A two-stage statistical word segmentation system for Chinese

Guohong Fu
Dept of Linguistics
The University of Hong Kong
Pokfulam Road, Hong Kong
ghfu@hkucc.hku.hk

K.K. Luke
Dept of Linguistics
The University of Hong Kong
Pokfulam Road, Hong Kong
kkluke@hkusua.hku.hk

Abstract

In this paper we present a two-stage statistical word segmentation system for Chinese based on word bigram and word-formation models. This system was evaluated on Peking University corpora at the First International Chinese Word Segmentation Bakeoff. We also give results and discussions on this evaluation.

1 Introduction

Word segmentation is very important for Chinese language processing, which aims to recognize the implicit word boundaries in Chinese text. During the past decades, great success has been achieved in Chinese word segmentation (Nie, et al, 1995; Yao, 1997; Fu and Wang, 1999; Wang et al, 2000; Zhang, et al, 2002). However, there still remain two difficult problems, i.e. ambiguity resolution and unknown word (so-called OOV word) identification, while developing a practical segmentation system for large open applications.

In this paper, we present a two-stage statistical word segmentation system for Chinese. In the first stage, we employ word bigram model to segment known words (viz. the words included in the system dictionary) in input. In the second stage, we develop a hybrid algorithm to perform unknown word identification incorporating word contextual information, word-formation patterns and word juncture model.

The rest of this paper is organized as follows: Section 2 presents a word bigram solution for known word segmentation. Section 3 describes a hybrid approach for unknown word identification. In section 4, we report the results of our system at the SIGHAN evaluation program, and in the final section we give our conclusions on this work.

2 The first stage: Segmentation of known words

In a sense, known word segmentation is a process of disambiguation. In our system, we use word bigram language models and Viterbi algorithm (1967) to resolve word boundary ambiguities in known word segmentation.

For a particular input Chinese character string $C = c_1 c_2 \cdots c_n$, there is usually more than one possible segmentation $W = w_1 w_2 \cdots w_m$ according to given system dictionary. Word bigram segmentation aims to find the most appropriate segmentation $\hat{W} = w_1 w_2 \cdots w_m$ that maximizes the probability $\prod_{i=1}^{m} P_r(w_i \mid w_{i-1})$, i.e.

$$\hat{W} = \arg\max_{W} P_r(W \mid C) \approx \arg\max_{W} \prod_{i=1}^{m} P_r(w_i \mid w_{i-1}) \qquad (1)$$

where $P_r(w_i \mid w_{i-1})$ is the probability that word w_l will occur given previous word w_{i-1}, which can be easily estimated from segmented corpus using maximum likelihood estimation (MLE), i.e.

$$P_r(w_i \mid w_{i-1}) \approx \frac{Count(w_{i-1} w_i)}{Count(w_{i-1})} \qquad (2)$$

To avoid the problem of data sparseness in MLE, here we apply the linear interpolation technique (Jelinek and Mercer, 1980) to smooth the estimated word bigram probabilities.

3 The second stage: Unknown word identification

The second stage mainly concerns unknown words segmentation that remains unresolved in first stage. This section describes a hybrid algorithm for unknown word identification, which can incorporate word juncture model, word-formation patterns and contextual information. To avoid the complicated normalization of the probabilities of different dimensions, the simple superposition principle is also used in merging these models.

3.1 Word juncture model

Word juncture model score an unknown word by assigning word juncture type. Obviously, most unknown words appear as a string of known words after segmentation in first stage. Therefore, unknown word identification can be viewed as a process of re-assigning correct word juncture type to each known word pair in input. Given a known word string $W = w_1 w_2 \cdots w_n$, between each word pair $w_i w_{i+1} (1 \le i \le n-1)$ is a *word juncture*. In general, there are two types of junctures in unknown word identification, namely *word boundary* (denoted by t_B) and *non-word boundary* (denoted by t_N).

Let $t(w_i w_{i+1})$ denote the type of a word juncture $w_i w_{i+1}$, and $P_r(t(w_i w_{i+1}))$ denote the relevant conditional probability, then

$$P_r(t(w_i w_{i+1})) \stackrel{def}{=} \frac{Count(t(w_i w_{i+1}))}{Count(w_i w_{i+1})} \quad (3)$$

Thus, the word juncture probability $P_{CJM}(w_U)$ of a particular unknown word $w_U = w_i w_{i+1} \cdots w_j$ $(1 \le i \le j \le n)$ can be calculated by

$$P_{CJM}(w_U) = P_r(t_B(w_{i-1} w_i)) \times P_r(t_B(w_j w_{j+1})) \times \prod_{l=i}^{j-1} P_r(t_N(c_l c_{l+1})) \quad (4)$$

In a sense, *word juncture model* mirrors the affinity of known word pairs in forming an unknown word. For a word juncture (w_i, w_{i+1}), the larger the probability $P_r(t_N(w_i w_{i+1}))$, the more likely the two words are merged together into one new word.

