An Indexing Method Based on Sentences*

Li Li [1], Chunfa Yuan[1] , K.F. Wong[2], and Wenjie Li[3]
[1]State Key Laboratory of Intelligent Technology and System
[1]Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084
Email: lili97@mails.tsinghua.edu.cn; cfyuan@tsinghua.edu.cn

[2]Dept. of System Engineering & Engineering Management, The Chinese University of Hong Kong, Hong Kong.

Email: kfwong@se.cuhk.edu.hk

[3] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong.

Email: cswjli@comp.polyu.edu.hk

Abstract

Traditional indexing methods often record physical positions for the specified words, thus fail to recognize context information. We suggest that Chinese text index should work on the layer of sentences. This paper presents an indexing method based on sentences and demonstrates how to use this method to help compute the mutual information of word pairs in a running text. It brings many conveniences to work of natural language processing.

Keywords: natural language processing, index file, mutual information

1. Introduction

Natural Language Processing often needs to analyze the relationships between words within the same sentences or the syntax of the sentences by considering the specific words. To obtain such information, sentences are usually considered as the basic processing units [4]. The fixed window approach is often used in previous studies to observe the contexts of the specific words and extract them from corpora to form a sub corpus for some purposes [5,6]. To observe the other words, corpora have to be scanned again and again. Therefore, creating an index file in advance will help locate the specified words fast and could extend the ability to cope with the large-scale problems.

Although the traditional indexing methods can locate the specific words fast, it needs extra work to provide the context information. Traditional computer indexing methods record the physical position of the words in the corpus. The position information is stored in the index file. To find out where the specified word is, the index file can provide physical position directly. Then the word in the corpus can be quickly located [3]. However, if we want to extract the sentences containing the words, the traditional processing methods have to search forward and backward to find the boundary of these sentences.

The indexing method presented in this paper creates the index file based on sentences. Unlike traditional indexing methods that record the physical position of the word in the corpus, this new method records the logical positions of the words. Not only can the index file give the numbers of the sentences in which the specified word occurs, but also locate these sentences in the corpus instantly. Since the indexing method based on sentences records the information of the contexts of the words, we are able to conveniently study some problems with the words in the sentences concerned, which could be called the logical layer. That makes it feasible to solve some natural language processing problems in a large-scale corpus.

The rest of this paper is organized as follows. The second section describes the principle of the method proposed in this paper. Then the third section summarizes its advantages. And an example applying the method is given in the fourth section. The fifth section closes this paper with conclusion.

2. Description of the method

As mentioned above, the difference between the indexing method presented in this paper and

*Supported by Natural Science Foundation of China(69975008) and 973 project (G1998030507)

the traditional ones is: the method presented here records the logical positions (sentence number), which can be mapped to physical positions (file pointer), while traditional ones only record the physical positions. By using the method presented here, when we want to get where the concerned word is, what we need to know first is not the physical positions, but the logic ones. Then we extract the sentences including the word from the corpus with the logic positions mapping to physical positions.

The indexing method presented in this paper deals with the following five kinds of files:

(1) Corpus File: a large-scale text file.
(2) Separation File: a binary file, recording the positions of the delimiter of each sentence in the corpus.
(3) Word List File: a text file, which consists of a sorted list of words.
(4) Frequency File: a binary file, which records the frequencies and the starting positions of the corresponding blocks in the Index File.
(5) Index File: a binary file, which consists of a series of blocks, the logical positions of the words in the corpus.

Corpus File and Word List File are provided by users. The other three kinds of files, Separation File, Frequency File and Index File, are created in indexing process. With the method presented here, we deal with one large-scale text file as our corpus. Thus we avoid the problem of coding the multiple documents and subdirectories. It is generally believed that Chinese information retrieval should be based on words, not characters [1,2]. So we process the corpus with segmentation.

Before creating an index file, we must have a Word List File that we want to create an index for. Generally, the word list is a sorted list. Thus we can fast locate any specified word in the list.

