Computer and Information Science
www.ccsenet.org/journal.html
Vol. 2, No. 1
February 2009

# To Construct Search Engine Analyzer for Electrical Enterprises Based on Lucene

Kehe Wu, Xia He, Tingshun Li & Hongyu Tao
Department of Computer Science & Technology
North China Electric Power University
Beijing 102206, China
E-mail:hexia1984@gmail.com

**Abstract**

There are many professional vocabularies in electrical enterprises, and existing analyzer could not fulfill the application when constructing the search engine for electrical enterprises. In this article, we take the operation system of electrical enterprises as the background, and put forward a sort of word segmentation algorithm based on the implementation of vocabulary in order to design the analyzer of search engine which could be applied in electrical enterprises. The analyzer is completed based on the electrical professional dictionary and could solve many unsatisfactory problems of existing analyzer. At the same time, we adopt the method constructing the word tree, and when loading the vocabulary, first construct a words and expressions tree in the memory, and corresponding word could be segmented only by traversing the tree when segmenting word, which could solve the limitation that one maximum word length must be enacted in usual maximum matching algorithm, and largely enhance the efficiency of word segmentation and avoid meaningless matching algorithm. Finally, we compare the analyzer with two interior analyzers in Lucene, and the result indicated that the analyzer was better than the internal analyzer in Lucene whether for time and the efficiency of word segmentation for the application system of electrical enterprise, which proved that the analyzer could fulfill the requirement to construct the search engine for electrical enterprises.

**Keywords:** Lucene, Analyzer, Vocabulary, Electrical enterprise, Search engine

## 1. Introduction

Many operation databases and large numbers of documents exist in the interior of electrical enterprise, and these resources are dispersed in various application systems and servers, and many resources have not been effectively treated and utilized. On the one hand, employees who need to require resources can not find out necessary documents. On the other hand, large numbers of resources can not be utilized, which induces low work efficiency. Therefore, it is necessary to establish search engine in the interior of electrical enterprises, and it is the key measure to enhance the level of resource utilization, and the key technology to establish the search engine is the Lucene developed by Apache.

Lucene is a telescopic information research base with high performance (Qiu, 2007). More and more application systems select Lucene to add index and search abilities for applications. It is very important to select a proper analyzer when using Lucene. There are many professional vocabularies in electrical enterprise, the interior analyzer of Lucene can not fulfill the requirement, so we need to construct a user-defined analyzer, and the component module of Lucene makes this process become very easy.

## 2. Interior analyzer of Lucene and its limitation

The interior analyzer of Lucene could deal with usual application for English word segmentation, but it has inborn shortage for Chinese information processing, and though we can find two Chinese processing analyzers, i.e. Chinese Analyzer and CJK Analyzer, in Sandbox of Lucene, but the word segmentation effect of these two analyzers is disappointed, and it is difficult to fulfill actual applications. Chinese Analyzer segments by word, and it is basically same the Standard Analyzer, and CJK Analyzer segments by two-character word, and it is arbitrary, and garbage Token (Lucene uses Token to represent the word after segmentation, and one word is one Token) will occur and influence the size of index (Otis Gospodnetic, 2007). These two analyzers are more awkward to Chinese word segmentation in special domain, so it is necessary to establish the analyzer which could be applied in special domain. In this article, aiming at characters of electrical enterprises, we construct the analyzer of search engine to fit electrical enterprises.

## 3. To construct the analyzer based on Lucene

Analysis is the process which translates the field text into the most basic index denotation unit, i.e. term in Lucene. The analyzer operation occurs in two stages, i.e. to establish index and to use Query Parser object (Otis Gospodnetic, 2007).

In Lucene, the class of analyzer is the basic class of all analyzers. It literally translates the text into word unit stream by a sort of good mode, and one example is Token Stream. The only statement to require analyzer implement the method is

public Token Stream to ken Stream (String fieldname, Reader reader).

To construct the analyzer based on Lucene, there are flowing key approaches.

(1) Segment Token from the document stream according to our own word segmentation method. It needs to further extend the basic class of Tokenizer, and the key method is next(), and every next() will return a Token.

(2) Implement the method of Token Stream. The function of Token Stream is similar with enumerator, and it will continually return Token objects, and return null when arriving at the end of text.

In this article, we adopt the word segmentation algorithm based on the vocabulary of electric enterprises to realize the analyzer of search engine which is fit for electrical enterprise, and the analyzer is current, and it could be conveniently applied in other industries only by replacing the vocabulary of the analyzer.

### *3.1 Traditional word segmentation algorithm*

People have put forward many automatic word segmentation algorithms by computer and these algorithms could be divided into two sorts, i.e. comprehensive segmentation method (knowledge word segmentation method) and mechanic matching method (or conformation word segmentation method) (Feng, 2002, P.29).

