International Journal of Hybrid Information Technology
Vol. 10, No. 8 (2017), pp. 53-66
http//dx.doi.org/10.14257/ijhit.2017.10.8.04

Research on Application of Lucene Search Engine in Social Network Platform

Mei Yu[1], Wentao Xing[2], Jian Yu[3*], Jie Gao[4], Shengguang Ma[5] and Tenghai Wang[6]

[1 2 3 4 6]School of Computer Science and Technology, Tianjin University, Tianjin, P.R. China
[5]Production and operation Department, Karamay Petrochemical Company, PetroChina
{[1]yumei;[2]Winterto1990; [3]yujian; [4]gaojie; [6]wangth}@tju.edu.cn
[5]mshgksh@petrochina.com.cn

Abstract

With the development of Web2.0 technology, social networks begin to play an increasingly important role in people's life. Widely used social network for researchers has brought some potentially useful information, such as the user's interests and preferences. At the same time, constant search results provided by the search engine can not meet the individual needs of users, a new way search engine, personalized search is an urgent need to explore. Based on this demand, the paper from Sina microblog *mining user interests, and the use of open-source Lucene search engine completes personalized search.*

In this paper, the structure and principle of Lucene search engine are summarized and some related knowledge are introduced, such as text preprocessing and vector space model. Then, this article proposes the Lucene TagMatch Ranking (LTR) algorithm. The main idea is using user's Sina microblog *texts to extract tags of interest and measure the matching degree between web pages and user's interests which named value of tag matching degree by the vector space model, then combine the traditional Lucene scoring mechanism, finally realize personalized ranking results based on the user's interest. At last the Eclipse programming algorithm based on Java is used to carry out comparison experiment to confirm the effectiveness of the algorithm. The results are presented to the user in the ranking which based on the user's interests, it will be able to reach recommendation algorithm based on user's interests.*

Keywords*: social network; Lucene; interest tags; LTR*

1. Introduction

The search engine can help users to quickly get the information needed by users from numerous network information. But the traditional search engine provides only a simple search service, for the search term returned to the unified of no difference in the results, the users need to examine the search results. In many cases, this is a time-consuming and laborious work. Therefore, the traditional search engine can't meet the individual needs of different users.

Actually, most of the search engines are short [1,5] and inconclusive[2,3], and different users may have completely different information needs and objectives under the same conditions[1,4,9].

Microblog texts as the object of the study, can accurately reflect the user's attentions and interest. The paper combines the Lucene search engine with the social network platform, improves the scoring mechanism of Lucene. It is mainly based on the user's microblog texts to mining out of interest tags and create a search engine that can

ISSN: 1738-9968 IJHIT
Copyright © 2017 SERSC

understand the user’s interest. The main focus of the paper is the personalized sorting algorithm. In order to realize the personalized search of Lucene, the LTR algorithm is designed to replace the original scoring mechanism of Lucene.

2. Related Works

Personalized search is considered an effective solution to this problem since different search results based on preferences of users are provided. Various personalization strategies including [6, 7, 10, 11, 12, 13, 14] have been proposed and personalized web search systems have been developed, but they are far from optimal.

Leung [15] introduces a set of personalized search engine in the mobile terminal. The overall architecture of the study includes client, server and back-end search engines. Client is mainly responsible for the collection of user's browsing history and concept extraction tags, creating a user profile user profile stored in the local. Back end search engine is a common traditional search engine without personalized search. The server to connect the client and back-end search engines, is received from a client search request and the user profile and the back-end search engine returns the search results according to a user profile for reordering and returns it to the client. This personalized search advantage is that the user does not provide any information to the search engines, and effectively protect user’s privacy. But the drawback is that if the user's browsing history relatively lax or shared a client, it will cause the deviation of the judgment of user preferences, lead to inaccurate search results. Nathaneal Ramesh [16] method is introduced to search an analysis of user’s interest based on user behavior on Facebook personalization, and proposes a Lingo clustering algorithm for the new information. The benefits of this approach are more accurate for the user interest analysis, but the drawback is that the algorithm's time complexity is high which affects the efficiency of search engine. Xu Z [17] introduces in the paper is a kind of personalized search method on the social network. A major feature of social networks is that users can share pages and comment with others on Web pages. Based on this characteristic, the author presents a new algorithm web-based overall evaluation of the user's overall assessment of the similarity between the user - Double personalized rankings (D-PR). The advantage of this algorithm not only considers the user's interests, but also the similar user comments of interest, improves the accuracy of personalized. But the disadvantage is less unfair for comment. Liu J [18], a scheme was put forward, based on the user's click behavior on the web page to make personalized news recommendation. A scheme [19], a scheme was proposed which is based on user interest to solve personalized recommendation. The [20] solves the problem of personalized recommendation by the method of clustering.

A related work, Teevan [21] proposed, the user through the use of less accurate, but easy to write, query and determine the use of the web, users can often achieve their search targets.

The topic mainly studies the application of Lucene search engine in the social network platform. Buildthe Lucene search engine as the platform, analysis on the recent microblog content of users, then join the label agreement factor concept which based on Sina microblog. Then put forward a suitable weight calculation formula for personalized ranking, which will fit the user's interest tendency more.

