# An Algorithm Rapidly Segmenting Chinese Sentences into Individual Words

Zhibin Xiong

*College of Creative Arts Hainan Tropical University, Sanya 572022, Hainan, China*

**Abstract.** This paper proposes an improved Trie tree structure. The tree node records the position information of the characters participating in the word formation, and the child node uses the hash search mechanism. On this basis, the forward maximum matching algorithm of Chinese word segmentation is optimized. In the process of word segmentation, the automaton mechanism is used to judge whether it constitutes the longest word, and the problem that the forward maximum matching algorithm needs to adjust the string according to the word length is solved. The algorithm time complexity is 1.33, and the comparison test results show that there is a fast word segmentation speed. The forward maximum matching algorithm based on the improved Trie tree structure improves the Chinese word segmentation speed, especially when the dictionary structure needs to be updated in real time.

**Keywords:** Natural language processing; Chinese word segmentation; Forward maximum matching algorithm.

## 1 Introduction

The forward maximum matching algorithm in Chinese automatic word segmentation is a highly practical word segmentation algorithm. The algorithm only needs a dictionary and is divided according to the principle of "long word first". Literature [1] introduces the forward maximum matching algorithm. The basic idea of the algorithm is: assuming that the number of Chinese characters of the longest term in the word segmentation dictionary is n, each time a substring W of length n is intercepted from the string S to be sliced as a matching string. When looking up the word segmentation dictionary, if the match is successful, the W is segmented from S as a word. If the match is unsuccessful, subtract a word from the tail of W. The previous matching process with a substring of length n-1 can be repeated until the match is successful. According to statistics, one word and two words account for 96.4% of Chinese vocabulary.[1] According to the longest word in the dictionary segmentation will cause a lot of time waste, and search efficiency is low.

Dictionary structure is the key technology of dictionary segmentation algorithm, which directly affects the performance of segmentation algorithm[2]. Literature [3] investigates three typical lexicographical mechanisms through experiments: whole word dichotomy, Trie tree and verbatim dichotomy. It is concluded that the Trie tree and the verbatim dichotomy time efficiency are roughly equivalent. Using the forward maximum matching algorithm, the search speed of the Trie tree and the verbatim dichotomy is 15.3 times that of the whole word.

Literature [4] proposes an improved Chinese word segmentation forward maximum matching algorithm. The dictionary consists of the first word hash table, the morpheme table, and the dictionary body. According to the length of entries recorded in the morpheme table, the idea of dynamically determining the length of the text to be processed is dynamically determined. This method reduces the number of matches, and still does not completely overcome the restriction of word length. Literature [5] proposes an improved fast word segmentation algorithm. The dictionary consists of a first word hash table and a dictionary text. The entries are arranged according to the Chinese code, the first word is hashed, and the end part is found by binary search. The time complexity of this method is 1.66. Literature [6] proposes a word segmentation method based on automaton, and dictionary structure is organized by binary tree, according to the status of Chinese characters in the dictionary to match the implementation of fast search. The experimental results show that this method is superior to the method proposed in literature [5]. Literature [7] proposed a word segmentation algorithm based on the double-array Trie tree dictionary structure. This algorithm is the fastest word segmentation algorithm in the literature we have seen so far. The problem with this method is that the dictionary structure is complicated and difficult to maintain. Every time a new word is inserted, the entire dictionary structure must be reconstructed again. Therefore, this limits the scope of application of the

method, and is preferably applied to a closed dictionary with low real-time requirements.[7]

The key step in the forward maximum matching algorithm is to adjust the string each time based on the length of the word in the dictionary during the segmentation process. In this paper, a new Trie tree dictionary structure is proposed. The node records the position information of the characters in the words, and the child nodes adopt the hash mechanism to design the dictionary construction algorithm. The forward maximum matching algorithm is optimized on the new dictionary structure, and the automaton mechanism is used to determine whether the longest word is formed by the position information recorded in the node. In the process of segmentation, it is no longer necessary to adjust the length of the string according to the word length, reduce the number of matches, and improve the efficiency of Chinese word segmentation. The dictionary is simple in construction, and there is no need to reconstruct the entire dictionary structure when inserting new words.