The Separation File is created according to the Corpus File. So the Separation File needs to be updated if the Corpus File is changed. The Frequency File and the Index File correspond to the Word List File. These three files are bound together. We may have many groups of these three kinds of files built on the same Corpus File and the corresponding Separation File. If the Word List File changes, the corresponding Frequency File and Index File will be updated as well.

The procedure of creating index files is divided into two steps, which are described respectively in the following two parts.

2.1 Create the Separation File

Five delimiters are defined as the separation punctuations of Chinese sentences: Comma (，), Period (。), Interrogation (？), Semicolon (；), and Interjection (！). We scan the corpus for these five delimiters and record the physical positions into the separation file. The Separation File is composed of a series of records; each record consists of two parts:

(1) The code of the delimiters, which distinguishes the different kinds of the delimiters;
(2) The physical position of each delimiter found in the corpus.

The following table shows the structure of the Separation File (one row represents one record):

Code of the 1st delimiter	Physical position 1
Code of the 2nd delimiter	Physical position 2
Code of the 3rd delimiter	Physical position 3
…	…

Table-1 The structure of the Separation File

From the Separation File, the sentence with the specified number from the corpus can be extracted quickly. For instance, the i_th sentence in the corpus is obviously between the physical position stored in record i-1 and the one in record i.

2.2 Create the Frequency File and Index File

From the Separation File, we can retrieve each sentence from the Corpus File in ascending order of the sentence number. Then we record the logical position of every word in the sentence, that is, the sentence number, into the index file. The index file is composed of a series of the blocks. Each word in the Word File corresponds to some consecutive blocks stored in the Index File. The number of the blocks each word associates equals the frequency of the word in the Corpus File. So we need to create the Frequency File to record the frequency of the word and to store the position of the starting block in the Index File.

Each record of the Frequency File consists of two parts:

(1) The frequency of the word occurring in the Corpus, which is equal to the number of the blocks the word associates in the Index File.

(2) The starting position in the Index File, which is the starting position of a series of corresponding blocks in the index file.

The following table shows the structure of the Frequency File (one row represents one record):

Frequency of the 1st word	Starting position 1
Frequency of the 2nd word	Starting position 2
Frequency of the 3rd word	Starting position 3
…	…

Table-2 The structure of the Frequency File

A word may appear several times in one sentence. We record the sentence number for each occurrence of the word, in the Index File. That is, the Index File will have some sequential blocks recording the identical sentence number for the word.

2.3 Search

When a user input the word, the program will search that word in the word file first and get the word number, such as No.i. Then the No.i record in the Frequency File will be obtained. The No.i record includes the information of word frequency and the starting position of its blocks in the Index File. From these blocks the logical positions (sentence numbers) are obtained and will be transformed into physical positions by Separation File. Then, we can extract all the sentences containing the word if necessary.

The following is the data-flow map, which illustrates the procedures described above.

