Chinese Word Segmentation by Classification of Characters

Chooi-Ling GOH Masayuki ASAHARA Yuji MATSUMOTO
Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma,
Nara 630-0192,
Japan.
{ling-g,masayu-a,matsu}@is.naist.jp

Abstract

During the process of Chinese word segmentation, two main problems occur: segmentation ambiguities and unknown word occurrences. This paper describes a method to solve the segmentation problem. First, we use a dictionary-based approach to segment the text. We simply apply maximum matching algorithm to segment the text forwardly (FMM) and backwardly (BMM). Based on the difference between FMM and BMM, and the context, we apply a classification method based on Support Vector Machines to re-assign the word boundaries. By this way, we are using the output of a dictionary-based approach, and then applying a statistics-based approach to solve the segmentation problem. The experimental results show that our model achieves as high as 99.0 point of F-measure for overall segmentation, given the condition that no unknown word in the text, and 95.1 if unknown words exist.

1 Introduction

The first step in Chinese information processing systems is word segmentation. It is because in written Chinese, all characters are joined together, and there is no separator to mark word boundaries. A similar problem also occurs in languages like Japanese, but at least in Japanese, there exist three types of characters (hiragana, katakana and kanji), and this could be a clue for finding word boundaries. For Chinese, as there is only one type of characters (hanzi), more segmentation ambiguities may happen in a text. During the process of segmentation, two main problems occur: segmentation ambiguities and unknown word occurrences. This paper focuses on solving the segmentation ambiguity problem, and proposes a sub-model to solve the unknown word problem. There are basically two types of segmentation ambiguities: covering ambiguity and overlapping ambiguity. The definitions are given below.

Let x, y, z be some strings in Chinese which could consist of one or more characters. Subsequently, covering ambiguity is defined as follows: For string $w = xy$, $x \in W$, $y \in W$ and also $w \in W$, where W is a dictionary. As almost any single character in Chinese can be considered as a word, the definition reflexes only those cases where both word boundaries $.../xy/...$ and $.../x/y/...$ can be realized in some sentences. On the other hand, overlapping ambiguity is defined as follows: For string $w = xyz$, $w_1 = xy \in W$ and also $w_2 = yz \in W$. Although most of the time, the segmentation of one form is more preferred than the other form, but still we need to know where to use the other form. Both ambiguities depend heavily on the contexts to decide which is the correct segmentation given that particular occurrences in the texts.

(1a) and (1b) show examples of covering ambiguity. Given the string "一家", it is treated as a word in (1a), but as two words in (1b).

(1a) 胡 / 世庆 / 一家 / 三 / 口 /
Hu/ Shiqing/ whole family/ three/ member
(The whole three members of Hu Shiqing family)

(1b) 在 / 巴黎 / 一 / 家 / 杂志 / 上 /
in/ Paris/ one/ company/ magazine/ at/
(At one of the magazine company in Paris)

On the other hand, (2a) and (2b) give examples of overlapping ambiguity. The string "不可以" is segmented as "不 / 可以" in (2a) and "不可 / 以" in (2b), according to the context of the sentence.

(2a) 不 / 可以 / 淡忘 / 远在 / 故乡 / 的 / 父母 /
not/ can/ forget/ far away/ hometown/ DE/ parents/
(Cannot forget the parents who are far way at hometown)

(2b) 不可 / 以 / 营利 / 为 / 目的 /

cannot/ by/ profit/ be/ intention/
(Cannot have the intention to make profit)

We intend to solve the ambiguity problems by combining a dictionary-based approach with a statistical model. By this way, we make use of the information in a dictionary to a statistical approach. Maximum Matching (MM) algorithm, a very early and simple dictionary-based approach, is used to initially segment the text by referring to a dictionary. It tries to match the longest possible words found in the dictionary. We can either parse the sentence forwardly or backwardly. Normally, the difference between forward and backward parsing will indicate the location where overlapping ambiguities occur. Then, we use a Support Vector Machine-based (SVM) classifier to decide which output should be the correct answer. For covering ambiguity, most of the cases, forward and backward MM will give the same outputs, in this case, we will just make use of the contexts to decide whether or not to split a word into two words and etc. Our results showed that the proposed method could produce the correct answers for overlapping ambiguities up to 92%, and 52% correctly split the words for covering ambiguities.