## 2 Improved dictionary structure

### 2.1 Dictionary structure

The Trie tree is a key tree represented by a tree's multiple linked list, abbreviated from the English word "Retrieval"[8]. The dictionary structure proposed in this paper improves the Trie tree. The dictionary consists of a first word hash table and a Trie tree node. In the process of constructing the Trie tree, the position information of the characters participating in the word formation is defined into different states. In this way, all the characters of all entries with a character as the first word can be organized into a finite state automaton. According to the character state, it can quickly determine whether the longest word is made. The specific judgment method is introduced in the second part of the algorithm query.

Definition: In the entire character set of all entries with a certain character as the first word, if a character is not the tail word of an entry, it is called the continuation state, and is represented by 0; If a character is the end of an entry but the word can also be used as a prefix to form a longer term, called an extended state, which is denoted by 1; A character is the last word of an entry and it cannot be used as a prefix to form a longer entry, called the final state, denoted by 2.

According to the above definition, in the improved Trie tree structure, each character in the word constitutes a node, and the corresponding logical structure is shown in Figure 1.

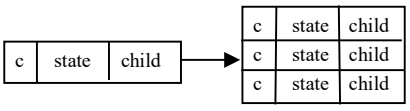


**Figure 1.** Node logic structure.

Among them, C: character; state: state; child: pointing to child node. Child nodes are all character sets prefixed with character c. The data structure of child nodes can be constructed by sequential structure, chain structure and hash structure. The data structure determines the time complexity of the lookup. Assuming that there are n characters in the child node, the time complexity of using the sequential structure binary search is log2n; With a chain structure, the time complexity is n/2; With a hash structure, the time complexity is constant. Search speed is a key indicator of the performance of the word segmentation system. The child nodes of the improved Trie tree use a hash structure.

The node formed by the first word of the entry is stored in the first word hash table, and the child node adopts a hash structure. The logic structure of the improved Trie tree is shown in Figure 2.

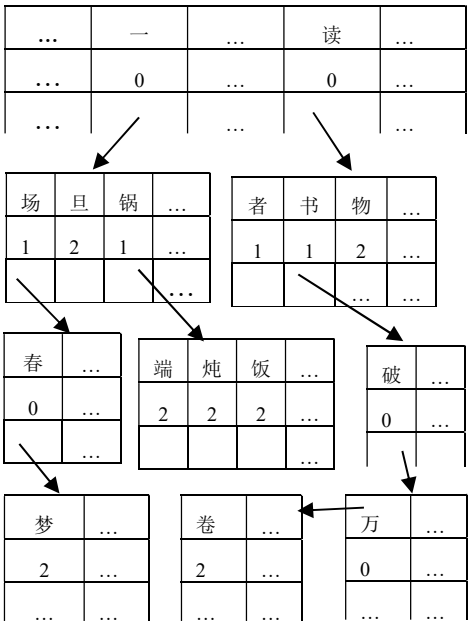


**Figure 2.** Dictionary logic structure.

For example, "一场", "一旦" and "一锅" have the same prefix "一" and store the "一" node in the first word hash table; All sets of characters prefixed with "一", such as "场", "旦", "锅", etc., are stored in a hash table to form a child node of "一".

Since the characters are represented by codes in the machine, the first word hash table in the dictionary structure can be implemented with an array of node types. According to the coding scheme of Chinese characters, the coding of Chinese characters is mapped to array index. The index access of the array is faster than the hash table. Character encoding schemes are related to specific implementation languages, such as Java using Unicode encoding.

### 2.2 Dictionary construction algorithm

Before word segmentation, you must load the thesaurus into memory to construct the Trie tree. In the construction process, insert the entries into the Trie tree. The dictionary construction process is described in algorithm 1.