3. TagMatch Ranking Algorithm in Lucene

3.1. Scoring Mechanism of Lucene

Lucene has a complete set of scoring mechanism. It evaluates and calculates the score of each page in real-time. The score of a page will vary according to the different user input keywords. The more close to user requirements, page score will be higher. The scoring mechanism of the Lucene shows the frequency of the keyword in each page, as shown in the Equation (1).

$$\sum_{\mathrm{t\ in\ q}} tf(t\ in\ d) \cdot idf(t) \cdot boost(t.field\ in\ d) \cdot lengthNorm(t.field\ in\ d) \tag{1}$$

where t is the key field after user query split, tf(t in d) is on behalf of the document d in the search term t frequency, idf(t) represents the frequency of the search term t in the inverted document, boost (t.feld in d) is on behalf of the weighted factor of domain, the value of the index is set in the process of indexed. The lengthNorm(t.field in d) is on behalf of the field value of standardization, indicating that how much entries stored in the field show that this value is calculated in the indexing process, and is also stored in the index.

3.2. Text to Quantization

Vector space model (VSM) is a classic text quantization means. This theory was first proposed by Salton in 1980s [8], and now has become one of the necessary means of text analysis. A good understanding of the concept of VSM is words appear in the text as feature vector.Thus,it is converient to calculate the study in the later.

Description of vector space:

Let D be a set of documents containing N target documents, D={D_1, D_2, ...D_n},which , D_i is a random target document in D. Feature items of all the document set D. For D ,the vector can be expressed as D_i={d_{i1}, d_{i2},... d_{im} },in which ,d_{ij} is expressed as the weight of any text feature item T_j.

For the weight of d_{ij} selection will direction affect the results of the text analysis.The TF-IDF (Term Frequency - Inverse Document Frequency) is often used for text vector in the statistical weight calculation method. TF express as the word frequency, the IDF expresses the ratio that the files which contain the key words in all of the documents, IDF value is general importance of a statistical feature term (keyword) in the entire document set. The above two weights can be calculated by the formula (2) and the formula (2).

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{2}$$

In Equation (2), n_{kj} represents the number of times that the feature item appears in the document Dj.

$$IDF_i = \log \frac{|D|}{|\{j : T_i \in D_j\}|} \tag{3}$$

However, there are shortcomings in the TF-IDF. If a particular feature item in the document set D with very high frequency, it has a high representative for the entire corpu which should be given higher importance of quantifying the value. However, TF-IDF method weakens the weight. In addition, TF-IDF does not consider the feature item, resulting in the analysis result deviation.

3.3. Generation of User Interest Tags

There are four relevant definitions here.

1.*TagMatch(A,U)*, a value which represents the matching degree between the page A and the user U's interest tag. This value is higher, show that this page is more in line with the user's interest in the U.

2. $IncTag_{A,U}$, regard as a row vector which contains N elements (the N regards as the total number of label element vector of user U). It records page A contains the interest tabs of user U. Each element of this vector is 0 or 1, 0 indicates that it does not contain the corresponding label, and the 1 is the opposite.

3. $TagWeight_u$, regard as a column vector which contains N elements, records the weight of U's each label. Each element of this vector is a number between 0 and 1.

In order to realize the personalized search, the most important thing is to understand the users' interest, which is based on the users' interest to predict the users' most needed information. It makes sense to understand the user's interest though social networks and most users of social networks (this study mainly takes Sina microblog as an example) express their interests or hobbies simply. For example, a user tweeted: "Juventus will win!". So it's easy to know that this is a Juventus fan. But, how to let the search engine to understand the user's interest is not a simple problem. First, the search engine can't understand natural language, and it can't read the user's interest from the user's blog directly. Secondly, search engine can't judge whether a web page meets users interest or not intuitively. Therefore, we need a measure standard to help search engine ranking.

In order to solve the first problem, we need to get users' microblog content for text processing as to obtain the labels representing users' interests. The algorithm [9] this thesis used is called WUK (Weibo User Keyword Algorithm), which generates labels of users' interests.

The WUK algorithm mainly contains two serial modules, which are Chinese word segmentation module and feature extraction module.

3.3.1. Chinese Word Segmentation Module

As mentioned in the second chapter, word segmentation is an essential step for Chinese text information mining. There are many comprehensive Chinese word segmentation software at present. However, due to the following characteristics of the microblog text:

(1) Microblog text has the characteristic of short passage. Although it requires 140 limited words, each microblog users have hundreds and thousands of text data need to deal with. Hence, the vector space model formed by text data has the characteristic of high dimension and sparse.

(2) The complex data structure of each microblog text. Every microblog text needs to store some specific information of users, including the time of microblog releasing, the client information users used to release microblog, the comment and forward of microblog, and the users list associated with microblog.

(3) Microblog text has many special symbols, including "#...#", "@" and emoticon icons.

(4) Microblog text has some special words, which different from the words used in the general corpus, such as Web links, video and so on.

Therefore, most of word segmentation software are not very good to complete the work of word segmentation. According to the four characteristics mentioned, the algorithm of WUK has improved respectively. According to the characteristic of 1, the improved method will be shown in the following feature extraction. According to the characteristic of 2, some useful information is extracted and the content of forwarded text is associated; According to the characteristic of 3, as the thesis mainly focus on users' features based on microblog text and the relationship between users won't be further discussed, so the users' information behind the "@" symbol is ignored. The theme words in the "# #" symbol are the keywords of users. According to the characteristic of 4, WUK adds new stop words through statistical method and filter URL format data in microblog.

3.3.2. Feature Extraction Module

Traditional weight calculation TF-IDF method itself has two obvious shortcomings, these will have some certain influence on the accuracy and authority of the results of feature words extraction. The core reason is that the combination of TF weights and IDF weights leads to the measuring bias of comprehensive weights. Combined with microblog text doesn't has the characteristic of randomness like the common text set, the WUK algorithm proceeding feature extraction only based on TF.

Table 1 is pseudocode of the WUK algorithm.