2 Previous Work

Solving the ambiguity problems is a fundamental task in Chinese segmentation process. Although many previous researches have been done for segmentation, only a few have reported on the accuracy to solve ambiguity problems. (Li et al., 2003) suggest an unsupervised method for training the Naïve Bayes classifiers to resolve overlapping ambiguities. They achieved 94.13% accuracy with 5,759 cases of ambiguity. A variation form of TF.IDF weighting is proposed for solving covering ambiguity problem in (Luo et al., 2002). They focus on 90 ambiguous words and achieve an accuracy of 96.58%.

Most of the previous methods reported on the accuracy for overall segmentation. Recently, many researches are done by combining multiple models. Furthermore, most people have realized that working on character-based is more efficient than word-based for Chinese word segmentation. In (Xue and Converse, 2002), two classifiers are combined for Chinese word segmentation. First, a Maximum Entropy model is used to segment the text, then an error driven transformation model is used to correct the word boundaries. Similarly, they also use character-based tagging on the position of characters in words. They achieved an F-measure of 95.17. Another recent report is by (Fu and Luke, 2003), where hybrid models for integrated segmentation is proposed. Modified word juncture models and word-formation patterns are used to find the word boundaries and at the same time to identify the unknown words. They achieved 96.1 F-measure. As both methods use different corpora for the experiments, it is difficult to tell which method has done better than the other.

Solving unknown word problem is also an important process in word segmentation. An unknown word is defined as a word not found in a dictionary. Therefore, they cannot be segmented correctly by simply referring to the dictionary. Many approaches have been reported for unknown word detection such as (Chen and Bai, 1997; Chen and Ma, 2002; Fu and Wang, 1999; Lai and Wu, 1999; Ma and Chen, 2003; Nie et al., 1995; Shen et al., 1998; Zhang et al., 2002; Zhou and Lua, 1997). There are rule-based, statistics-based or even hybrid models. In other words, we cannot ignore the unknown word problem, as there always exist some unknown words (such as person names, numbers and etc) in the text even if we can get a very large dictionary. The creation of new words in Chinese is unlimited and is a continuous process. For example, the name for new diseases, technical terms, new expressions and etc. The accuracy is better if one focuses only on certain types of unknown words such as person names, place names or transliteration names, with over 80%. However, for general unknown words such as common nouns, verbs etc, the accuracy ranging from 50% to 70% only.

3 Proposed Method

The underlying concept of our proposed method is as following. We regard the problem as a character classification problem. We believe that each character in Chinese holds its characteristics to appear in a certain position in a word. In other words, it can be either used at the beginning of a word, in the middle of a word, at the end of a word, or as a single-character word. By looking at the usage of the characters, we will decide the position tag of the characters using a machine learning based model, which is the Support Vector Machines (Vapnik, 1995). This method serves as a model to solve ambiguity problem, and at the same time, em-

beds a model to detect unknown words. We will now describe the method in more details in the following section.

3.1 Maximum Matching Algorithm

We intend to solve the ambiguity problem by combining a dictionary-based approach with a statistical model. Maximum Matching (MM) algorithm is regarded as the simplest dictionary-based word segmentation approach. It starts from one end of a sentence, and tries to match the first longest word wherever possible. It is a greedy algorithm, but it has been empirically proved to achieve over 90% accuracy if the dictionary used is large. However, it cannot solve ambiguity problems and impossible to detect unknown words because only words exist in the dictionary can be segmented correctly. If we look at the outputs produced by segmenting the sentence forwardly (FMM), from the beginning of the sentence, and backwardly (BMM), from the end of the sentence, we will realize the places where overlapping ambiguities occur. As an example, FMM will segment the string "即将来临时" (when the time comes) into "即将/ 来临/ 时/" (immediately/ come/ when), but BMM will segment it into "即/ 将来/ 临时/" (that/ future/ temporary).