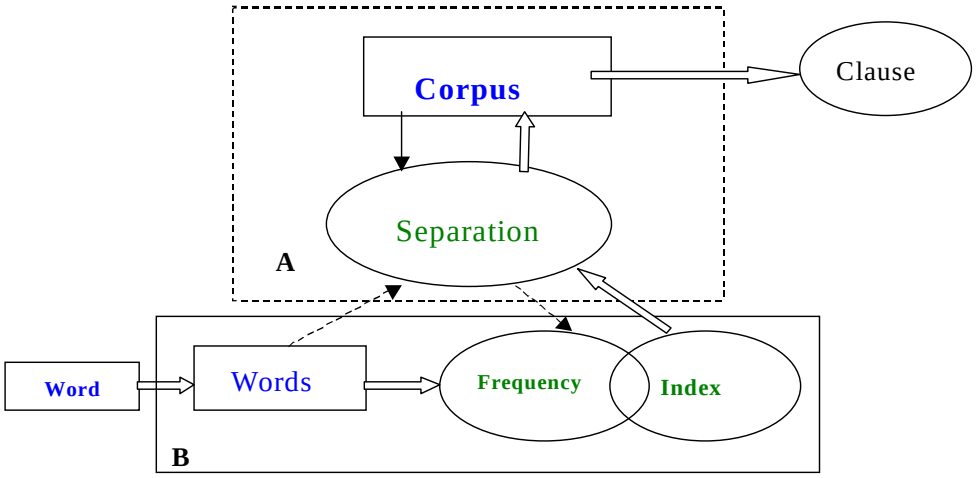

Fig-1 The data flowing map

: Create separation file
: Create index file and frequency file
: Find a word in corpus through index
A : Corpus File and Separation File bounded
B : Word File, Frequency File and Index File bounded

3. The advantages of the method

Text files include many control characters, such as carriage-return and new-line characters. So the natural language content is separated by these control characters. The meaningful separations should be some punctuations in natural languages. Our indexing method screens the effects of the control characters and brings more convenience for natural language processing than traditional ones.

The method can be applied on both raw corpora and processed corpora, quickly supplying the sentences containing keywords. Traditional indexing method can only give the physical positions of the keywords, lacking context information. It has to search forward and backward to find out sentence boundaries if needed. Actually, our method has done the portion of sentence locating work, recorded the information already in the procedure of creating Separation File and saved the time of searching.

When we study the relationship of some words in a large corpus, the method allows preprocessing on the sentences, which make viable some kinds of real-time computing in large-scale corpora. Traditional methods often use fixed-size window to observe the contexts of specified words and thus limit the ability to solve large-scale problems. The sentences, however, are the natural observing windows. The indexing method based on sentences reduces much time consumed for matching words in the corpus and concentrates on the concerned ranges directly. The next section demonstrates an example that applies the method to compute the mutual information of an adjective-noun word pair in a large-scale corpus.

4. An Example applying the method

In some natural language processing tasks, we may need to compute the mutual information of word pairs. In this example, it is assumed that the objective is to compute the mutual information of an adjective-noun pair. The adjective is "美丽" ("beautiful"), and the noun is "草原" ("grassland"). Firstly, we create the Separation File for the corpus, the Frequency File and Index File for the Word List File. Secondly, we get the sentences containing the adjective and the noun. Finally, we select the proper sentences and compute the mutual information.

4.1 Source Files

The initial sources are a corpus file and a word list file. The program runs in a personal computer with Pentium II 466 processor and 128 MB RAM. It costs one hour and two minutes to create the Separation File, three hours and fifteen minutes to create the Frequency File and Index File. Table-3 shows the size and content of these files.

FILE NAME	SIZE	CONTENT
Corpus	240,000KB	120 million tokens
Word List	385KB	62,467 word items
Separation	27,7000KB	5,816,952 sentences in Corpus
Frequency	488KB	62,467 records
Index	100,000KB	26,351,631 word occurrences in Corpus

Table-3 The source files

4.2 Search the adjective and noun pairs

When we search the adjective and the noun in the corpus, we can obtain the adjective's sentence numbers and the noun's sentence numbers from the Frequency File and Index File. By comparing the two series of sentence numbers in order and finding the common ones. We get the sentences in which the adjective and the noun both appear. In fact, we do not see these sentences now, but only get the sentence numbers in the corpus. However, we can extract these sentences from the corpus according to the separation file if necessary. If we are only concerned about the frequency that the adjective-noun pair co-occurrences and don't care about the contexts, there's no need to use the separation file and the corpus file.