Table 1. Pseudocode of the WUK algorithm

Pseudo code of the WUK algorithm

```
    Step 0.    Input: Microblog text
    data Step 1.       for i←1 to
    user.microblogSum

User I microblog content to the
user.content if User I have forwarded
microblog records Add forwarding
content to user.content
en
d
if
i←
i+
1
en
d
for

    Step 2.    remove the URL format
  data in user.content

    Step 3.    remove the expression data in
    user.content Step 4.       removal of special
    symbols "@"

    Step 5.    for all string=="#...#"
```

```
Join in the keyword
list Delete from
User.content
end for

    Step 6.    Split user.content into
    word[] Step 7.    Put word[] into
    keyword list
    Step 8.    Lead into disable thesaurus of stoplist,
  add the stop words of microblogging features to
  generate mystoplist

    Step 9.    for each word in
keyword if (word in mystoplist)

delete the
word end if

end for

    Step 10.   Keyword table for the TF sort to
  generate word list

    Step 11.   Output: TF word cloud results
```

When users' features are extracted by the WUK algorithm, each word is a tag of a user's interest. In order to measure the status of each label, TF value of every word divided by TF value of all feature words, then sum up the result will reach the weight of each label, which is also the definition of vector element value (the value of $TagWeight_U$).

3.4. Calculation of Label Matching Degree

As mentioned earlier, to achieve the personality search, search engines also need a standard to measure content of page and the matching degree of users' interest tags.

Obviously, each user may has many interest tags and each page may also contains many matching content, but it doesn't mean that the more interest tags a page has, the more users' interests increased. Because the position of each tag in users' mind is unequal, that's also the reason why use TF as a feature extraction in the last section. Generally speaking, the more often an interest label appears in one's microblog, the higher position each label in his mind, that is, the more interesting of the words he has. Correspondingly, the label of the TF value is higher.

In the third definition, $IncTag_{A,U}$ page A contains user U's tags of interest, such as page A contains the first label of user U, then the first element of the vector $IncTag_{A,U}$ is 1, but if page A doesn't contain user U's second label, then the second element of the vector $IncTag_{A,U}$ is 0.

After that, we can get the following formula to calculate the matching degree between the page and the user's interest, as shown in the Equation (4):

$$TagMatch(A,U) = IncTag_{A,U} \cdot TagWeight_U$$

(
4)

As the first definition, *TagMatch(A,U)* were used to characterize the matching degree of interest tag between page A and user U, the higher values of *TagMatch(A,U)*, the more interest in page A user U has.

In summary, combined with the scoring mechanism of Lucene itself mentioned in the

$$T - Rank = \alpha \cdot S(q,d) + \beta \cdot TagMatch(A,U) \quad (5)$$

Where the α and β are between 0 to 1 of the coefficient, and the α+ β=1.
The algorithm is named as LTR (Lucene TagMatch Ranking) Algorithm.

To achieve the Equation (5) scoring methods, there are several steps. Firstly, create a domain score. Secondly, create a custom query object according to the score domain and the original query. Thirdly, rewrite the score method. Lastly, register using rewriting.

4. Experiment and Result Analysis

4.1. Data Sources and Data Preprocessing

Sina official API is a convenient way to get microblog data. API is an application programming interface. Opening API is equal to opening application programming interface. Opening API allows other users to access and invoke their data contents via an open interface. The opening API of Sina microblog is an open platform which is based on information subscription, share and exchange function of Sina microblog system. Opening microblog platform for researchers and application developers provide massive microblog user information, relationships, tweets and the instant occurrence fission information dissemination channels.

By adding Sina microblog Java SDK into project, realize and get the certification of official open platform power OAath2.0. It can acquire microblog text datum which satisfys the experiment demand. The open platform of Sina microblog provides different SDK interfaces which are convenient for researchers to access the data and carry out the research smoothly.

The data set taken by this thesis is NLPIR microblog content corpus, which is by the Beijing Science and mining and safety laboratory Zhang, PhD, University of technology network search, obtain from Sina microblog by public collection and extraction, in order to promote micro computing research, through natural language processing and information retrieval sharing platform www.nlpir.org be publicly shared among the 23 million pieces of data.

The WUK keyword generation algorithm adapted to the microblog text has been introduced before. Algorithm operation including special symbol removal and standardization can achieve the above purpose by using the RCurl, XML, rJava etc. Toolkit to deal with the microblog data and redact code in R studio platform environment for the optimization of micro blog culture to stop words list, the number, time removal and the URL removal. The first half of the WUK algorithm is to preprocess the data. Table 2 describes the data preprocessing R language code prior to WUK algorithm.

Table 2. Pretreatment of Microblog Data -R Script

WUK algorithm pretreatment-R Script

```
 words<-removePunctuation(words)
words<-removeNumbers(words)
 words<-removeWords(words,
c(emotions))

 ...

 microblogCorpus<-tm_map(microblogCorpus, removeURL)
```

After the above pretreatment process, we can use the word segmentation module WUK to split the content to words. Feature extraction can be carried out after word segmentation.

In the experimental process, the inputting datum in the process of text quantization includes corpora which generated by using Chinese data sets. (i.e., before using the vector space elements required dimension reduction operation to avoid too high dimension), and target microblog text after a specific pretreatment.

Vector space model is a classical model for text mining, it can be implemented directly by quoting the open source SDK package. The realization codes are shown as Table 3.