Let O_f and O_b be the outputs of FMM and BMM respectively. According to (Huang, 1997), for overlapping cases: If $O_f = O_b$, then 99% that both the MMs have the correct answer. If $O_f \neq O_b$, then 99% that either O_f or O_b has the correct answer. However, for covering ambiguity cases, even $O_f = O_b$, but both O_f and O_b could be correct or both could be wrong. If there exist unknown words, normally they will be segmented as single characters by both FMM and BMM. Based on the differences and context created by FMM and BMM, we will apply a machine learning based model to reassign the position tags which indicate the character position in the word.

3.2 Re-classification of Characters

We plan to re-classify the outputs of FMM and BMM character by character. First, we will convert the output of the MMs into character-based, where each character will be assigned with a position tag such as described in Table 1. The BIES tags are as in (Uchimoto et al., 2000) and (Sang and Veenstra, 1999) for named entity extraction. These tags show the possible character position in a word. For example, if we look at the character "本", it is used as a single character in "一/ 本/ 书/" (a book), at the end of a word in "剧本" (script), at the beginning of a word in "本来" (originally), and at the middle of a word in "基本上" (basically).

Tag	Description
S	one-character word
B	first character in a multi-character word
I	intermediate character in a multi-character word (for words longer than two characters)
E	last character in a multi-character word

Table 1: Position tags in a word

Then, based on these features, we will re-classify the tags by using a Support Vector Machine-based (SVM) chunker (Kudo and Matsumoto, 2001). The solid box in Figure 1 shows the features used to determine the class of the character "春" at location *i*. Based on the output position tags, finally, we will get the segmentation as "迎/ 新春/ 联谊会/ 上/" (welcome/ new year/ get-together party/ at/).

Position	Char.	FMM	BMM	Output
i - 2	迎	B	S	S
i - 1	新	E	B	B
i	春	B	E	E
i + 1	联	E	B	B
i + 2	谊	S	E	I
i + 3	会	B	B	E
i + 4	上	E	E	S

Figure 1: An illustration of classification process - 'At the New Year gathering party'

4 Experiments and Results

We have run our experiments with two datasets, PKU Corpus and SIGHAN Bakeoff data. The evaluation is done by using the tool provided in SIGHAN Bakeoff (Sproat and Emerson, 2003).

4.1 Experiment with PKU Corpus

4.1.1 Accuracy on Solving Ambiguity Problem

The corpus used for this experiment is from Peking University (PKU) [1], consisting of about 1 million words. It is a segmented and POS

[1] Institute of Computational Linguistics, Peking University, http://www.icl.pku.edu.cn/

tagged corpus, but we only use the segmentation information for our experiments. We divide the corpus randomly into 80% and 20%, for training and testing respectively. Since our purpose in this experiment is only for solving ambiguity problem, not the unknown word detection, we assume that all words could be found in the dictionary. We create a dictionary with all words from the corpus, which has 62,030 entries (referred to as Experiment 1). This experiment intends to show the performance of the method for solving ambiguity problem.

It is sometimes very difficult to determine how many cases of ambiguities appearing in a sentence. For example, in the sentence in Figure 1, "迎新" (welcome the new year), "新春" (new year), "春联" (a red paper that pasted on the door, written with some greeting words for celebrating new year in China), "联谊" (get-together), "联谊会" (get-together party), "会上" (at the meeting) and "上" (at) are all possible words. How many overlapping cases and covering cases are there? It is quite impossible to answer. A word candidate may cause more than one ambiguities with the alternative word candidates. Therefore, we try to represent the ambiguities by character units since our method is character based. We group each character into one of these six categories.

