Natural Language Engineering **4** (4): 309–324. Printed in the United Kingdom

© 1998 Cambridge University Press

Splitting-merging model of Chinese word tokenization and segmentation

YUAN YAO and KIM TEN LUA

Department of Information Systems & Computer Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260

e-mail: {yaoyuan,luakt}@iscs.nw.edu.sg

(*Received 16 September 1997; Revised 3 March 1998*)

Abstract

Currently, word tokenization and segmentation are still a hot topic in natural language processing, especially for languages like Chinese in which there is no blank space for word delimitation. Three major problems are faced: (1) tokenizing direction and efficiency; (2) insufficient tokenization dictionary and new words; and (3) ambiguity of tokenization and segmentation. Most existing tokenization and segmentation methods have not dealt with the above problems together. To tackle the three problems in one basket, this paper presents a novel dictionary-based method called the Splitting-Merging Model (SMM) for Chinese word tokenization and segmentation. It uses the mutual information of Chinese characters to find the boundaries and the non-boundaries of Chinese words, and finally leads to a word segmentation by resolving ambiguities and detecting new words.

1 Introduction

A word is the smallest meaningful unit of any natural language system, and plays a primary role in almost all linguistic theories and language processing systems. Tokenization is the initial step in natural language processing which maps sentences from character strings into word strings. It leads to the word segmentation of sentences.

Currently, word tokenization and segmentation problems exist in almost all natural languages such as Chinese (Chen and Liu 1992; Bai, 1995), Japanese (Yosiyuki, Takenobu and Hozumi 1992), Korean (Yun, Lee and Rim 1995), German (Pachunke, Mertineit, Wothke and Schmidt 1992) and English (Garside, Leech and Sampson 1987), in diverse media forms such as continuous speech recognition and handwriting recognition, and in numerous applications such as translation, recognition, indexing and proof-reading. Depending on the resources applied, word tokenization and segmentation solutions can be broadly categorized as either orthography-oriented or dictionary-based. Orthography-oriented tokenization and segmentation are mainly tackled by using orthographic hints such as blank spaces as the word delimiter, and they are usually good enough for alphabetical languages such as English (Garside, Leech and Sampson 1987). On the other hand, if a full dictionary

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

is put into service, this is dictionary-based tokenization and segmentation. For graph-like languages such as Chinese and Japanese, in which there is traditionally no equivalent to the English blank space orthographic construction for word delimitation, the use of a dictionary for sentence tokenization and segmentation is inevitable (Webster and Kit 1992; Chen and Liu 1992; Chiang, Chang, Lin and Su 1992; Sun and T'sou 1995).

In Chinese text, the lack of an explicit token delimiter makes the otherwise simple problem of tokenization and segmentation surprisingly difficult. Many researchers have studied it, and have proposed a series of methods (Liang 1986; Jie, Liu and Liang 1991; Yeh and Lee 1991; Wu and Su 1993; Nie, Jin and Hannan 1994; Sproat, Shih, Gale and Chang 1994; Guo 1996*a*, 1996*b*) under a basic framework called the Principle of Maximization tokenization and segmentation. All of these methods can be categorized (Guo 1996*b*) into Shortest tokenization and segmentation, Forward Maximum tokenization and segmentation, Backward Maximum tokenization and segmentation, Forward-Backward tokenization and segmentation, Critical tokenization and segmentation and Profile tokenization and segmentation. However, Chinese word tokenization and segmentation still faces three major problems.

1. Tokenizing direction and efficiency.
2. Insufficient tokenization dictionary and new words.
3. Ambiguity of tokenization and segmentation.

The first problem exists in almost all of the above methods. For a greedy method like the Forward or Backward Maximum tokenization methods, different tokenizing directions may give different results for the same sentence, i.e. generate tokenization ambiguities. Any one tokenization ambiguity has an accumulated effect, and will affect the tokenizing thereafter. For example, a sentence

便沿海岸线传布了两千公里之远.

may be tokenized and segmented to

便/沿海/岸/线/传布/了/两千/公里/之/远/.

by the Forward Maximum tokenization method. Clearly, it is an incorrect segmentation, for the correct one should be

便/沿/海岸线/传布/了/两千/公里/之/远/.

On the other hand, compared to the greedy methods, exhaustive tokenization gives all of the probable tokens of words. In such methods, the tokenization efficiency falls down exponentially as the sentence length increases, until it becomes unmanageable. Thus, the tokenizing direction and tokenizing mechanism are the important factors which need to be considered carefully.

Because Chinese word tokenization and segmentation are based on a tokenization dictionary, they rely heavily on the completeness and consistency of the dictionary. When the tokenization dictionary in insufficient, the segmentation performance will reduce greatly if unknown words exist in the sentences being tokenized. However, it is impossible to set up or find a complete dictionary: (a) any dictionary comes from

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

a certain corpus, no matter how large or small, and is thus limited by the coverage of the corpus; (b) languages develop constantly and new words never stop being added. Thus, the latter problem will always exist in dictionary-based tokenization and segmentation.