#### 3.1.1 Comprehensive segmentation method

The word segmentation system of the comprehensive segmentation method is composed by vocabulary, repository and inference machine. Lemmas are stored in the vocabulary, and formalized language rules, expression knowledge, the general knowledge and experiences when linguists implement reasoning and judgment in the process of word segmentation are stored in the repository, and the inference machine utilizes the vocabulary and repository to offer large numbers of data and knowledge, simulate linguists' logic thinking process and realize automatic word segmentation. That is an expert system of automatic word segmentation in fact. And this system has large spending, and the problem of system complexity exists except for theoretic difficulties, and it is difficult to be realized.

#### 3.1.2 Mechanic matching method

The mechanic matching method is mainly based on the principle of character string matching. It doesn't implement expression analysis and semantic analysis, and only matches and compares mechanically. Based on enough large vocabulary, it adopts certain processing strategy to match the character string in the text with the word in the vocabulary one by one, and if succeed, it will cognize the string is the word. The usual word segmentation methods include positive maximum matching method, inverse maximum matching method and the least segmentation method (Zou, 2000, P.4 & Lei, 2000, P.1270). The usual maximum matching word segmentation algorithm in mechanic matching method is easily to be implemented, but it has many obvious deficiencies.

(1) Length limitation

Because the maximum matching method must first enact an initial value of matching word length, and this length limitation is a sort of compromise between efficiency and word length for the maximum matching method. The word length is too long, the efficiency is low, and the word length is too short, the long word will be segmented wrongly.

(2) Low efficiency

Even if the word length could be enacted very shortly, but when the word length is 5 (notice: we can not shorten the word length again, because the words exceeding 5 length are too much, and we can not scarify the nicety of word segmentation), most word lengths are 2, so three times matching algorithm are wasted at least.

(3) Maximum matching is certainly not the wanted word segmentation mode

The idea of the maximum matching method is to find out the maximum matching word, but sometimes maybe we only need one part of the word except for the maximum matching word.

Based on above analysis, we put forward the improved solution project which makes the efficiency of word segmentation algorithm and the length limitation of word segmentation even the treatment of different meanings to be improved.

### *3.2 Improved Chinese word segmentation algorithm*

#### 3.2.1 To establish the vocabulary of electrical enterprises

The vocabulary structure in the application adopts simple text formatting. And every word occupies one line, and it could use # to note, and the content of notation will be ignored. Single-character word would not be embodied in the vocabulary. The structure of the vocabulary is seen in Figure 1.

In turn, n-character word could be denoted as $C_1C_2C_3\ldots C_n$, and it should not be confused with sentence. In practice, our vocabulary is stored in one text document by the UTF-8 coding. If you want to open it by the editor which is similar with notepad, please notice the problem of coding.

In the application, our vocabulary is seen in Figure 2, and it is stored in the text document which must be coding format of UTF-8.

To improve the efficiency, the length limitation of word segmentation even the treatment of different meanings for the maximum matching word segmentation algorithm, we must have a vocabulary to match the word in the text with the word in the vocabulary. It needs to rebuild the vocabulary and make the vocabulary more fit for matching and word segmentation (Chen, 2000, P.419-423).

The method in the article is to establish the tree of words and expressions after reading the vocabulary into the memory, and every node of the tree contains one word, for example, China, Chinese, Chinese Ethnic Peoples and the People's Republic of China could compose the structure of the tree. After the vocabulary is rebuilt by such way, any one sentence will be divided into single word to match with single word with tree structure, and the length of word becomes into the altitude of the tree, and every matching becomes into the traverse of tree, and the efficiency of this traverse is linear.

### 3.2.2 The idea of algorithm design

After establishing the above Chinese vocabulary, we analyze the approaches of word segmentation.

(1) Read the text which will be segmented into the buffer.

(2) First traverse the text in the tree structure, and if the matching is found, continue, and if meeting the terminal symbol, we will find that the word is a complete word, so we can regard the word as one word segmentation.

(3) Continue to do the approach (2) for the next word of the segmentation until the word is segmented.

We can see that the process of word segmentation is similar with the association function of input. Read one word, and associate until the association could not be continued. If the present segmentation could compose the word, return one Token. If the present segmentation could not compose word, remount to the nearest node which could compose the word, return. The worst situation is to return the first single word, and then begin to associate from the next word in the return result.

Here, the efficiency of character matching is almost linear. Take one word to find corresponding matching in the tree, and every matching cost is O(1), so the time complexity of matching is the length of the character string. For the character string which length is n, its complexity of word segmentation is O(n). The average complexity of maximum matching is $O(n^2)$. And we didn't consider the inclusion of various meanings and the branch disposal, and the complexity is still limited even if adding these instances. The core codes of word segmentation algorithm includes

```
public final Token next() throws java.io.IOException {
  length = 0;
  start = offset;
//construct association stream and simulate the association function of input
  AssociateStream assoStream = new AssociateStream(WordTreeFactory.getInstance());
//process of association word segmentation
  if (assoStream.associate(c)) {
          push(c);
   if (!assoStream.canContinue()) {
      assoStream.reset();
      return flush();}}
 else {
//when present node could compose one word, return a Token
 if (assoStream.isWordEnd()) {
     assoStream.reset();
bufferIndex--;
offset--;
return flush();}
else if (assoStream.isOccurWord()) {
//if the word exists in the association stream, remount to the former node which could compose the word, and return token
assoStream.backToLastWordEnd();
bufferIndex=bufferIndex-(length-
    assoStream.getSetp()) - 1;
offset = offset - (length-
    assoStream.getSetp()) - 1;
length=assoStream.getSetp();
assoStream.reset();
return flush(); }
//if the word doesn't exist in the association stream, output the single word as a Token
else {if (length > 0) {
bufferIndex = bufferIndex - (length - 1) - 1;
offset = offset - (length - 1) - 1;
length = 1;
assoStream.reset();
return flush();}
assoStream.reset();
push(c);
return flush();}
}
}
```