Table 3. Examples for VSM Code Achieving

VSM R Core-Script

```
   microblogCorpus<-Corpus(VectorSource(res))#form a
   corpus microblogData<-as.data.frame(microblogCorpus)
   microblogData<-t(microblogData)
   microblogData<-as.data.frame(microblogData)
   microblogCorpusForAnys<-
   Corpus(DataframeSource(microblogData))

   microblogTDMatrix<-
TermDocumentMatrix(microblogCorpusForAnys, control =
list(wordLengths = c(1, Inf))
```

Using TF values as weights to calculate extracted features after quantifying keywords which represent a user’s interests can be selected.

This experiment selects five microblog users to research randomly and selects the highest top 10 TF values of their keywords as their tags of interest. Results are shown in Table 4.

Table 4. User's Interest Tags and Weights

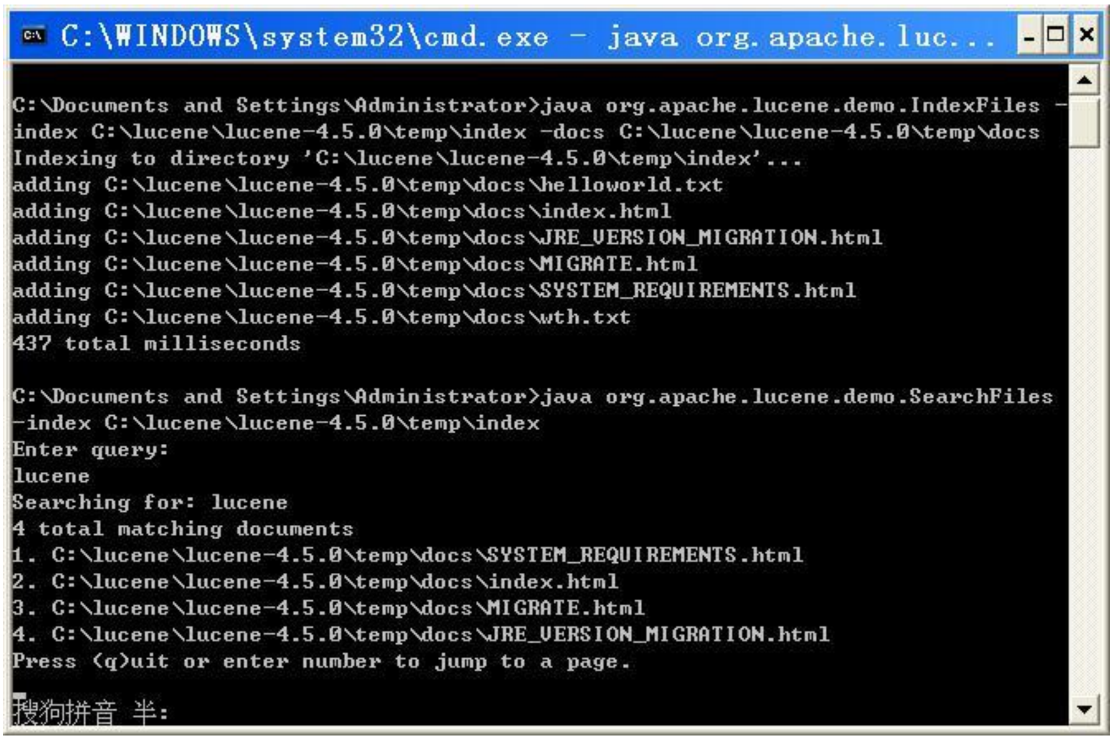
	User 1	User 2	User 3	User 4	User 5
Label 1	婚纱	CCTV	篮球	美容	星座
Weight 1	0.2523	0.2168	0.2164	0.1934	0.2962
Label 2	照片	新闻	音乐	教学	狮子座
Weight 2	0.1456	0.1855	0.2083	0.1876	0.1759

Copyright ⓒ 2017 SERSC

Label 3	爱	北京	火箭	彩妆	爱情
Weight 3	0.1136	0.1343	0.1577	0.1535	0.1281
Label 4	幸福	政府	比赛	美	金牛
Weight 4	0.1017	0.1091	0.1091	0.1292	0.1053
Label 5	女人	调查	周杰伦	化妆品	运势
Weight 5	0.0954	0.0916	0.1043	0.1045	0.0874
Label 6	习作	专家	好听	转发	发现
Weight 6	0.0759	0.0829	0.0569	0.0779	0.0657
Label 7	色调	财经	深夜	希望	总结
Weight 7	0.0692	0.0612	0.0424	0.0632	0.0598
Label 8	修片	公众	专辑	脸	娱乐
Weight 8	0.0605	0.0497	0.0376	0.0485	0.0436
Label 9	留念	记者	国战	更新	同道
Weight 9	0.0428	0.0382	0.0351	0.0261	0.0225
Label 10	美食	小康	征途	感动	流行
Weight 10	0.0394	0.0307	0.0322	0.0161	0.0155