For the third problem mentioned above, the ambiguity of tokenization and segmentation means that a sentence being tokenized may have more than one choice of segmentation in different sentence meanings. As a typical and simple example[1], the Chinese sentence

†自我才能得到存全

may have the following different meaningful segmentations:

自我/才能/得到/存全/.

自我/才/能/得到/存全/.

There are two types of ambiguities, (a) *combinational ambiguity*, and (b) *inter-cross ambiguity*. For a character string $C = c_1 c_2 \ldots c_i UVWc_j \ldots c_n$, in which U, V and W are its sub-strings while c_i is a character, if UV, U and V can be tokenized as three words by the tokenization dictionary, respectively, then UV is a combinationally ambiguous string. Tor example, in the above sentence '自我才能得到存全.', let $U =$ 才 (*freshly*) and $v =$ 能 (can, energy), then $UV =$ 才能 (ability) is a combinationally ambiguous string. On the other hand, if the sub-string UVW can be segmented into either UV/W or U/VW, then UVW is an inter-cross ambiguous string. For example, in the sentence '高兴得搓着手说.', let $U =$ 搓 (twine), $V =$ 着 (touch, have done) and $W =$ 手 (hand), then $UVW =$ 搓着手 is an inter-cross ambiguous string, for it can be segmented into 搓/着手 and 搓着/手.

Currently, most existing tokenization and segmentation methods have not dealt with the above three problems together for Chinese word segmentation. To tackle all of the problems at the same time, this paper presents a novel dictionary-based method called the Splitting-Merging Model (SMM) for Chinese word tokenization and segmentation. It uses the mutual information of Chinese characters to find the boundaries and non-boundaries of Chinese words, and leads to a word segmentation by resolving ambiguities and detecting new words.

2 Mutual information of characters

Mutual information is quite a good measurement for describing the relationships of pairs of characters in a string, and will be used here to detect the boundaries and non-boundaries of words in a character string.

Definition 1. For a character string $C = c_1 c_2 \ldots c_n$, if c_i and c_{i+1} are two successive characters in C where $0 < i < n-1$, then the mutual information of the characters c_i and c_{i+1} is

$$I(c_i; c_{i+1}) = \log \frac{P(c_i, c_{i+1})}{P(c_i)\, P(c_{i+1})}. \tag{1}$$

[1] The sentence is selected from the CKIP corpora (CKIP 1995).

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Since the characters c_i and c_{+1} can also be considered as two sub-strings of C, the evaluation of probabilities in the above equation can be obtained from the following definition.

Definition 2. If C_i and $_{i+1}$ are two successive sub-strings of C, then the probabilities $P(C_i, C_{i+1})$ and $P(C_i)$ are

$$P(C_i, C_{i+1}) = \frac{count(C_i\, C_{i+1})}{\sum_{k=1}^{n-1} count(C_k\, C_{k+1})}, \tag{2}$$

$$P(C_i) = \frac{count(C_i)}{\sum_{k=1}^{n} count(C_k)}. \tag{3}$$

For a pair of characters c_i and c_{i+1}, the smaller the $I(c_i; c_{i+1})$ is, the more mutually independent the characters c_i and c_{i+1} are. On the contrary, the larger the $I(c_i; c_{i+1})$ is, the more mutually dependent the characters c_i and c_{i+1} are. The mutual information $I(c_i; c_{i+1})$ describes the closeness of two characters c_i and c_{i+1}, and it is the same for the pair of character strings C_i and C_{i+1}. This suggests that in a character string C, the boundaries of words will be in between the character pairs or sub-string pairs that have a little mutual information, while the non-boundaries will be in between the character pairs or sub-string pairs that have a lot of mutual information. The tokenization and segmentation method presented in this paper is based on this.

3 Data source

As we know, dictionary-based tokenization and segmentation relies greatly on the completeness of the dictionary. Thus, the larger the corpus the diction covers, the few problems with new words will arise in tokenization and segmentation. The dictionary used here comes from the Frequency Dictionary of Modern Chinese Words (1988) and the CKIP corpora (CKIP 1995).

The frequency Dictionary of Modern Chinese words (FDMCW) not only collects 46,520 distinct words[2] from various Chinese texts which contain about 25 million words in all, but also gives the frequency of each distinct word shown in the texts. It covers the following main types of texts published in mainland China from the 1920s to 1890s: natural and social science subjects divided into 35 categories; literature and art works; daily conversions; and newspaper editorials. Table 1 shows the word distribution of the FDMCW.

