valuable suggestions and for checking the English in this paper. We also thank the members of the Natural Language Computing Group at Microsoft Research Asia. We especially thank the anonymous reviewers for their insightful comments and suggestions, based on which the paper has been revised.

References

Abney, S., “Parsing by Chunks”, *Principle-Based Parsing*, Kluwer Academic Publishers, Dordrecht, 1991, pp. 257-278.

Berger, A., S. A. Della Pietra, and V. J. Della Pietra, “A Maximum Entropy Approach to Natural Language Processing,” *Computational Linguistics*, 22(1), 1996, pp. 39-71.

Bikel, D.M., and D. Chiang, “Two Statistical Parsing Models Applied to The Chinese Treebank,” In *Proceedings of the second Chinese Language Processing Workshop,* Hong Kong, China, 2000, pp. 1-6.

van den Bosch, A., and S. Buchholz, “Shallow Parsing on the Basis of Words Only: a Case Study,” In *Proceedings of the 40th Annual Meeting of ACL*, Philadelphia, PA, USA, 2002, pp. 433–440.

Carroll, J., T. Briscoe, G. Carroll, M. Light, D. Prescher, M. Rooth, S. Federici, S. Montemagni, V. Pirrelli, I. Prodanof and M. Vannocchi, “Phrasal Parsing Software”, *Sparkle Work Package 3*, 1997, Deliverable D3.2.

Della Pietra, S., V. J. Pietra, and J. Laffery, “Inducing Features for Random Fields,” *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19(4), 1997, pp. 380-393.

Fung, P., G. Ngai, Y. Yang, and B. Chen, “A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning,” *ACM Transactions on Asian Language Information Processing*, 3(2), 2004, pp. 159-168.

Gao, J., J. Goodman, M. Li, and K. Lee, “Toward a Unified Approach to Statistical Language Modeling for Chinese,” *ACM Transactions on Asian Language Information Processing*, 1(1), 2002, pp. 3-33.

Li, H., C. N. Huang, J. Gao, and X. Fan, “Chinese Chunking with Another Type of Spec,” In *Proceedings of the 3rd ACL SIGHAN Workshop*, Barcelona, Spain, 2004, pp. 41-48.

Li, S., Q. Liu, and Z. Yang, “Chunk Parsing with Maximum Entropy Principle,” *Chinese Journal of Computers*, 25(12), 2003, pp. 1722-1727.

Li, X., and D. Roth, “Exploring Evidence for Shallow Parsing,” In *Proceedings of the CoNLL-2001*, Toulouse, France, 2001, pp. 38-44.

Kim Sang, E. Tjong, and S. Buchholz, “Introduction to the CoNLL-2000 Shared Task: Chunking,” In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 127-132.

Kudoh, T., and Y. Matsumoto, “Use of Support Vector Learning for Chunk Identification,” In *Proceeding of CoNLL-2000 and LLL-2000,* Lisbon, Portugal, 2000, pp. 142-144.

Kinyon, A., “A Language-independent Shallow-parser Compiler,” In *Proceedings of 39th ACL Conference*, Toulouse, France, 2001, pp. 322-329.

Koeling, R., “Chunking with Maximum Entropy Models,” In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 139-141.

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz, “Building a Large Annotated Corpus of English: The Penn Treebank,” *Computational Linguistics*, 19(2), 1993, pp. 313-330.

McCallum, A., D. Freitag, and F. Pereira, “Maximum Entropy Markov Models for Information Extraction and Segmentation,” In *Proceedings of ICML’2000*, Stanford, CA, USA, 2000, pp. 591-598.

Luo, X., "A Maximum Entropy Chinese Character-based Parser," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, 2003.

Osborne, M., "Shallow Parsing as Part-of-speech Tagging," In *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 145-147.

Park, S. B., and B. T. Zhang, "Text Chunking by Combining Hand-crafted Rules and Memory-based Learning," In *Proceedings of the 41st Annual Meeting of ACL*, Sapporo, Japan, 2003, pp. 497-504.

Ramshaw, L. A., and M. P. Marcus, "Text Chunking Using Transformation-based Learning," In *Proceedings of the 3rd ACL/SIGDAT Workshop*, Cambridge, MA, USA, 1995, pp. 222-226.

Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," In *Proceedings of the IEEE*, 77(2), 1989, pp. 257-285.

Ratnaparkhi, A.,"A Maximum Entropy Model for Part-Of-Speech Tagging," In *Proceedings of EMNLP'1996*, New Brunswick, New Jersey, USA, 1996, pp. 133-142.

Xia, F., M. Palmer, N. Xue, M. E. Okurowski, J. Kovarik, F. Chiou, S. Huang, T. Kroch, and M. Marcus, "Developing Guidelines and Ensuring Consistency for Chinese Text Annotation," In *Proceedings of the second International Coference on Language Resources and Evaluation*, Athens, 2000.

Xu, J., S. Miller, and R. Weischedel, "A Statistical Parser for Chinese," In *Proceedings of Human Language Technology Workshop*, San Diego, USA, 2002.

Xun, E., C. Huang, and M. Zhou, "A Unified Statistical Model for the Identification of English BaseNP," In *Proceedings of the 38th ACL*, Hong Kong, China, 2000, pp. 109-117.

Xue, N., and F. Xia, "The Bracketing Guidelines for the Penn Chinese Treebank(3.0)," *Technical report*, University of Pennsylvania, 2000, URL: http://www.cis.upenn.edu/~chinese/.

Yu, S., H. Duan, and X. Zhu, B. Sun, "The Basic Processing of Contemporary Chinese Corpus at Peking University," *Journal of Chinese Information Processing*, 16(6), 2002, pp. 58-65.

Zhao, J., and C. N. Huang, "Analysis of Chinese BaseNP Structure," *Chinese Journal of Computers*, 22(2), 1999, pp. 141-146.

Zhang, Y., and Q. Zhou, "Automatic Identification of Chinese Base Phrases," *Journal of Chinese Information Processing*, 16(6), 2002, pp. 1-8.