4.2 Personalized Search of Lucene

Download and install JDK and Lucene kit. The experimental versions are jdk1.6.0_45 and lucene-4.5.0. After configuring the environment variable, you can run the Lucene, as shown in Figure 1.



```
C:\Documents and Settings\Administrator>java org.apache.lucene.demo.IndexFiles -
index C:\lucene\lucene-4.5.0\temp\index -docs C:\lucene\lucene-4.5.0\temp\docs
Indexing to directory 'C:\lucene\lucene-4.5.0\temp\index'...
adding C:\lucene\lucene-4.5.0\temp\docs\helloworld.txt
adding C:\lucene\lucene-4.5.0\temp\docs\index.html
adding C:\lucene\lucene-4.5.0\temp\docs\JRE_VERSION_MIGRATION.html
adding C:\lucene\lucene-4.5.0\temp\docs\MIGRATE.html
adding C:\lucene\lucene-4.5.0\temp\docs\SYSTEM_REQUIREMENTS.html
adding C:\lucene\lucene-4.5.0\temp\docs\wth.txt
437 total milliseconds

C:\Documents and Settings\Administrator>java org.apache.lucene.demo.SearchFiles
-index C:\lucene\lucene-4.5.0\temp\index
Enter query:
lucene
Searching for: lucene
4 total matching documents
1. C:\lucene\lucene-4.5.0\temp\docs\SYSTEM_REQUIREMENTS.html
2. C:\lucene\lucene-4.5.0\temp\docs\index.html
3. C:\lucene\lucene-4.5.0\temp\docs\MIGRATE.html
4. C:\lucene\lucene-4.5.0\temp\docs\JRE_VERSION_MIGRATION.html
Press (q)uit or enter number to jump to a page.

搜狗拼音 半:
```

Figure 1. Running Example Diagram

In Figure 1, the first command:

java org.apache.lucene.demo.IndexFiles -index C:\lucene \lucene-4.5.0 \temp \index -docs C:\lucene\lucene-4.5.0\temp\docs

Copyright ⓒ 2017 SERSC

Is creates an index for the C:\lucene\lucene-4.5.0\temp\docs directory under the file, and the index is stored in the directory C:\lucene\lucene-4.5.0\temp\index. The file here can be a web page, TXT, PDF, word, and other different types of files. In this experiment files are downloaded from the web page.

Second commands:

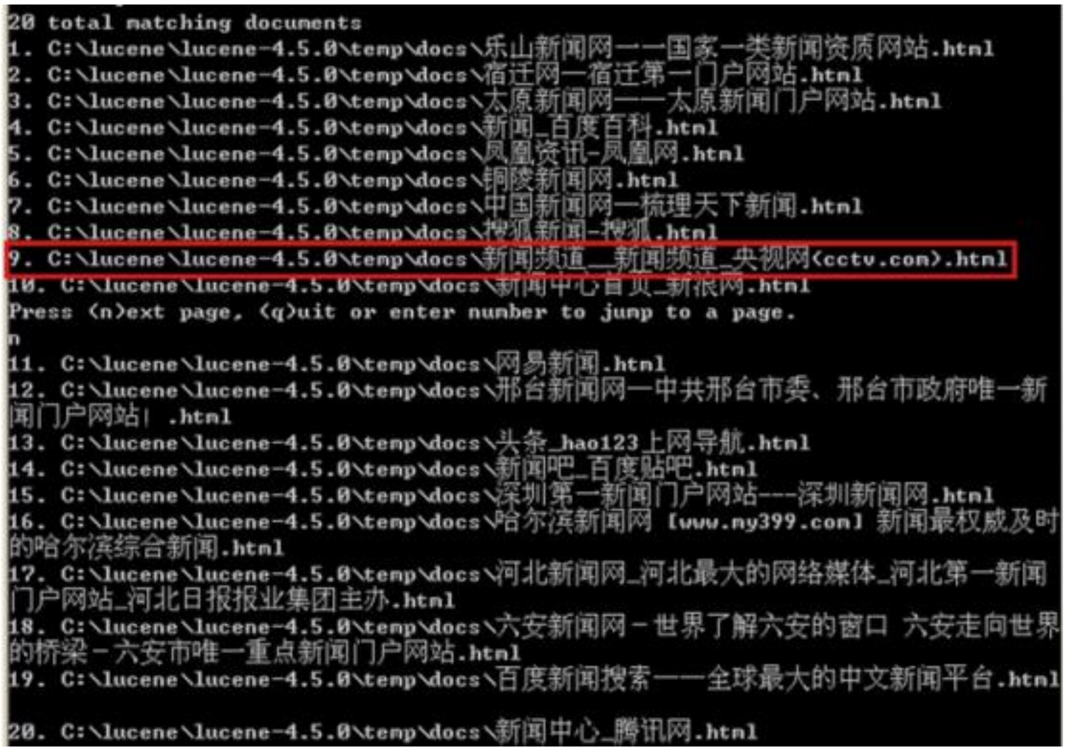
java org.apache.lucene.demo.SearchFiles –index C:\lucene \lucene-4.5.0 \temp
\index It is an index based on the index directory to execute the query.

Create jar in the eclipse project and introducing Lucene toolkit package, then achieve personalized sorting algorithm of Lucene.

In order to make the scientific and objective, for each search, both prior to in Baidu search engine on the corresponding keyword search and download before 20 pages as a Lucene engine of the original data.

Take the user 2 in Table 4 as an example. This is obviously a concern CCTV news and information which users like, so when the users use the "news" as a keyword search and hope to get the results. The result is the CCTV news website obviously.

Figure 2 shows the results of the Lucene search for the "news" in the original scoring mechanism.