Let,

O_f = Output of FMM
O_b = Output of BMM
Ans = Correct answer
Out = Output from our system

Category	Conditions
Allcorrect	$O_f = O_b = Ans = Out$
Correct	$O_f \neq O_b$ and $Ans = Out$
Wrong	$O_f \neq O_b$ and $Ans \neq Out$
Match	$O_f = O_b$ and $O_f \neq Ans$ and $Ans = Out$
Mismatch	$O_f = O_b$ and $O_f \neq Ans$ and $Ans \neq Out$
Allwrong	$O_f = O_b = Ans$ and $Ans \neq Out$

Table 2: Categories for Characters

Table 2 shows the conditions for each category. Category *Allcorrect*, *Correct* and *Match* have correct answers, whereas category $Wrong$, *Mismatch* and *Allwrong* have wrong answers. We could roughly say that category *Correct* and $Wrong$ belong to overlapping ambiguities and category *Match*, *Mismatch*, and *Allwrong* belong to covering ambiguities. We could also say that *Match* and *Mismatch* are cases where we need to split the words, and *Allwrong* are cases where we should not split the words but have been split by the system. Table 3 shows the results of the method for solving ambiguity problem.

Category	No. of Char.	Percentage
Allcorrect	330220	96.35%
Correct	7663	2.23%
Wrong	658	0.19%
Match	1876	0.55%
Mismatch	1738	0.51%
Allwrong	571	0.17%
Total	342726	100.00%

Table 3: Results on Disambiguation

In total, we could obtain about 99.13% that the characters are correctly tagged. If we only consider the overlapping cases (*Correct* and $Wrong$), 92.09% characters are correctly tagged. For covering cases, if we look at only those cases where we need to split the words (*Match* and *Mismatch*), 51.91% have been successfully split.

Table 4 shows the results of word segmentation. We also compare our method with a Hidden Markov Model-based (HMM) morphological analyzer, where word bi-gram is used to calculate the probability. The size of the dictionary used for HMM is the same as previous experiment, but with real POS tags. The HMM does segmentation and POS tagging simultaneously, but we only take the results of segmentation for comparison. The results show that our proposed method can achieve higher accuracy with over 99.0%. It means that our method is able to solve ambiguity problem given the information where the ambiguous locations occurred by looking at the output of FMM and BMM.

4.1.2 Accuracy on Solving Unknown Word Problem

The corpus used is the same as in Section 4.1.1, but the setting is different. In this round we divide the corpus into three sets, referring to as Set 1, Set 2 and Set 3. Set 1 plus Set 2 (80%) are used for training and Set 3 (20%) is used for testing, same as the previous experiment. The difference is the preparation of dictionary. There are two ways of preparation here. In the first case, all the words from Set 1 and Set 2 are

	FMM	BMM	SVM (char. only)	FMM + SVM	BMM + SVM	FMM + BMM + SVM (=Experiment 1)	HMM (with POS tag)
Recall	96.9	97.1	94.0	98.7	98.7	**98.9**	97.9
Precision	97.7	97.9	94.3	98.9	99.0	**99.1**	98.5
F-measure	97.3	97.5	94.1	98.8	98.9	**99.0**	98.2

Table 4: Segmentation Results

used to create the dictionary. There are 49,433 entries, and there exist 8,346 (4.0%) unknown words in the testing data (referred to as Experiment 2). This experiment intends to investigate the performance of the method if unknown words exist. In the second case, only the words from Set 1 are used to create the dictionary, making the situation that there exist unknown words in the training data (referred to as Experiment 3). The top part of Table 5 shows the proportion of Set 1 and Set 2, with the size of the dictionaries and the numbers of unknown words in Set 2 and testing data. Set 2 serves as a learning model for unknown word detection. While we segment Set 2 using FMM and BMM, most of the unknown words will be segmented into single characters (namely tag 'S'). Based on these tags and contexts, SVM-based chunker will be trained to change the tag into the correct answers. The last experiment (referred to as Experiment 4) is the reverse of Experiment 2, where nothing is used to create the dictionary. All the words are considered as unknown words. Only the characters are used as features during classification, meaning no information from FMM and BMM is available.

