Solving Segmentation Ambiguities in Chinese

GOH Chooi Ling ASAHARA Masayuki MATSUMOTO Yuji
Graduate School of Information Science, Nara Institute of Science and Technology.
{ling-g,masayu-a,matsu}@is.aist-nara.ac.jp

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 are no separators to mark the word boundaries. The similar problem also occurs in language like Japanese, but at least in Japanese, there exist three types of characters, and that could be some clues for finding the word boundaries. For Chinese, as there is only one type of characters, more ambiguitions could happen in the text. During the process of segmentation, two main problems occur: segmentation ambiguities and unknown word occurrences. This paper focuses only on solving the segmentation ambiguity problem. There are basically two types of segmentation ambiguity: 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 w and xy 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 minority 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 problem by combining a dictionary-based approach with a statistical model. By this way, we make use of the information in the 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 works have been done for segmentation but only a few have reported on the efficiency to solve ambiguity problems. [Li et al., 2003] suggest an unsupervised method for training the Naïve Bayesian classifiers to resolve overlapping ambiguities. They have achieved 94.13% accuracy with 5,759 cases of ambiguity. A variation form of TFIDF 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%.

3 Proposed Method

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 approach of segmentation. 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 proved to achieve over 90% accuracy. However, it cannot solve ambiguity problems (and impossible to detect unknown words). If we look at the outputs produced by segmenting the sentence forwardly (FMM), meaning from the beginning of sentence, and backwardly (BMM), from the end of the sentence, we will immediately 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 /", but BMM will segment it into "即 / 将来 / 临时 /".

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% the MM has the correct answer, if $O_f \neq O_b$, then 99% 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.

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. These tags show the possibility of a character being appeared in that 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).

Table 1: Position tags in a word

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

Then, based on these features, we will re-classify the tags by using Support Vector Machine-based 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 tags, finally, we will get the segmentation as "迎 / 新春 / 联谊会 / 上 /".

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'

Table 2: Categories for Characters

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 $Ans \neq Out$

4 Experimental Results

4.1 Experiment with PKU Corpus

The corpus used for this experiment is from Peking University (PKU), consisting of about 1 million words. We divide the corpus randomly into 80% and 20%, for training and testing respectively. Since our purpose is only for segmentation disambiguition, 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.

It is sometimes very difficult to determine how many cases of ambiguities 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. Therefore, we will evaluate our ambiguity result in a different way. Since our method is character-based, we will evaluate them character by character. We group the characters into six categories. Let,

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

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 disambiguation.

Table 3: Results on Disambiguation

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%

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 correct. 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 segmentation. We also compare our method with a Hidden Markov Model-based (HMM) morphological analyzer, where word bi-gram is considered. The size of the dictionary used for HMM is the same as described, but with POS tags. The

Table 4: Segmentation Results

	FMM	BMM	SVM (char. only)	SVM (char. + FMM + BMM)	HMM
Recall	96.86	97.12	93.80	**98.83**	97.93
Precision	97.67	97.94	94.27	**99.11**	98.50
F-measure	97.26	97.53	94.03	**98.97**	98.21

HMM does segmentation and POS tagging simultaneously, but we only take the results of segmentation for comparison. We also compare the results where the answers produced by FMM and BMM are not used in SVM classification, in other words, only the characters are used as features. The results show that our proposed method can achieve higher accuracy then others. 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.

4.2 Experiment with Bakeoff Data

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 compare the efficiency 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). The training data has 1,121,017 words and the testing data has 17,194 words. The dictionary is created from the training data and there are 55,226 words. There exist 1,189 (6.9%) unknown words in the testing data.

The best result in the bakeoff achieved 95.1 in F-measure. Since our method does not work on unknown words, we could only achieve 88.0. The recall for unknown words is very low, only 9.6% while the best one has 72.4%. Fortunately, the recall for known words is satisfactory, with 98.7%, while the best one is slightly lower, with 97.9%.

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. Unfortunately, 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 still need to combine with another method for solving the unknown word problem.

References

[Huang, 1997] Huang, Changning. (1997). Segmentation problem in Chinese Processing. In *Applied Linguistics*, 1:72–78. (in Chinese)

[Kudo and Matsumoto, 2001] Kudo, Taku and Matsumoto, Yuji. (2001). Chunking with Support Vector Machines. In *Proceedings of NAACL 2001.*

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

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

[Sproat and Emerson, 2003] Sproat, Richard and Emerson, Thomas. (2003). The First International Chinese Word Segmentation Bakeoff. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143.