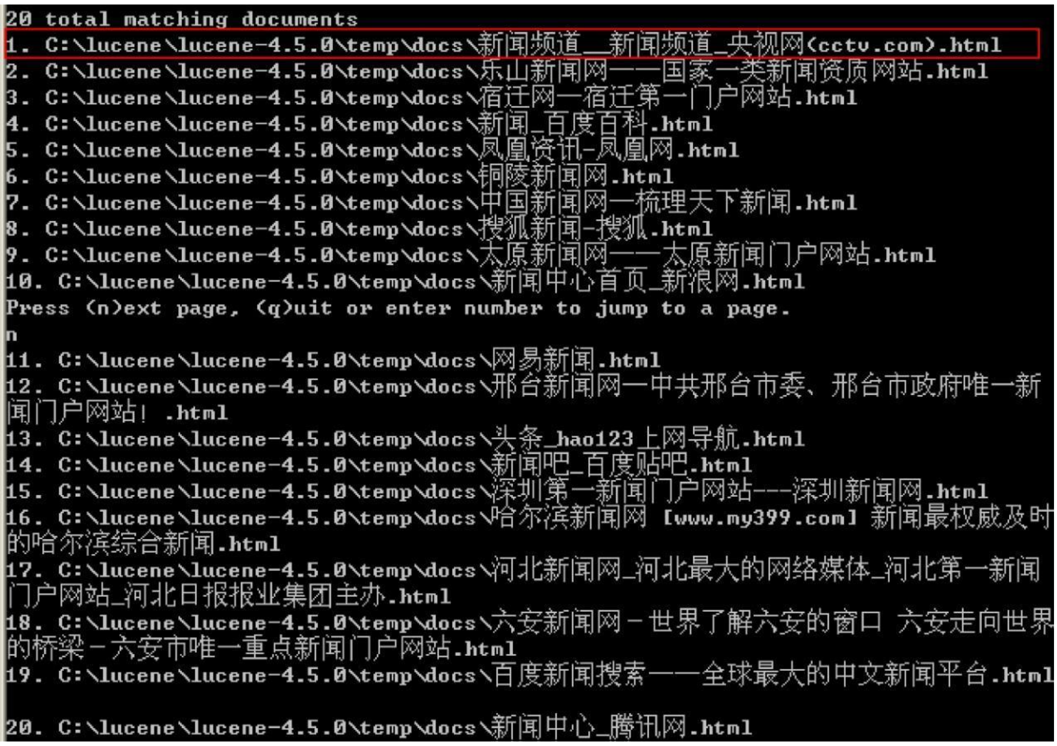
```
20 total matching documents
1. C:\lucene\lucene-4.5.0\temp\docs\乐山新闻网——国家一类新闻资质网站.html
2. C:\lucene\lucene-4.5.0\temp\docs\宿迁网—宿迁第一门户网站.html
3. C:\lucene\lucene-4.5.0\temp\docs\太原新闻网——太原新闻门户网站.html
4. C:\lucene\lucene-4.5.0\temp\docs\新闻_百度百科.html
5. C:\lucene\lucene-4.5.0\temp\docs\凤凰资讯-凤凰网.html
6. C:\lucene\lucene-4.5.0\temp\docs\铜陵新闻网.html
7. C:\lucene\lucene-4.5.0\temp\docs\中国新闻网—梳理天下新闻.html
8. C:\lucene\lucene-4.5.0\temp\docs\搜狐新闻-搜狐.html
9. C:\lucene\lucene-4.5.0\temp\docs\新闻频道__新闻频道_央视网<cctv.com>.html
10. C:\lucene\lucene-4.5.0\temp\docs\新闻中心首页_新浪网.html
Press <n>ext page, <q>uit or enter number to jump to a page.
n
11. C:\lucene\lucene-4.5.0\temp\docs\网易新闻.html
12. C:\lucene\lucene-4.5.0\temp\docs\邢台新闻网—中共邢台市委、邢台市政府唯一新
闻门户网站| .html
13. C:\lucene\lucene-4.5.0\temp\docs\头条_hao123上网导航.html
14. C:\lucene\lucene-4.5.0\temp\docs\新闻吧_百度贴吧.html
15. C:\lucene\lucene-4.5.0\temp\docs\深圳第一新闻门户网站---深圳新闻网.html
16. C:\lucene\lucene-4.5.0\temp\docs\哈尔滨新闻网 [www.my399.com] 新闻最权威及时
的哈尔滨综合新闻.html
17. C:\lucene\lucene-4.5.0\temp\docs\河北新闻网_河北最大的网络媒体_河北第一新闻
门户网站_河北日报报业集团主办.html
18. C:\lucene\lucene-4.5.0\temp\docs\六安新闻网－世界了解六安的窗口 六安走向世界
的桥梁－六安市唯一重点新闻门户网站.html
19. C:\lucene\lucene-4.5.0\temp\docs\百度新闻搜索——全球最大的中文新闻平台.html

20. C:\lucene\lucene-4.5.0\temp\docs\新闻中心_腾讯网.html
```

Figure 2. Lucene Results of the Original Scoring Mechanism

Figure 3 shows the Lucene algorithm to improve the LTR (coefficient of alpha 0.6) under the search for "news" results:

From the comparison we can see that the CCTV news network ranking from ninth directly to the first after the improvement. Obviously, the results of the LTR algorithm will make users more satisfied than the original Lucene scoring results.



```
20 total matching documents
1. C:\lucene\lucene-4.5.0\temp\docs\新闻频道__新闻频道_央视网(cctv.com).html
2. C:\lucene\lucene-4.5.0\temp\docs\乐山新闻网——国家一类新闻资质网站.html
3. C:\lucene\lucene-4.5.0\temp\docs\宿迁网—宿迁第一门户网站.html
4. C:\lucene\lucene-4.5.0\temp\docs\新闻_百度百科.html
5. C:\lucene\lucene-4.5.0\temp\docs\凤凰资讯-凤凰网.html
6. C:\lucene\lucene-4.5.0\temp\docs\铜陵新闻网.html
7. C:\lucene\lucene-4.5.0\temp\docs\中国新闻网—梳理天下新闻.html
8. C:\lucene\lucene-4.5.0\temp\docs\搜狐新闻-搜狐.html
9. C:\lucene\lucene-4.5.0\temp\docs\太原新闻网——太原新闻门户网站.html
10. C:\lucene\lucene-4.5.0\temp\docs\新闻中心首页_新浪网.html
Press (n)ext page, (q)uit or enter number to jump to a page.
n
11. C:\lucene\lucene-4.5.0\temp\docs\网易新闻.html
12. C:\lucene\lucene-4.5.0\temp\docs\邢台新闻网—中共邢台市委、邢台市政府唯一新
闻门户网站! .html
13. C:\lucene\lucene-4.5.0\temp\docs\头条_hao123上网导航.html
14. C:\lucene\lucene-4.5.0\temp\docs\新闻吧_百度贴吧.html
15. C:\lucene\lucene-4.5.0\temp\docs\深圳第一新闻门户网站---深圳新闻网.html
16. C:\lucene\lucene-4.5.0\temp\docs\哈尔滨新闻网 [www.my399.com] 新闻最权威及时
的哈尔滨综合新闻.html
17. C:\lucene\lucene-4.5.0\temp\docs\河北新闻网_河北最大的网络媒体_河北第一新闻
门户网站_河北日报报业集团主办.html
18. C:\lucene\lucene-4.5.0\temp\docs\六安新闻网－世界了解六安的窗口 六安走向世界
的桥梁－六安市唯一重点新闻门户网站.html
19. C:\lucene\lucene-4.5.0\temp\docs\百度新闻搜索——全球最大的中文新闻平台.html

20. C:\lucene\lucene-4.5.0\temp\docs\新闻中心_腾讯网.html
```

Figure 3. The Results of LTR Algorithm

4.3 Experimental Result Analysis

In order to judge the performance of the LTR algorithm and the Lucene original scoring algorithm scientifically, the MRR evaluation criteria are introduced. MRR (Mean reciprocal rank) is an international generic mechanism used for evaluating the search algorithm. Namely the first match score of the results is 1, the second match score is 0.5, the n-th match score is 1/n, if there is no matched sentence the score is 0. The final score is the average of all the scores. Its formula is as follows:

$$MRR = \sum_{i=1}^{n} 1/(r_i \cdot n) \tag{6}$$

Among them, r_i is the related document in the position of searching results, n is the total number of queries. The higher the MRR value is, the better the performance of the search algorithm.

Similarly taking user 2 as an example: in the Lucene original scoring mechanism and the relevant documents to the first query ("新闻") position in the search results for 9. The following four queries, documents are 3, 8, 7, 11 respectively. Then the five query MRR value is equal to 0.16. Under the LTR algorithm, first query related document position for 1, the following four queries, documents are 2, 3, 3, 5 respectively, then the five query MRR value is equal to 0.47.

Similarly, a small amount of data of one user does not reflect the problem. In the final experiment, 5 users in Table 4 were queried 20 times in Lucene original scoring mechanism and LRT algorithm with different coefficients calculate the mean of MRR. The results come as shown in Figure 4.

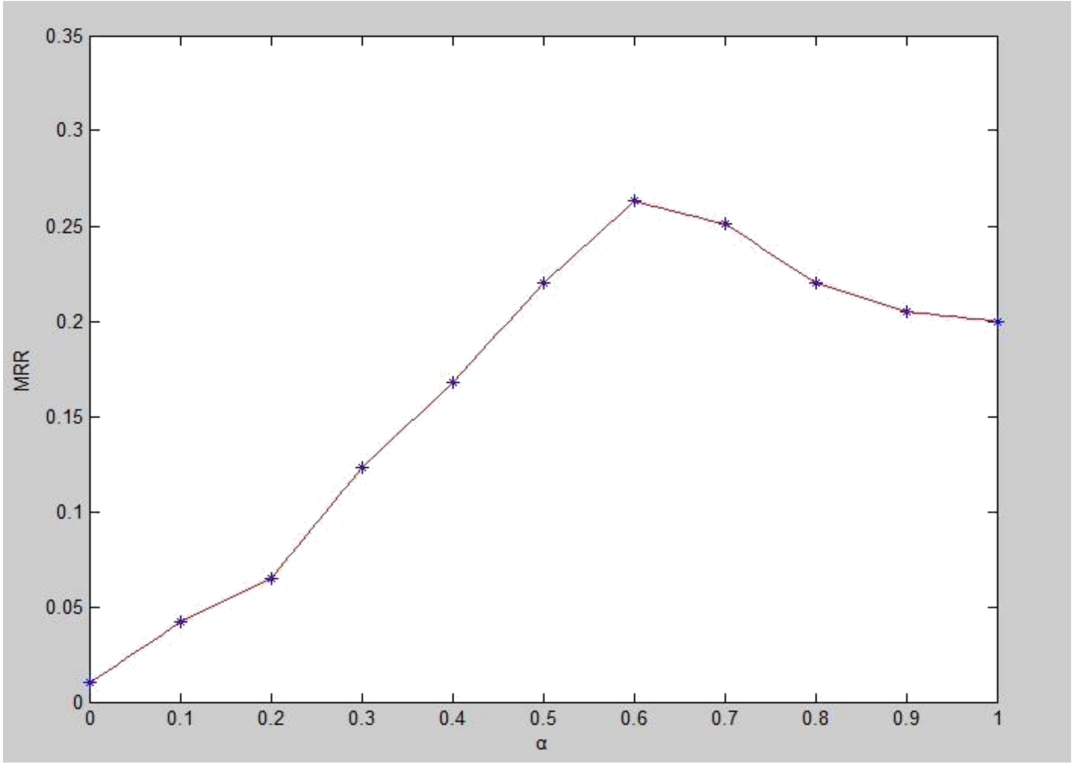


Figure 4. Experimental Results under Different Coefficients

As we can see from Figure 4, the average MRR value of the original scoring mechanism of Lucene is 0.200. But for the LRT algorithm, the average MRR value is only 0.01. The result was very poor. This is because at this time the document similarity score is not given consideration. Just considering the user’s interest label lead to put the cart before the horse, it is very difficult to get the desired results. With α coefficient increasing gradually, the MRR value of the LTR algorithm increases gradually. The value of MRR reached maximum 0.263 when $\alpha = 0.6$. It means that the individual effect is the best at this time and with further increase of the α coefficient, the value of MRR gradually decline. This is because the influence of the label matching score of the algorithm LTR becomes smaller and smaller.