Bottom part of Table 5 shows the results of these experiments. Our method in fact works quite well for both solving segmentation ambiguity and unknown word detection. The problem is, while the accuracy for unknown word detection improves, at the same time, the performance degrades in solving the ambiguity problem. It is because the precision of unknown word detection is not one hundred percent. The highest recall that we can get for known words is 98.9% and for unknown words is 69.3%. However the best overall segmentation result is by dividing the training corpus into 40%/40%. This is the optimal point where a balance is found for detecting unknown words, while at the same time maintaining the accuracy of segmentation ambiguity for known words. Figure 2 shows the F-measure for segmentation, recall

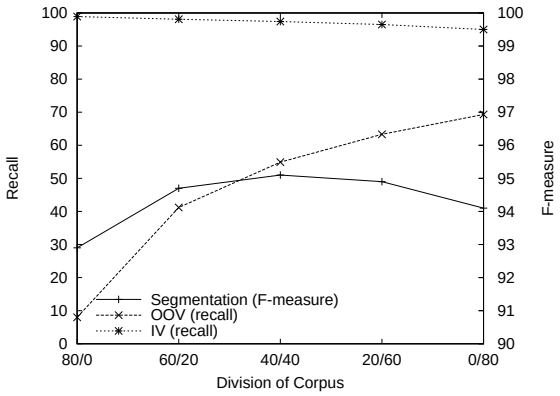

Figure 2: Accuracy by segmentation (F-measure), OOV (recall) and IV (recall)

for unknown words and known words, by different division of training corpus for creating dictionary.

4.2 Experiment with SIGHAN Bakeoff Data

As far as we know, there is no standard definition for Chinese word segmentation. The text can be segmented differently by different people depending on the linguists who decide on the rules and also the usage of the segmentation. Therefore, it is always difficult to compare the result with other methods as the data used is different. The First International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003) intended to evaluate the accuracy of different segmenters by standardizing the training and testing data. In their closed test, only the training data is allowed to be used for training but no other material. Under this strict condition, it is possible to create a lexicon from the training data, but of course the unknown words will exist in the testing data. We have run an experiment using the bakeoff data (only the corpus from PKU and Penn Chinese

	Experiment 1	Experiment 2	Experiment 3			Experiment 4
Set1(%)/Set2(%)		80/0	60/20	40/40	20/60	0/80
# of words in Dict.	62,030	49,433	41,582	33,355	22,363	0
# of unk-words in Set 2	0	0	10,927	25,297	53,353	All
# of unk-words in Test	0	8,346	9,768	11,924	17,115	All
Recall	98.9	95.3	**95.8**	95.7	95.2	94.0
Precision	99.1	90.7	93.5	94.5	**94.7**	94.3
F-measure	99.0	92.9	94.7	**95.1**	94.9	94.1
OOV (recall)	–	8.0	41.2	54.9	63.3	**69.3**
IV (recall)	98.9	**98.9**	98.1	97.4	96.5	95.0

Table 5: Different Settings and Segmentation Results with Unknown Words

Treebank, CHTB [2]). Since our system works only on two-byte codings, some ascii codes in the data have been converted to GB codes before processing, especially numbers and alphabets. The distribution of the data is as shown in Table 6. The original dictionaries consist of 55,226 and 19,730 words respectively. According to these dictionaries, there are 1,189 and 7,216 unknown words in the testing data. After converting to GB codes, it left only 781 and 7,171 unknown words. It also means that about 34.3% and 0.6% of the unknown words automatically become known words after the conversion. The conversion reduced the number of unknown words because for example, if a numeral word "１９９８" written in GB code exists in training data, but it is written in ascii code "1998" in testing data, it is treated as unknown word at the first place. After the conversion, it will become known word.

We have set up the experiments similar to Experiment 2 and Experiment 3 above. For Experiment 2, all the words in the training data are used for creating the dictionary. For Experiment 3, it is based on our previous experiments where the division of half of the training corpus generated the best result by F-measure. Therefore, only 50% of the training corpora are used while creating the dictionaries. As a result, the new dictionaries contain 36,830 and 12,274 words respectively. Table 6 shows the details for the setting.