The CKIP corpora, developed by the Taiwan Chinese Knowledge Information Processing Group, are the tagged corpora. The corpora contain about 280,000 sentences[3] from informal spoken language to formal written language, and cover various types of texts in the areas of philosophy, science, sociology, art, life and

[2] Only the words with a frequency of more than five are collected.
[3] 'Sentences statistics are based on the punctuation ',', '.', '?', '!', ';' and '...'.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Table 1. *Word distribution of the FDMCW*

Word length	1	2	3	4	5	6	7	Total
Number of words	5070	31187	5125	4566	382	144	46	46520

Table 2. *Word distribution of the CKIP Corpora*

Word length	Number of words	%
1	3560	3·94
2	45989	50·90
3	25676	28·43
4	9936	11·00
5	2063	2·28
6	962	1·07
7	735	0·81
8	452	0·50
9	316	0·35
10	238	0·26
11	203	0·22
12	186	0·21
13	156	0·17
14	129	0·14
15	110	0·12
16[a]	1	0·00
Total	90312	100

[a] The words with a length of more than 15 are foreign words. All of them are here defined as a special word with a length of 16.

literature. Here, a word frequency library based on the corpora is set up statistically. Table 2 shows its distribution.

To form the tokenization dictionary used here, the FDMCW and the word frequency library of the CKIP corpora have been combined, giving rise to 100,908 distinct words. Thus, the dictionary covers the largest number of texts from both mainland China and Taiwan. Since this dictionary contains the frequency of each word, the evaluation of mutual information of characters or character strings becomes quite easy, according to Definitions 1 and 2. In the mutual information calculation, for any a new character or a new character string, its frequency is simply set to one.

4 The splitting-merging model

The Splitting-Merging Model (SMM) presented here uses mutual information to perform splitting and merging of character strings. Before describing the SMM algorithm, definitions of the *splitting* and *merging* operations are first given.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

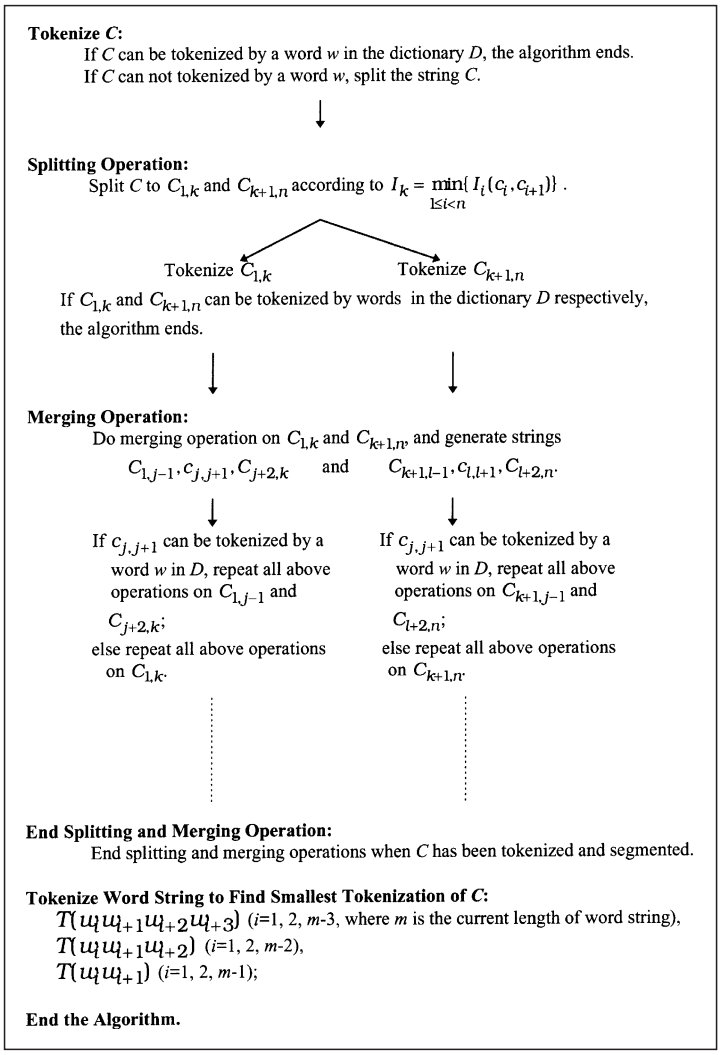


Fig. 1. The SMM algorithm.

Definition 3. Consider a character string $C_{1,n} = c_1 c_2 \ldots c_n$, where the subscript of $C_{1,n}$ denotes the range of the string, and the splitting operation $S(C_{1,n} \mid C_{1,k}, C_{k+1,n})$ of $C_{1,n}$ denotes the range in between k and $k+1$ based on the mutual information $I_k = \min_{1 \leqslant i < n}\{I_i(c_i, c_{i+1})\}$, and cuts the string $C_{1,n}$ into two sub-strings $C_{1,k}$ and $C_{k+1,n}$.

Definition 4. Consider a character string $C_{1,n} = c_1 c_2 \ldots c_n$. A merging operation $M(C_{1,n} \mid C_{1,k-1}, C_{k,k+1}, C_{k+2,n})$ generates a non-boundary in between k and $k+1$ based on the mutual information $I_k = \max_{1 \leqslant i < n}\{I_i(c_i, c_{i+1})\}$, and merges two characters c_k and c_{k+1} into an inseparable character-like string $C_{k,k+1}$. At the same time, the merging operation also generates two other sub-strings $C_{1,k-1}$ and $C_{k+2,n}$.