3.2 Word-formation patterns

Word-formation pattern model scores an unknown word according to the probability of how each internal known word contributes to its formation. In general, a known word w may take one of the following four patterns while forming a word: (1) w itself is a word. (2) w is the beginning of an unknown word. (3) w is at the middle of an unknown word. (4) w appears at the end of an unknown word. For convenience, we use S, B, M and E to denote the four patterns respectively. Let $pttn(w)$ denote a particular pattern of w in an unknown word and $P_r(pttn(w))$ denote the relevant probability, then

$$P_r(pttn(w)) \stackrel{def}{=} \frac{Count(pttn(w))}{Count(w)} \quad (5)$$

Obviously, $\sum_{pttn} P_r(pttn(w)) = 1$. And $1 - P_r(S(w))$ is the word-formation power of the known word w.

Let $P_{pttn}(w_U)$ be the overall word-formation pattern probability of a certain unknown word $w_U = w_1 w_2 \cdots w_l$, then

$$P_{pttn}(w_U) = \prod_{w_i \in w_U} P_r(pttn(w_i)) \quad (6)$$

Theoretically speaking, a known word can take any pattern while forming an unknown word. But it is not even in probability for different known words and different patterns. For example, the word 性 (xing4, nature) is more likely to act as the suffix of words. According to our investigation on the training corpus, the character 性 appears at the end of a multiword in more than 93% of cases.

3.3 Hybrid algorithm for unknown word identification

Current algorithm for unknown word identification consists of three major components: (1) an unknown word extractor firstly extracts a fragment of known words $w_1 w_2 \cdots w_n$ that that may have unknown words based on the related word-formation power and word juncture probability and its left and right contextual word w_L, w_R from the output of the first stage. (2) A candidate word constructor then generates a lattice of all possible new segmentations $\{W_U \mid W_U = x_1 x_2 \cdots x_m\}$ that may involve unknown words from the extracted fragment. (3) A decoder finally incorporates word juncture model $P_{WJM}(W_U)$, word-formation patterns $P_{pttn}(W_U)$ and word bigram probability

$P_{bigram}(W_U)$ to score these candidates, and then applies the Viterbi algorithm again to find the best new segmentation $\hat{W}_U = x_1 x_2 \cdots x_m$ that has the maximum score:

$$\begin{aligned}\hat{W}_U &= \arg\max_{W_U}\{P_{pttn}(W_U) + P_{CJM}(W_U) + P_{bigram}(W_U)\} \\ &= \arg\max_{W_U}\{\sum_{i=1,\cdots,n}(P_{pttn}(x_i) + P_{CJM}(x_i) + P_r(x_i \mid x_{i-1}))\}\end{aligned} \quad (7)$$

where $x_0 = w_L$ and $x_{n+1} = w_R$. Let w_U denote any unknown word in the training corpus. If x_i is an unknown word, then $P_r(x_i \mid x_{i-1}) = \frac{\sum_{w_U} Count(x_{i-1} w_U)}{Count(x_{i-1})}$.

4 Experiments

Our system participated in both closed and open tests on Peking University corpora at the First International Chinese Word Segmentation Bakeoff. This section reports the results and discussions on its evaluation.

4.1 Measures

In the evaluation program of the First International Chinese Word Segmentation Bakeoff, six measures are employed to score the performance of a word segmentation system, namely test recall (*R*), test precision (denoted by *P*), the balanced F-measure (*F*), the out-of-vocabulary (OOV) rate for the test corpus, the recall on OOV words (R_{OOV}), and the recall on in-vocabulary (R_{iv}) words. OOV is defined as the set of words in the test corpus not occurring in the training corpus in the closed test, and the set of words in the test corpus not occurring in the lexicon used in the open test.

4.2 Experimental lexicons and corpora

As shown in Table 1, we only used the training data from Peking University corpus to train our system in both the open and closed tests. As for the dictionary, we compiled a dictionary for the closed test from the training corpus, which contained 55, 226 words, and used a dictionary in the open test that contained about 65, 269 words.

Items	# words in lexicon	# train. words	# test. words
Closed	55,226	1,121,017	17,194
Open	65,269	1,121,017	17,194

Table 1: Experimental lexicons and corpora

4.3 Experimental results and discussion

Items	F	R	P	OOV	ROOV	Riv
Closed	0.939	0.936	0.942	0.069	0.675	95.5
Open	0.937	0.933	0.941	0.094	0.762	95.0

Table 2: Test results on PK corpus

Segmentation speed: There are in all about 28,458 characters in the test corpus. It takes about 3.21 and 3.07 seconds in all for our system to perform full segmentation (including known word segmentation and unknown word identification) on the closed and open test corpus respectively, running on an ACER notebook (TM632XC-P4M). This indicates that our system is able to process about 531,925~556,182 characters per minute.