For PKU corpus, the best result in the bakeoff achieved 95.1 in F-measure (Zhang et al., 2003). They use hierarchical Hidden Markov Models to segment and POS tag the text. Although it is a closed test, they have used extra information such as class-based segmentation and role-based tagging models (Zhang et al., 2002), which give better result for unknown word recognition. Our method has done only slightly worse than theirs, with 94.7. The recall for unknown words is comparable, with 71.0% while the best one has 72.4%. Unfortunately, the recall for known words drops a bit, with 97.3%, while the best one is slightly better, with 97.9%, as shown in Table 7. We also compare with (Asahara et al., 2003), where similar method is used for reassigning the word boundaries, except that the words are first categorized into 5 or 10 classes (which is assumed equivalent to POS tags) using Baum-Welch algorithm, then the sentence is segmented into word sequence by a Hidden Markov Model-based segmenter. Finally, the same Support Vector Machine-based chunker is trained to correct the errors made by the segmenter. Our method which is simply a forward and backward maximum matching algorithm, has done a lot better then theirs, where complicated statistical based models are involved. They have achieved only 92.4 F-measure while we have 94.7.

On the other hand, our results for CHTB corpus are not as comparable as the best result in the bakeoff. We could only get 84.7 point of F-measure, while the best one has 88.1 (Zhang et al., 2003) and 82.9 by (Asahara et al., 2003). It may be due to the reason that the training corpus is a lot smaller than the PKU corpus and the testing data contains more unknown words.

[2] We work only on GB code, the standard coding used for simplified Chinese characters. However, it can be modified easily to suit Big5 coding for traditional Chinese characters.

	PKU Data			CHTB Data		
	# of words	# of unk-words	unk-word rate	# of words	# of unk-words	unk-word rate
Original Training	1,121,017	0	0%	250,841	0	0%
Original Testing (In GB code)	17,194	1,189 (781)	6.9% (4.5%)	39,922	7,216 (7,171)	18.1% (18.0%)
Set 1	560,649	0	0%	125,405	0	0%
Set 2	560,368	29,303	5.2%	125,436	13,976	11.1%
Testing Data	17,194	1,121	6.5%	39,922	9,769	24.5%

Table 6: Bakeoff Data

However, we still could get quite good recall for unknown words, with 57.7%, while the others have 70.5% and 41.2% respectively.

As a conclusion, our results cannot transcend the best results in the bakeoff for both corpora. However, our method is simpler. We only need a dictionary that created from a segmented corpus, FMM and BMM modules, and a classifier, without the intervention of human knowledge. We get quite comparable results for both known words and also unknown words. The result is worse when the training corpus is small and there exist a lot of unknown words such as in CHTB testing data. Therefore, we still need to investigate on the relationship between the size of training corpus and the division of corpus for training of ambiguity problem and unknown word detection.

5 Conclusion

Apparently, our proposed method has generated better result than the baseline models, FMM and BMM. We get nearly 99% accuracy if unknown words do not exist. However, in the real world, it is impossible that there is no unknown word at all even we could get a very large dictionary. Therefore, we also embedded a model to detect the unknown words. Unfortunately, while the accuracy for unknown word detection increased, the performance on solving known word ambiguity drops. As shown in the experiments with bakeoff data, our model works well only when the training corpus is large enough. As a conclusion, while our model is suited for solving segmentation ambiguity problem, it can also be used for unknown word detection. However we still need to find a balance point for solving these two problems simulteneously. We also need to research on the relationship between training corpus size and the best proportion to divide the corpus for training optimally on solving ambiguity problem and unknown word detection.

6 Acknowledgements

Thanks to Mr. Kudo for his tool on Support Vector Machine-based chunker, Yamcha. We also thank the reviewers for their invaluable comments, Peking University and SIGHAN for providing the corpora in our experiments.

References

Masayuki Asahara, Chooi Ling Goh, Xiaojie Wang, and Yuji Matsumoto. 2003. Combining Segmenter and Chunker for Chinese Word Segmentation. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 144–147.

Keh-Jiann Chen and Ming-Hong Bai. 1997. Unknown Word Detection for Chinese By a Corpus-based Learning Method. In *Proceedings of ROCLING X*, pages 159–174.