According to the above definitions, both the splitting and merging operations separate a character string into two or three sub-strings by finding boundaries and

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

non-boundaries. Clearly, if these operations are carried out on a character string alternately, the string will eventually be tokenized and segmented.

Definition 5. Consider a character string $C = c_1 c_2 \ldots c_n$ and a tokenization dictionary D. If $\exists w \in D$ and $w = c_1 c_2 \ldots c_n$, then the word w is defined as a token of the string C generated by tokenizations $T(C)$.

The SMM is a series of successive dictionary-based tokenizations $T_i(C_i)$ $(i = 1, 2, \ldots)$ on sub-strings C_i of a character string C, alternately generated by the splitting and merging operations. The detailed algorithm is described below. Figure 1 shows a partial procedure of splitting and merging operations, and figure 2 gives an example of sentence segmentation using SMM.

Step 0: Read in an n-character string $C = c_1 c_2 \ldots c_n$, and give each Chinese character c_i an initial token '0' and non-Chinese character an initial token'1';
Step 1: Tokenize C;
Step 2: IF $T(C) = w$, go to step 13;
Step 3: ELSE Split C by using splitting operation $S(C \mid C_{1,k}, C_{k+1,n})$;
  Step 4: Tokenize $C_{1,k}$ and $C_{k+1,n}$;
  Step 5: IF $T(C_{1,k}) = w_{1,k}$ and $T(C_{k+1,n}) = w_{k+1,n}$, go to step 13;
  Step 6: RLDR IG $T(C_{1,k}) = 0$, do merging operation
    $M(C_{1,k} \mid C_{1,j-1}, c_{j,j+1}, C_{j+2,k})$;
    Step 7: Tokenize $c_{j,j+1}$;
      Step 8: If $T(c_{j,j+1}) = w_{j,j+1}$, $c_{j,j+1}$ is tokenized, let $C = C_{1,j-1}$ and $C = C_{j+2,k}$ respectively, repeat steps from step 1 to step 10 until all character strings are tokenized.
      Step 9: ELSE cancel the merging operation on c_j and c_{j+1}, let $C = C_{1,k}$ and $C = C_{k+1,n}$ respectively, repeat steps from step 1 to step 10 until all character strings are tokenized.
  Step 10: ELSE IF $T(C_{k+1,n}) = 0$, let $C_{1,k} = C_{k+1,n}$, do step 6–9;
Step 11: For the tokenization $w_1 w_2 \ldots w_m$ of $C = c_1 c_2 \ldots c_n$, tokenize $T(w_i w_{i+1} w_{i+2} w_{i+3})\,(i = 1, 2, m-3)$, $T(w_i w_{i+1} w_{t+2})$ $(i = 1, 2, m-2)$, and $T(w_i w_{i+1})$ $(i = 1, 2, m-1)$;
Step 12: IF anyone of above word strings can be tokenized by a word in D, merge this string to the word;
Step 13: Output the tokenization and segmentation result;
Step 14: END.

Clearly, the SMM is a greedy algorithm without any tokenization directions. It successively separates a string at locations only depending on word boundaries until the string cannot be cut or has been completely tokenized. Thus, the first problem mentioned in section 1 is naturally resolved. Step 11 of the algorithm is designed to achieve maximum tokenization by merging the available words to longer words. On the other hand, the algorithm here is unable to perform word disambiguation and new word detection. These problems will be discussed in later sections.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Sentence: 这些组织已开始和ＡＮＳ竞争.

Initialization:

Number	Character String	Token	Entropy	Mutual Information
0	这	0	2.96554e-02	6.383211
1	些	0	2.19280e-03	-0.092952
2	组	0	2.25544e-03	9.989749
3	织	0	8.74686e-04	-0.504000
4	已	0	7.90700e-03	-2.173055
5	开	0	3.80612e-03	9.371429
6	始	0	8.03600e-04	-2.507048
7	和	0	4.44103e-02	
8	ＡＮＳ	1	8.61062e-06	
9	竞	0	1.58093e-04	10.374996
10	争	0	9.54280e-04	

The split strings: 0— 7, 9— 10

Tokenizing and Splitting:

0	这	0	2.96554e-02	6.383211
1	些	0	2.19280e-03	-0.092952
2	组	0	2.25544e-03	**9.989749**
3	织	0	8.74686e-04	-0.504000
4	已	0	7.90700e-03	**-2.173055**
5	开	0	3.80612e-03	9.371429
6	始	0	8.03600e-04	-2.507048
7	和	0	4.44103e-02	
8	ＡＮＳ	1	8.61062e-06	
9	竞争	1	4.70350e-04	

The split strings: 0— 6, 7— 7

Merging and Tokenizing:

0	这	0	2.96554e-02	6.383211
1	些	0	2.19280e-03	
2	组织	1	4.23224e-03	
3	已	0	7.90700e-03	-2.173055
4	开	0	3.80612e-03	9.371429
5	始	0	8.03600e-04	
6	和	1	4.44103e-02	
7	ＡＮＳ	1	8.61062e-06	
8	竞争	1	4.70350e-04	