### *3.3 Problems should be concerned when segmenting word*

When using the method based on the vocabulary, we must face one problem, i.e. to read the vocabulary into the memory, and it usually wastes long time, but fortunately we only need do the work once, and when we load the vocabulary into the memory, all works will be implemented in the memory, and the speed of word segmentation will be enhanced largely. When the opportunity that we preload the vocabulary is the time that we first implement word segmentation, it is same to lazy load, and we will initialize it only when we use it.

### *3.4 Experimental result of Chinese word segmentation*

In the experiment, we select the article with about two thousand words (essay in electrical industry), and disposal it by three sorts of analyzers. And the experimental results are seen in Table 1.

Standard Analyzer and CJK Analyzer didn't check the vocabulary, so they didn't segment the word accruing to any semantics. Standard Analyzer only simply divided the Chinese letter in the article into single word, but CJK Analyzer segmented the text to two-character word according to the two-character segmentation method, and many garbage words and expressions occurred. The experiment result is very amazing, and after preloading the vocabulary, the word segmentation speed of the analyzer in this article actually far exceeds the algorithm of Standard Analyzer which needs not checking any vocabulary.

## 4. Conclusions

Aiming at the applied particularity of electrical enterprise, after analyzing the key technology of Lucene analyzer, we put forward the analyzer implementation based on the vocabulary, realize the analyzer by the interface offered by Lucene and acquire better effect of word segmentation.

## References

Chen, Guilin & Wang, Yongcheng. (2000). A Sort of Improved Rapid Word Segmentation Algorithm. *Journal of Computer Research and Development.* No.37. P.419-423.

Feng, Shuxiao, Xuxin & Yang, Chunmei. (2002). The Progress of Domestic Study for Chinese Participle Technology. *Journal of Information.* No.11. P.29.

Han, Kesong, Wang, Yongcheng & Chen, Guilin. (1999). The Word Segmentation Module System without Dictionary for Language. *Application Research of Computers.* No.10. P.8-9.

Leiming, Liu, Jianguo & Wang, Jianyong. (2000). A Sort of Search Engine System Dynamic Renewal Module Based on Dictionary. *Journal of Computer Research and Development.* No.10. P.1270.

Li, Jianhua & Wang, Xiaolong. (2000). An Effective Method on Automatic Identification of Chinese Name. *Chinese High Technology Letters.* No.2.

Li, Qinghu, Chen, Yujian & Sun, Jiaguang. (2003). A New Dictionary Mechanism for Chinese Word Segmentation. *Journal of Chinese Information Processing.* No.17. P.13-18.

Otis Gospodnetic & Erik Hatcher, interpreted by Tanhong, Li, Junhong & Gao, Chengshan. (2007). *Lucene in Action.* Beijing: Electronic Industry Press.

Ou, Zhenmeng & Yu, Shunzheng. (2000). Research of Chinese Word Automatic Segmentation Used in Search Engine. *Computer Engineering and Applications.* No.8. P.80-84.

Qiuzhe & Fu, Taotao. (2007). *Developing Our Own Search Engine: Lucene2.0+Heritrix.* Beijing: People's Posts & Telecommunications Publishing House.

Yan, Weimin & Wu, Weimin. (1992). *Data Structure.* Beijing: Tsinghua University Press.

Zou, Haishan, Wuyong, Wu, Yuezhu & Chenzhen. (2000). Chinese Information Processing Technology in Chinese Search Engine. *Application Research of Computers.* No.12. P.4.

Table 1. Result of experiment

| Analyzer | Word segmentation algorithm | Time consumption |
|---|---|---|
| Standard Analyzer | Segmenting by word | 67ms |
| CJK Analyzer | Segmenting by two-character word | 7ms |
| Chinese Analyzer (it is implemented in the article) | Improved word segmentation algorithm | Without vocabulary preload: 1247ms<br>Vocabulary preload: 44ms |

#two-character word

$C_1$C2

# three-character word

C3C4C5

#four-character word

C6C7C8C9

Figure 1. Vocabulary Structure

Electric power

Quality of electricity

Power failures

Generate Electricity

Power generating enterprises

Generated energy

Set

Figure 2. Vocabulary Structure in this Application