Algorithm 1 dictionary construction algorithm

S1: Read the first word of the entry, insert the first word hash table as the head node of a sub tree, and continue the state;

S2: Read the next character. If there is no such character in the subtree, insert the character and continue the state. If the parent node is in the final state, modify the parent node to be in an extended state;

S3: Repeat step S2 until the last character;

S4: If the last node is a newly inserted node, the state is the final state, otherwise the state is the extended state;

S5: Repeat step S1 until all entries have been inserted.

## 3 Forward maximum matching algorithm

### 3.1 Algorithm process

In the process of string segmentation, the current character Ci is used as the first word of the entry to find the improved Trie tree dictionary structure. The characters of all the entries with the character Ci as the first word constitute a finite automaton. Each character node is in a state of the automaton, and the next character Ci+1 of the string is read as the input of the automaton. The transition is performed according to the state of the automaton. When the final state is reached or the machine is stopped due to no transition, a search is completed and a longest word is generated. The segmentation process is described in algorithm 2.

Algorithm 2 Forward maximum matching algorithm

S1: Read the character Ci from the target string S;

S2: Find Ci from the first word hash table to form the current node;

S2: Read the next character Ci+1 from S, and look for Ci+1 in the child from the current node hash; If there is no Ci+1 in the child node, stop, otherwise a new current node will be formed and character state will be read;

S3: If stop, turn step S6, if continue state, turn to step S2. If the extended state, turn to step S4, and if final state, turn to step S5;

S4: Mark extended state, turn to step S2;

S5: Extract words, turn to step S1;

S6: Determine whether the previous state is extended, if yes, go to step S7, otherwise go to step S8;

S7: Extract the words, go to step S1;

S8: The index is traced back to i+1 and go to step S1.

### 3.2 Algorithm process example

In combination with Figure 2, an example is given to illustrate the segmentation method of string "读书破坏心情". All characters in the dictionary with all the entries headed by "读" constitute an automaton. When scanning to "读", automaton state is 0, is the continuation state; Then when scanning to the "书", the automaton state is 1, is the extended state; Then when scanning to "破", the automaton state is 0, is the continuation state; Then when scanning to "坏", the automaton does not define the input from "破" to "坏", can not be transferred, stop; Return to the last extended state of "书", complete the search, split the word "读书". A new round of scanning begins with "破". This method of word-by-word search can quickly determine whether the longest word has been reached through the position information of the characters in the dictionary, thus overcoming the problem of setting the initial length of the target string according to the length of the longest word.

## 4 Algorithm evaluation and experimental comparison

### 4.1 Time complexity

The improved Trie tree dictionary structure supports verbatim hash, and the first word is mapped into an array subscript. The average time complexity of the word segmentation algorithm can be calculated according to the word frequency. The literature [1] provides word frequency statistics, as shown in Table 1.

**Table 1.** Word frequency statistics.

| Number of words | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Number of entries | 9919 | 658913 | 26352 | 21699 | 5124 | 2446 | 980 |
| Frequency | 56.75 % | 39.65 % | 2.21% | 1.19% | 0.144 % | 0.083 % | 0.023 % |

Using the time complexity calculation method provided in literature[1], the calculation process is as follows:

(1) Hash search varies according to hash conflict, and time complexity is slightly different.[8] But they are constant. The loading factor a is supposed to be 0.75. According to the calculation formula of open addressing time complexity [9], the hash time complexity is calculated once.

$$\frac{1}{\alpha}\ln\frac{1}{1-a}=1.848$$

(2) The first word uses the array structure, the time complexity is 1, and the word length of different lengths is calculated as follows.

The average search length of two words is

1+1.848 = 2.848

The average search length of three words is

1+1.848×2 = 4.696

The average search length of four words is

1+1.848×3= 6.544

The average search length of five words is

1+1.848×4= 8.392

The average search length of six words is

1+1.848×5= 10.24

The average search length of seven words is

1+1.848×6= 11.08

According to the word frequency data of Table 1, the average time complexity of segmenting a word is

0.3965×2.848+0.0221×4.696+0.0119×6.544+0.00144 × 8.392 + 0.00083×10.24 + 0.00023×11.08= 1.33

Theoretical analysis shows that the results are higher than 1.66 in the literature [5], 2.29 in the literature [4] and 2.89 in the literature [1].