The split strings: 0— 1, 3— 5

Tokenizing and Splitting:

0	这些	1	6.35032e-03	
1	组织	1	4.23224e-03	
2	已	0	7.90700e-03	**-2.173055**
3	开	0	3.80612e-03	**9.371429**
4	Ê¼	0	8.03600e-04	
5	和	1	4.44103e-02	
6	ＡＮＳ	1	8.61062e-06	
7	竞争	1	4.70350e-04	

The split strings: 2— 2, 3— 4

Merging and Tokenizing:

0	这些	1	6.35032e-03	
1	组织	1	4.23224e-03	
2	已	1	7.90700e-03	
3	开始	1	3.83344e-03	
4	和	1	4.44103e-02	
5	ＡＮＳ	1	8.61062e-06	
6	竞争	1	4.70350e-04	

The Tokenization and Segmentation Result:

这些 / 组织 / 已 / 开始 / 和 / ＡＮＳ / 竞争

Fig. 2. An example of word segmentation using SMM.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

5 The experiment

Three types of tests are conducted in the experiment: (A) inside test, (B) outside test with a few new words, and (C) outside test with many new words. For tests A and B, 7530 sentences with lengths from 5–10 are randomly selected from the CKIP corpora, depending on the sentence distribution of the sentence length. All the words in these sentences and their parameters are included and evaluated in the tokenization dictionary for test A. So test A is an inside test. For test B, the tokenization dictionary is re-evaluated to delete the efforts of these words. After this, upon comparison with the dictionary and the words in the test sentences, 122 new words arise, with a new words rate of 0·23 % (122 new words/52,720 words in the test sentences) of all the words in the test sentences, and 1·62 % (122 new words/7530 test sentences) of the test sentences. Thus, this is an outside test with a few new words. For test C, 1200 sentences are randomly selected from the Singapore Primary School Chinese Text Books (about 8400 Chinese sentences from Primary 1 to Primary 6) Corpus. The rate of new words in sentences is 8·08 %, which means there are eight new words per 100 sentences.

The algorithm is implemented in the C programming language on a Digital Alpha Server 8400 5/350 with four processors. Because mutual information is dynamically computed based only on the tokenization dictionary and Definition 1 mentioned in sections 2 and 3, the algorithm does not require a very large space, as is usual when mutual information of Chinese characters and character strings is involved. It needs about 13·5 MB of memory for the experiment. The time token for tokenizing and segmenting a character string in the experiment is from 0·02–3·5 ms, depending on the string length.

An automatic detector is designed to pick up the incorrectly segmented sentences by comparing each segmented sentence with the original sentence in the CKIP corpora (all sentences in the CKIP corpora are segmented and tagged). Tables 3–5 show the distributions of incorrectly segmented sentences in tests A, B and C, respectively.

Test A is an inside test where the problem of new words does not exist. There are three factors that will result in incorrect tokenization and segmentation. From Table 3, the most important factor is the inconsistency of the tokenization dictionary, and then type (b) and (a) ambiguities. The inconsistency of the tokenization dictionary results from the combination of the FDMCW and the word library of the CKIP corpora, because of the inconsistencies in word segmentation between the two systems (mainland China and Taiwan). For example, the string '不应该 (should not)' is tokenized as a word '不应该 (should not)' in the CKIP corpora, but two words '不 (not)' and '应该 (should)' in FDMCW. The simple combination of the word library and the FDMCW will collect three words '不应该', '不' and '应该' in the tokenization dictionary, and results in inconsistencies. Thus, the inconsistent dictionary needs to be improved. Here, the improvement is based on the FDMCW, which means that if any inconsistent word segmentation occurs, the word segmentation of the FDMCW will be used. For example, the word '不应该 ' will not occur in the improved dictionary. From Table 3, it is found that the total sentence error rate is reduced from 3·51 % to

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Table 3. *Distribution of incorrectly segmented sentences in test A*

Sentence length	5	6	7	8	9	10	Total
Number of sentences	1720	1640	1400	1150	920	700	7530
Ambiguity (a)	4	8	19	4	3	8	46
Ambiguity (b)	6	30	16	23	12	5	92
Dictionary inconsistent	15	20	30	28	11	22	126
Total incorrectly segmented sentences	25	58	65	55	26	35	264
Error rate (Sentence) (%)	1·45	3·54	4·64	4·78	2·83	5·00	3·51
Error rate (improved) dictionary) (%)	0·58	2·32	2·50	2·35	1·63	1·86	1·83

Table 4. *Distribution of incorrectly segmented sentences in test B*

Sentence length	5	6	7	8	9	10	Total
Number of sentences	1720	1640	1400	1150	920	700	7530
Ambiguity (a)	4	8	19	4	3	8	46
Ambiguity (b)	6	30	16	23	12	5	92
New words	27	21	24	23	18	9	122
Total incorrectly segmental sentences	37	59	59	50	33	22	260
Error rate (Sentence) (%)	2·15	3·60	4·21	4·35	3·59	3·14	3·45

Table 5. *Distribution of incorrectly segmented sentences in test C*

Number of sentences	Ambiguity (a)	Ambiguity (b)	New words	Total incorrectly segmented sentences	Error rate (Sentence) (%)
1200	6	24	97	127	10·58

1·83% in test A. This is a very encouraging result in using SMM, though two kinds of segmentation ambiguities still exist. The improved dictionary is also used in tests B and C.