Experimental results show that LTR algorithm can achieve better personalized ranking when coefficient is taken into consideration.

5. Summary

Microblog text is the main media platform for people to understand the Internet, so it serves as the object of study, can accurately reflect the user's attention and interest. In this paper, combining with the Lucene search engine and the social network platform, according to the user's microblog text uses WUK algorithm to extract features. Using the LTR algorithm to improve the personalized Lucene scoring mechanism and using the results of MRR to reflect the effect of scoring mechanism.

The main focus of the paper is the personalized sorting algorithm. In order to realize the personalized search of Lucene, the LTR algorithm is designed to replace the original scoring mechanism of Lucene. The basic idea of LTR algorithm is through text mining methods and Chinese text processing method from Sina microblog users to extract tags of interest. The TF value is based on the calculation of the tag weights. The final page score consists of document similarity score and tags of interest matching scores.

In the experimental process, determine the best solution by adjusting the coefficient α and β to compare the results, so that users can get a good experience. This scheme provides a more effective personalized search scheme. In the future, we plan to design more scientific interest tag extraction algorithm, such as time constraints, the use of machine learning methods to extract more of the summary of the tags, *etc*..

References

[1] B. J. Jansen, A. Spink and T. Saracevic, “Real life, real users, and real needs: a study and analysis of user queries on the web”, Information Processing and Management, vol. 36, no. 2, **(2000)**, pp. 207–227.

[2] R. Krovetz and W. B. Croft, “Lexical ambiguity and information retrieval”, Information Systems, vol. 10, no. 2, **(1992)**, pp. 115– 141.

[3] S. Cronen-Townsend and W. B. Croft, “Quantifying query ambiguity”, Proceedings of HLT, **(2002)**.

[4] J. Teevan, S. T. Dumais and E. Horvitz, “Beyond the commons: Investigating the value of personalizing web search”, Proceedings of PIA, **(2005)**.

[5] C. Silverstein, H. Marais, M. Henzinger and M. Moricz, “Analysis of a very large web search engine query log”, SIGIR Forum, vol. 33, no. 1, **(1999)**, pp. 6–12.

[6] J. Pitkow, H. Schutze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar and T. Breuel, “Personalized search”, Commun. ACM, vol. 45, no. 9, **(2002)**, pp. 50–55.

[7] A. Pretschner and S. Gauch, “Ontology based personalized search”, Proceedings of ICTAI, **(1999)**.

[8] X. Shen, B. Tan and C. Zhai, “Implicit user modeling for personalized search”, Proceedings of CIKM, **(2005)**.

[9] J.-Y. Li，“Social network user feature extraction based on content”, Tianjin：Tianjin University, **(2014)**.

[10] B. Tan, X. Shen and C. Zhai, “Mining long-term search history to improve search accuracy”, Proceedings of KDD, **(2006)**.

[11] G. Jeh and J. Widom, “Scaling personalized web search”, Proceedings of WWW, **(2003)**.

[12] P. Ferragina and A. Gulli, “A personalized search engine based on web-snippet hierarchical clustering”, WWW: Special interest tracks and posters of the 14th international conference on World Wide Web, **(2005)**.

[13] J. Teevan, S. T. Dumais and E. Horvitz, “Personalizing search via automated analysis of interests and activities”, Proceedings of SIGIR, **(2005)**.

[14] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu and Z. Chen, “Cubesvd: a novel approach to personalized web search”, Proceedings of WWW, **(2005)**.

[15] K. Leung, W.-T. Dik, L. Lee and W.-C. Lee, “PMSE: A Personalized Mobile Search Engine”, IEEE Transactions on Knowledge and Data Engineering, Washington, **(2013)**; DC, USA.

[16] N. Ramesh and J. Andrews, “Personalized Search Engine using Social Networking Activity”, Indian Journal of Science and Technology, **(2015)**.

[17] Z. Xu, T. Lukasiewicz and O. Tifrea-Marciuska, “Improving Personalized Search on the Social Web Based on Similarities between Users”, Scalable Uncertainty Management, **(2014)**; Berlin, Germany.

[18] J. Liu, P. Dolan and E. R. Pedersen, “Personalized news recommendation based on click behavior"”, Proceedings of the 15th international conference on Intelligent user interfaces, ACM, **(2010)**.

[19] Z. Ma, G. Pant and O. R. L. Sheng, “Interest-based personalized search”, ACM Transactions on Information Systems (TOIS), vol. 25, no. 1, **(2007)**, p. 5.

[20] K. W. T. Leung, W. Ng and D. L. Lee, “Personalized concept-based clustering of search engine queries”, Knowledge and Data Engineering, IEEE Transactions, vol. 20, no. 11, **(2008)**, pp. 1505-1518.

[21] J. Teevan, C. Alvarado, M. S. Ackerman and D. R. Karger, “The perfect search engine is not enough: A study of orienteering behavior in directed search”, Proceedings of Computer–Human Interaction Conference, **(2004)**; Vienna, Austria.