Results and discussions: The results for the closed and open test are presented in Table 2. We can draw some conclusions from these results.

Firstly, the overall performance of our system is very stable in both the closed and open tests. As shown in Table 2, the out-of-vocabulary (OOV) rate is 6.9% in the closed test and 9.4% in the open test. However, the overall test F-measure drops by only 0.2 percent in the open test, compared with the closed test.

Secondly, our approach can handle most unknown words in the input. As can be seen from Table 2, the recall on OOV words are 67.5% the closed-test and 76.2% in the open-test. Wang et al (2000) and Yao (1997) have proposed a character juncture model and word-formation patterns for Chinese unknown word identification. However, their approaches can only work for the unknown words that are made up of pure monosyllable character in that they are character-based methods. To address this problem, we introduce both word juncture model and word-based word-formation patterns into our system. As a result, our system can deal with different unknown words that consist of different known words, including monosyllable characters and multiword.

Although our system is effective for most ambiguities and unknown words in the input, it has its inherent deficiencies. Firstly, to avoid data sparseness, we do not differentiate known words and unknown words while estimating word juncture models and word-formation patterns from the

training corpus. This simplification may introduce some noises into these models for identifying unknown words. Our further investigations show that the precision on OOV words is still very low, i.e. 67.1% for closed test and 72.5% for open test. As a result, our system may yield a number of mistaken unknown words in the processing. Secondly, we regard known word segmentation and unknown word identification as two independent stages in our system. This strategy is obviously simple and more easily applicable. However, it does not work while the input contains a mixture of ambiguities and unknown words. For example, there was a sentence 中行长葛支行注重健身 in the test corpus, where, the string 中行长葛 is a fragment mixed with ambiguity and unknown words. The correct segmentation should be 中行/长葛/, where 中行(Zhonghang, the Bank of China) is a abbreviation of organization name, and 长葛 (Changge) is a place name. Actually, this fragment is segmented as 中/行长/葛/ in the first stage of our system. However, the unknown word identification stage does not have a mechanism to split the word 行长(Hangzhang, president) and finally resulted in wrong segmentation.

5 Conclusions

This paper presents a two-stage statistical word segmentation system for Chinese. In the first stage, word bigram model and Viterbi algorithm are applied to perform known word segmentation on input plain text, and then a hybrid approach is employed in the second stage to incorporate word bigram probabilities, word juncture model and word-based word-formation patterns to detect OOV words. The experiments on Peking University corpora have shown that the present system based on fairly simple word bigram and word-formation models can achieve a F-score of 93.7% or above. In future work, we hope to improve our strategies on estimating word juncture model and word-formation patterns and develop an integrated segmentation technique that can perform known word segmentation and unknown word identification at one time.

Acknowledgments

We would like to thank all colleagues of the First International Chinese Word Segmentation Bakeoff for their evaluations of the results and the Institute of Computational Linguistics, Peking University for providing the experimental corpora.

References

Fu, Guohong and Xiaolong Wang. 1999. Unsupervised Chinese word segmentation and unknown word identification. In: *Proceedings of NLPRS'99*, Beijing, China, 32-37.

Jelinek, Frederick, and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In: *Proceedings of Workshop on Pattern Recognition in Practice*, Amsterdam, 381-397.

Nie, Jian-Yuan, M.-L. Hannan and W.-Y. Jin. 1995. Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge. Communication of COLIPS, 5(1&2): 47-57.

Viterbi, A.J. 1967. Error bounds for convolutional codes and an asymmetrically optimum decoding algorithm. IEEE Transactions on Information Theory, IT-13(2): 260-269.

Wang, Xiaolong, Fu Guohong, Danial S.Yeung, James N.K.Liu, and Robert Luk. 2000. Models and algorithms of Chinese word segmentation. In: *Proceedings of the International Conference on Artificial Intelligence (IC-AI'2000)*, Las Vegas, Nevada, USA, 1279-1284.

Yao, Yuan. 1997. Statistics Based approaches towards Chinese language processing. Ph.D. thesis. National University of Singapore.

Zhang, Hua-Ping, Qun Liu, Hao Zhang, and Xue-Qi Cheng. 2002. Automatic recognition of Chinese unknown words based on roles tagging. In: *Proceedings of The First SIGHAN Workshop on Chinese Language Processing*, Taiwan, 71-77.