Tests B and C are outside tests with a few new words and many new words, respectively. From Tables 4 and 5, it can be seen that the problem of new words is the most important factor of all in sentence segmentation. This means that the more incomplete the dictionary is, the more likely it is that the sentences will be incorrectly segmented. Thus, the detection of new words becomes crucially important in lowering the error rate in sentence segmentation. The following sections discuss the disambiguation of sentence segmentation and the detection of new words.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

6 Disambiguation

In two ambiguities mentioned in the section 1, the first type (combinational ambiguity) is much more serious than the second (inter-cross ambiguity), because the disambiguation of combinational ambiguity depends not only upon the context words, but also upon whole sentences. Sometimes even human beings cannot decide consistently on this. The sentence '自我才能得到存全.' in section 1 is a good example. Another example is the sentence '我怎么没有用.'. It may have the following segmentations:

我/怎么/没有用/.

我/怎么/没有/用/.

Fortunately, this ambiguity does not occur frequently. Its effects can also be found in the tables above.

Here, disambiguation is conducted mainly for the inter-cross ambiguity. Considering the ambiguously segmented sentences in tests B and C, the inter-cross ambiguity can be classified into the following sub-types:

Definition 6. Suppose there is a character string $C = c_1 c_2 \ldots WXYZ \ldots c_n$, where W, X, Y and Z are the sub-strings of C. Two kinds of the inter-cross ambiguity are defined as

Ambiguity (b1), when string XYZ can be segmented into X/YZ or XY/Z;
Ambiguity (b2), when string $WXYZ$ can be segmented into WX/YZ or $W/XY/Z$.

For example, the string '搓着手' has type b1 ambiguity, for it may be segmented into '搓/着手' or '搓着/手', and string '大学校长' has type b2 ambiguity, for it may be segmented into '大学/校长' or '大/学校/长'.

For disambiguating ambiguity b1, the bigram probabilities $P(YZ|X)$ and $PP(Z|XY)$ are used to determine how to cut XYZ. For disambiguating ambiguity b2, the segmentation WX/YZ will always be used. Because any ambiguity in the SMM algorithm will occur after an unsuitable merging operation or splitting operation, steps 3 and 6 of the algorithm are improved here to deal with the above disambiguations, by check the context characters or character strings of the strings observed. The disambiguation results of tests B and C are shown in Tables 6 and 7 respectively.

Table 6. *Disambiguation result of test B*

Sentence length	5	6	7	8	9	10	Total
Number of sentences	1720	1640	1400	1150	920	700	7530
Ambiguity (a)	4	8	19	4	3	8	46
Ambiguity (b)	1	6	4	5	3	1	20
New words	27	21	24	23	18	9	122
Total incorrectly Segmented sentences	32	35	47	32	24	18	188
Error rate (sentence) (%)	1·86	2·13	3·36	2·78	2·61	2·57	2·50

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Table 7. *Disambiguation result of test C*

Number of sentences	Ambiguity (a)	Ambiguity (b)	New words	Total incorrectly segmented sentences	Error rate (sentence) (%)
1200	6	5	97	108	9·00

From Tables 6 and 7, it can be seen that ambiguity b1 still exists, but is much reduced. The results after disambiguation are encouraging.

7 Detection of new word

As mentioned in section 1, new words are a major problem in Chinese word tokenization and segmentation. The experiment above points out that they will affect any segmentation results a great deal. Thus, the detection of new words is necessary to word segmentation.

Most new Chinese words (outside the dictionary), such as Chinese names, names of places, transliterated names (from foreign language), etc., are built on quite random word-formation rules, and lack heuristic marks for identification, e.g. the capital letter, as in English. This makes the detection of new words quite difficult. Fortunately, in the above experiment, it has been found that most new words in the segmented sentences are cut into single-character-words by the SMM algorithm, as in the following (underlined):

棋 / 樟 / 山 / 现在 / 是 / 个 / 度假 / 胜地 / 。
乌 / 南 / 迪 / 还 / 跳到 / 椅子 / 上 / ,
薛 / 尔 / 思 / 桥 / 啊 / ,
屠 / 妖 / 节 / 到来 / 之前 / ,
组 / 屋 / 的 / 生活 / 这么 / 充实 / 、 / 美好 / ,
伤兵 / 们 / 都 / 很 / 感激 / 南 / 丁 / 格 / 尔 / 。

Therefore, the detection of new words becomes a combination of these successive single-character-words within a focused scope. We should be careful, however, not to form non-new-words through wrong combinations.

Under the morphological analysis of Chinese word-formation, it is found that there are four word-formation patterns for forming Chinese words:

(a) independent (used independently in a sentence),
(b) head (used as a prefix to other words),
(c) tail (used as a suffix to other words),
(d) middle (used as a middle character for other words).