Keh-Jiann Chen and Wei-Yun Ma. 2002. Unknown Word Extraction for Chinese Documents. In *Proceedings of COLING 2002*, volume 1, pages 169–175.

Guohong Fu and K.K. Luke. 2003. An Integrated Approach for Chinese Word Segmentation. In *Proceedings of PACLIC 17*.

Guohong Fu and Xiaolong Wang. 1999. Unsupervised Chinese Word Segmentation and Unknown Word Identification. In *Proceedings of NLPRS*.

Changning Huang. 1997. Segmentation problem in chinese processing. *Applied Linguistics*, 1:72–78.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with Support Vector Machines. In *Proceedings of NAACL*, pages 192–199.

Yu-Sheng Lai and Chung-Hsien Wu. 1999. Unknown Word and Phrase Extraction Using a

	Without Unknown Word Processing (=Experiment 2)	With Unknown Word Processing (=Experiment 3)	Baum-Welch + HMM + SVM (Asahara et al., 2003)	Best Result in Bakeoff (Zhang et al., 2003)
PKU Corpus				
Recall	94.6	95.5	93.3	**96.2**
Precision	89.4	**94.1**	91.6	94.0
F-measure	92.0	94.7	92.4	**95.1**
OOV (recall)	39.7	71.0	35.7	**72.4**
IV (recall)	**98.7**	97.3	97.5	97.9
CHTB Corpus				
Recall	82.2	86.0	85.2	**88.6**
Precision	67.4	83.5	80.7	**87.5**
F-measure	74.1	84.7	82.9	**88.1**
OOV (recall)	10.1	57.7	41.2	**70.5**
IV (recall)	**98.2**	92.2	94.9	92.7

Table 7: Segmentation Results with Bakeoff Data

Phrase-Like-Unit-Based Likelihood Ratio. In *Proceeding of ICCPOL '99*, pages 5–9.

Mu Li, Jianfeng Gao, Changning Huang, and Jianfeng Li. 2003. Unsupervised Training for Overlapping Ambiguity Resolution in Chinese Word Segmentation. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 1–7.

Xiao Luo, Maosong Sun, and Benjamin K. Tsou. 2002. Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information. In *Proceedings of COLING 2002*, pages 598–604.

Wei-Yun Ma and Keh-Jiann Chen. 2003. A Bottom-up Merging Algorithm for Chinese Unknown Word Extraction. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 31–38.

Jian-Yun Nie, Marie-Louise Hannan, and Wanying Jin. 1995. Unknown Word Detection and Segmentation of Chinese Using Statistical and Heuristic Knowledge. *Communications of COLIPS*, Vol.5:47–57.

Erik F. Tjong Kim Sang and John Veenstra. 1999. Representing Text Chunks. In *Proceedings of EACL '99*, pages 173–179.

Dayang Shen, Maosong Sun, and Changning Huang. 1998. The application & implementation of local statistics in Chinese unknown word identification. *Communications of COLIPS*, Vol.8.

Richard Sproat and Thomas Emerson. 2003. The first international chinese word segmentation bakeoff. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143.

Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. 2000. Named Entity Extraction Based on A Maximum Entropy Model and Transformational Rules. In *Processing of the ACL 2000.*

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory.* Springer.

Nianwen Xue and Susan P. Converse. 2002. Combining Classifiers for Chinese Word Segmentation. In *Proceedings of 1st SIGHAN Workshop on Chinese Language Processing*, pages 57–63.

Hua-Ping Zhang, Qun Liu, Hao Zhang, and Xue-Qi Cheng. 2002. Automatic Regcognition of Chinese Unknown Words Based on Roles Tagging. In *Proceedings of 1st SIGHAN Workshop on Chinese Language Processing.*

Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HHMM-based Chinese Lexical Analyzer ICTCLAS. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187.

Guo-Dong Zhou and Kim-Teng Lua. 1997. Detection of Unknown Chinese Words Using a Hybrid Approach. *Computer Processing of Oriental Language*, 11(1):63–75.