We describe the algorithm of obtaining the sentences including the adjective-noun pair in the following procedure:

(1)Get the sentence numbers of the adjective: $a1 \leqslant a2 \leqslant a3 \leqslant \ldots \leqslant am$ according to the frequency file and the index file;

(2)Similarly, get the sentence numbers of the noun:b1≤b2≤b3≤…≤bn;
(3)Initialize i=1, j=1, count=0;
(4)If ai=bj, then memorize the integer i, and i++, j++, count++
else if ai<bj then i++
else j++;
(5)Repeat (4) until i=m or j=n;
(6)If observing another adjective-noun pair, repeat (1)-(5)

Actually, we've got the intersection of the adjective's sentence number set and the noun's. The sentence numbers are naturally in ascending order, since we scan the corpus sentences one by one to create the index file. This reduces the complexity of the algorithm to be O(m+n), as is shown in Step (3) – Step (5). If they are not in order, the complexity of obtaining the intersection has to be O(m*n); if they are ordered in running programs, the complexity of algorithm has to be O(m*log(m)) or O(n*log(n)).

4.3 Compute mutual information

Mutual information is widely used to measure the association strength of the two events [1,6]. The following equation is used to compute the mutual information of the adjective-noun pair:

$$MI(\text{美丽},\text{草原}) = \log_2 \frac{p(\text{美丽},\text{草原})}{p(\text{美丽})p(\text{草原})}$$

$$p(\text{美丽}) \approx \frac{\mathrm{N}(\text{美丽})}{\mathrm{N}_c}, \quad p(\text{草原}) \approx \frac{\mathrm{N}(\text{草原})}{\mathrm{N}_c}$$

$$p(\text{美丽},\text{草原}) \approx \frac{\mathrm{N}(\text{美丽},\text{草原})}{\mathrm{N}_c}$$

N_c is the total number of sentences in the corpus, so $N_c = 5{,}816{,}952$. It is observed that $\mathrm{N}(\text{美丽}) = 1884,\ \mathrm{N}(\text{草原}) = 1984$ and $\mathrm{N}'(\text{美丽},\text{草原}) = 18$ which is the number that two words appear in the same sentences. If the observing window size is assumed to be one sentence and the goal is to compute the distributional joint probability of the two words, $\mathrm{N}(\text{美丽},\text{草原}) = \mathrm{N}'(\text{美丽},\text{草原}) = 18$, then $MI(\text{美丽},\text{草原}) = 4.807976$.

If only selecting the sentences in which the adjective "美丽" modifies the noun "草原", we need to extract the 18 sentences and parse them or perform semantic analysis, then $\mathrm{N}(\text{美丽},\text{草原}) \le \mathrm{N}'(\text{美丽},\text{草原}) = 18$. Consequently, the result is $\mathrm{N}(\text{美丽},\text{草原}) = 10$, that means in the other 8 sentences the adjective doesn't modify the noun, but some other word. So $MI(\text{美丽},\text{草原}) = 3.959981$.

5. Conclusion

This paper demonstrates how the method creates the index file and gives the sentences including keywords. It then shows an example that employs the method to discover the sentences containing the adjective-noun pairs and compute their mutual information. As it is shown, the method can effectively extract the sentences including specific words and make the real-time probabilistic computation possible. It is also easy to extend the algorithm to search for three or more specific words appearing in the same sentences or to obtain the intersection, union and difference of their sentence number sets.

The method can be widely applied for many applications in Chinese information processing, such as information extraction, segmentation, tagging, parsing, semantic analysis, dictionary compilation and information retrieval. It is particularly fit for the situation of dealing with specific words and sentences in large-scale corpora and is a supporting tool for the researches of natural language processing.

References

[1] Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey and Jason Meggs, Chinese Text Retrieval Without Using a Dictionary, In SIGIR, pages 42-49, 1997.

[2] Jian-yun Nie, Martin Brisebois and Xiaobo Ren, On Chinese Text Retrieval, In SIGIR, pages 225-233, 1996.

[3] Gerard Salton and Michael J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1983.

[4] R.Rosenfeld, A Whole Sentence Maximum Entropy Language Model, In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, 1997.

[5] 孙宏林, 词语搭配在文本中的分布特征, 中文信息处理国际会议论文集, 1998.

[6] 孙茂松, 黄昌宁, 方捷, 汉语搭配定量分析初探, 中国语文, 1997 年第 1 期, 29-38 页.