A character may employ any one of the above word-formation patterns in forming words, but with a certain probability that can be easily evaluated from a dictionary,

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

as shown in the following definition. Moreover, the above patterns can also be extended to words to form larger words.

Definition 7. Let $P(pttn(i)\,|\,c)$ denote the conditional probability of a certain word-formation pattern for character c, where *pttn* is the state that c is in, and i is a word-formation pattern. Then $P(pttn(i)\,|\,c)$ is defined as

$$P(pttn(a)\,|\,c) = \frac{count(c_is_the_word)}{count(c)} \tag{4}$$

$$P(pttn(b)\,|\,c) = \frac{count(c_is_the_prefix)}{count(c)} \tag{5}$$

$$P(pttn(c)\,|\,c) = \frac{count(c_is_the_suffix)}{count(c)} \tag{6}$$

$$P(pttn(d)\,|\,c) = \frac{count(c_is_in_the_middle_of_words)}{count(c)}, \tag{7}$$

where

$$P(pttn(a)\,|\,c) + P(pttn(b)\,|\,c) + P(pttn(c)\,|\,c) + P(pttn(d)\,|\,c) = 1{\cdot}0.$$

For example, in the CKIP word library, the frequencies of the character '是' in patterns (a), (b), (c) and (d) are 25,410, 885, 5761 and 138, respectively. Then, according to the above definition,

$$P(independent\,|\,是) = 0{\cdot}7893$$

$$P(head\,|\,是) = 0{\cdot}0275$$

$$P(tail\,|\,是) = 0{\cdot}1789$$

$$P(middle\,|\,是) = 0{\cdot}0043.$$

Based on these patterns, a word may be formed by characters or single-character-words in one of the following obvious head-middle-tail structures:

(1) head + tail,
(2) head + independent + independent + … + tail,
(3) head + independent,
(4) independent + tail,
(5) head + middle + middle + … middle + tail.

Thus, to detect a new word, it is necessary to determine the probability of a character string or a single-character-word string being one of the above structures. Here, the probability of a structure $P(s_k)$ defined below is used to assist in the detection of a new word.

Definition 8. Consider a single-character-word string $w_{1,n} = w_1 w_2 \ldots w_n$. There may exist $n \times 4$ different structures. Let $s_k = pttn_1(i_{1k})\, pttn_2(i_{2k}) \ldots pttn_n(i_{nk})$ $(1 \leqslant k \leqslant n \times 4)$ denote a structure, and $S = \{s_1, s_2 \ldots, s_{n\times 4}\}$ denote the structure set, where $pttn_j(i_{jk})$ is

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Table 8. *Post-processing result of test B*

Sentence length	5	6	7	8	9	10	Total
Number of sentences	1720	1640	1400	1150	920	700	7530
Ambiguity (a)	4	8	19	4	3	8	46
Ambiguity (b)	1	6	4	5	3	1	20
New words	5	3	4	5	4	2	23
Total incorrectly segmented sentences	10	17	27	14	10	11	89
Error rate (sentence) (%)	0·58	1·04	1·93	1·22	1·09	1·57	1·18

Table 9. *Post-processing result of test C*

Number of sentences	Ambiguity (a)	Ambiguity (b)	New words	Total incorrectly segmented sentences	Error rate (sentence) (%)
1200	6	5	12	22	1·92

the pattern the single-character-word w_j is in, and i_{jk} is its type, then the probability of a structure $P(s_k)$ is defined as

$$P(s_k) = \prod_{j=1}^{n} P(pttn_j(i_{jk}) \mid w_j). \tag{8}$$

Definition 9. For a single-character-word string $w_{1,n} = w_1 w_2 \ldots w_n$ and the structure s_m where $P(s_m) = \max_{1 \leqslant k \leqslant n \times 4}\{P(s_k)\}$, if s_m is one of the above head-middle-tail structures, the string $w_{1,n} = w_1 w_2 \ldots w_n$ is a new word.

Based on Definition 9, a new word detection strategy is proposed to post-process the results of the SMM.

Step 0: Find a single-character-word string $W = w_1 w_2 \ldots w_n$ from a segmented sentence, if it exists.

Step 1: Skip the string if it has only one word and go to the end.

Step 2: Combine the single-character-words in the string into a word if the words in the string have the reduplicated form, e.g. 重重, 怕怕, 打打杀杀, etc., and go to the end.

Step 3: Find all possible structures of the string and their probabilities according to Definitions 7 and 8.

Step 4: Combine the single-character-words in the string into a new word if the new word exists according to Definition 9. Otherwise, reduce the string $W = w_1 w_2 \ldots w_n$ by one word, namely $W = w_2 \ldots w_n$ and $W = w_1 w_2 \ldots w_{n-1}$, and repeat Steps 1–4 to them.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Step 5: Check the sentence up to find another single-character-word string $W = w_1 w_2 \ldots w_n$ until it does not exist, and go to the end.
Step 6: Repeat steps 1–5.
Step 7: The end.