### 4.2 Experimental test

Literature [7] is the fastest Chinese word segmentation algorithm we have learned. Experimental results in literature [6] show that the algorithm in this paper is superior to that in literature [1,5]. Therefore, the experiment in this paper is only compared with that in literature [6,7]. For the convenience of description, the algorithm of literature [7] is called literature 7 method, and the algorithm of literature [6] is called literature 6 method. The Java language is adopted to implement the algorithm proposed in literature [6,7] and this paper, in which the hash mechanism used by the neutron node of this paper algorithm adopts the Java language built-in HashMap class. The test corpus is used with a total of 1.05 million characters of 2M, and the test results are shown in Table 2.

**Table 2.** Experimental result.

| Algorithm name | Time overhead (MS) | Processing capacity (ten thousand characters/s) | Relative time comparison |
|---|---|---|---|
| Literature 7 method | 187 | 561 | 1 |
| Literature 6 method | 2047 | 49 | 10.95 |
| Method of this paper | 391 | 268 | 2.09 |

It can be seen that the double array Trie tree segmentation algorithm proposed in [7] is still the fastest algorithm in terms of segmentation speed. Compared with the algorithm provided by [6], this algorithm has greatly improved query speed. The literature [6] introduces the automaton mechanism, but the dictionary structure uses binary tree organization, sequential search, and the time complexity is n/2. The dictionary supports the word-by-word hash search, and the time complexity is constant level. Therefore, the segmentation speed of the algorithm is higher than that provided in the literature [6].

## 5 Conclusion

This paper proposes an improved Trie tree dictionary structure. Based on the data structure, the forward maximum matching algorithm is implemented, which solves the problem of adjusting the length of the string according to the length of the words in the dictionary in the forward maximum matching algorithm, reducing the number of matches and improving the speed of Chinese word segmentation. The dictionary is simple in construction, and inserting new words does not require changes to the entire dictionary structure, especially when the entry needs to be updated in real time. This dictionary construction uses a hash table, which requires a large space overhead. In the current hardware configuration, this is no longer a problem. The hash mechanism in the algorithm is the key factor that restricts the search speed, and there is still room for further improvement.

Fund project: Funded by the Natural Science Foundation of Hainan Province (20166225), funded by Hainan Provincial Key Research and Development Project (ZDYF2016166)
About the author: Xiong Zhibin (1973-), male, Ezhou, Hubei, master, associate professor, research direction for natural language processing

## References

1. Wu Shengyuan. A Chinese Word Segmentation Method [J]. Journal of Computer Research and Development, 1996, 33(4): 307-311.
2. Feng Guohe, Zheng Wei. Review of Chinese Automatic Word Segmentation [J]. Library and Information Service, 2011, 55(22): 41-45.
3. Sun Maosong, Zuo Zhengping, Huang Changning. An Experimental Study on Dictionary Mechanism for Chinese Word Segmentation [J]. Journal of Chinese Information Processing, 2000, 14(1): 1-6.
4. Wang Ruilei, Luan Jing, Pan Xiaohua, etc. An Improved Forward Maximum Matching Algorithm for Chinese Word Segmentation [J]. Computer Applications Software, 2011, 28(3): 195-197.
5. Chen Guilin, Wang Yongcheng, Han Kesong, etc. An Improved Fast Word Segmentation Algorithm [J]. Journal of Computer Research and Development, 2000, 37(4): 418-423.
6. Wu Jiansheng, Zhan Xuegang, Chi Chengying. An automaton-based Word Segmentation Method [J]. Computer Engineering and Applications, 2005, (8): 81-85.
7. Li Jiangbo, Zhou Qiang, Chen Zushun. Research on Fast Search Algorithm of Chinese Dictionary [J]. Journal of Chinese Information Processing, 2006, 20(5): 31-39.
8. Yan Weimin, Wu Weimin. Data Structure [M]. Beijing: Tsinghua University Press, 1997.
9. Cormen, Thomas H., et al. Introduction to Algorithms, Third Edition. The MIT Press, 2009.