Tables 8 and 9 show the post-processed results for new word detection on tests B and C. From these, it can be seen that about 80 % of the new words can be detected by the above strategy, and the final experimental results are quite good. The correct rate of sentence tokenization and segmentation can reach over 98 %, even if the percentage of new words is about 8 %.

8 Conclusion

Focusing on the three main problems of word tokenization and segmentation, this paper presents a novel model called SMM and solutions to disambiguation and new word detection for Chinese word segmentation. While tokenizing and segmenting words in a sentence, the mutual information-based splitting and merging operations are alternately used on the sentence to detect boundaries and non-boundaries of words, which will eventually lead to a segmentation of the sentence. At the same time, the bigram-based solution to the disambiguation, is also used to improve the segmentation. Finally, additional post-processing is conducted on the segmented sentence for a new word detection. New word detection is based on the four word-formation patterns and the head-middle-tail structures.

The large corpora-based experiment gives quite encouraging results here, showing that the SMM is an effective model for Chinese word tokenization and segmentation.

References

Bai, S. H. (1995) An integrated model of Chinese word segmentation and part of speech tagging. In *Advances and Applications on Computational Linguistics*, L. W. Chen and Q. Yuan (eds.), pp. 56–61. Tsinghua University Press.

Chen, K. J. and Liu, S. H. (1992) Word identification for mandarin Chinese sentences. *Proceedings 14th International Conference on Computational Linguistics* (*COLING-92*), Nantes, France, pp. 101–107.

Chiang, T. H., Chang, J. S., Lin, M. Y. and Su, K. Y. (1992) Statistical models for word segmentation and unknown word resolution. *Proceedings ROC Computational Linguistics Conference* (*ROCLING*) *V*, Taiwan, pp. 123–146.

Garside, R. G., Leech, G. N. and Sampson, G. R. (1987) *The Computational Analysis of English: A Corpus-based Approach*. London: Longman.

Guo, J. (1996*a*) Critical tokenization and its properties. Submitted. (Online available at URL: http://sunzi.iss.nus.sg:1996/guojin/papers/)

Guo, J. (1996*b*) Profile tokenization and its properties. Submitted. (Online available at URL: http://sunzi.iss.nus.sg:1996/guojin/papers/)

Jie, C. Y., Liu, Y. and Liang, N. Y. (1991) On the methods of Chinese automatic segmentation. *Journal of Chinese Information Processing* **3** (1): 1–9.

Liang, N. Y. (1986) On computer automatic word segmentation of written Chinese. *Journal of Chinese Information Processing* 1 (1).

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press

Nie, J. Y., Jin, W. Y. and Hannan, M. L. (1994) A hybrid approach to unknown word detection and segmentation of Chinese. *Proceedings of International Conference on Chinese Computing 1994* (*ICCC-94*), Singapore, pp. 326–335.

Pachunke, T., Mertineit, O., Wothke, K. and Schmidt, R. (1992) Broad coverage automatic morphological segmentation of German words. *Proceedings 14th International Conference on Computational Linguistics* (*COLING-92*), Nantes, France, pp. 1218–1222.

Sun, M. S. and T'sou, B. (1995) Ambiguity resolution in Chinese word segmentation. *Proceedings 10th Pacific Asia Conference on Language, Information and Computation* (*PACLIC-95*), Hong Kong, pp. 121–126.

Webster, J. J. and Kit, C-Y. (1992) Tokenization as the initial phase in NLP. *Proceedings 14th International Conference on Computational Linguistics* (*COLING-92*), Nantes, France, pp. 1106–1110.

Wu, M. W. and Su, K. Y. (1993) Corpus-based automatic compound extraction with mutual information and relative frequency count. *Proceedings ROC Computational Linguistics Conference* (*ROCLING*) *VI*, Taiwan, pp. 207–216.

Yeh, C. L. and Lee, H. J. (1991) Rule-based word identification for mandarin Chinese sentences – a unification approach. *Computer Processing of Chinese and Oriental Languages* 5 (2): 97–118.

Yosiyuki, K., Takenobu, T. and Hozumi, T. (1992) Analysis of Japanese compound nouns using collocation information. *Proceedings 14th International Conference on Computational Linguistics* (*COLING-92*), Nantes, France, pp. 865–869.

Yun, B. H., Lee, H. and Rim, H. C. (1995) Analysis of Korean compound nouns using statistical information. *Proceedings 1995 International Conference on Computer Processing of Oriental Language* (*ICCPOL-95*), Honolulu, Hawaii, pp. 76–79.

Chinese Knowledge Information Processing Group (CKIP) (1995) *中央研究院平衡语料库的内容与说明*.. Technical Report 95-02, Institute of Information Science, Academia Sinica, Taiwan.

刘源, 梁南元, 王德进, 张社英, 杨铁英, 揭春雨, 孙伟.*现代汉语常用词词频词典*. 1988 中国宇航出版社.

https://doi.org/10.1017/S1351324998002058 Published online by Cambridge University Press