2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014)

Algorithm of Repeated Results Re-ranking based on Polysemy

GUO Pengwei[1,a],ZHANG Bin[1,b],SUN Da-ming[1,c]

[1]College of Information Science and Engineering, Northeastern University, Shenyang 110004, China

[a]278654101@qq.com, [b]zhangbin@ise.neu.edu.cn, [c]bigming9981@163.com

Keywords: polysemy, re-rank, search, concept lattice

Abstract. In order to make search results better fit users' current search interest, this paper proposes an algorithm of repeated results re-ranking using a model of polysemy. The algorithm considers the characteristics of the keywords to improve the rank of repeated results. Based on the analysis of polysemy of the keywords, we propose a polysemous model of concept lattice, then we combine with the user interest model to change the rank of repeated results in a search session. The method of this paper considers the impact of polysemy of keywords which may improve the ranking. The experimental results show that the process based on the polysemy of the keyword can reduce the length of the search session, especially when the keywords have multiple meanings.

Introduction

Inrecent years, with the development of Internet technology, information on the Web is having an explosive growth which leads to difficulty for users to obtain relevant information. Emergence of search engines has greatly improved the efficiency of accessing relevant information. Search engine is a tool based on keywords which users submitted, so the quality of keywords always determines the effectiveness of the search. However, due to different search intention and different search experience of different users, the following two conditions often occur: the same keyword may have different search intentions and different keywords may have the same search intention. Both conditions mentioned above lead to that search results does not contain the result which the user interested in or these kinds of results have a lower rank. To deal with these problems, there have been researches of rank and re-rank based on users' interest in order to improve the search results [1, 2].

This paper focuses on the impact of polysemy of query in search re-ranking. A word that contains more than one meaning is called polysemy. Since polysemy is widespread, the keywords in a query are always polysemous. This leads to that the set of search result is constituted of each classification corresponding to each ambiguous word meaning of the keywords. If no distinction, it is difficult to give results which users interested in a higher rank, so that the rank of search results can't meet users' preference and this reduce the efficiency of finding the relevant page for users. So, it is important to distinguish the polysemy of keywords in queries for more accurate rank.

In this paper, we propose a method of re-ranking for repeated results in a search session. The method is based on analysis of polysemy of keywords, and then determines the user's current search interest combining with long-term and short-term interest. And then we revise the rank of repeated results in order to satisfy user's current interest. Different from current researches, this paper is based on the polysemy model of keywords, and we confirm one meaning which user currently interested in, that we determine the user's range of interest, so that the accuracy of rank can be greatly improved.

The experimental results show that, the polysemy model of keywords based on log of search can exposit the characteristics of query effectively and the model of user's interest can determine the user's range of interest accurately. The results also show that the stronger the ambiguity of the query is, the more effective our method is comparing with previous methods.

Related Research

In this section, we explain related research from the following two aspects: related research in polysemy and related research in re-rank.

Polysemy. In the text-based process of information, ambiguity often occurs and affects the results. Ambiguity can be divided into two specific situations: homonymy and polysemy[3]. We typically believe that, in the process of retrieval, if only related documents of one meaning is retrieved among all the documents of all the meanings, the accuracy of the retrieval will increase[4]. Therefore, in the field of information retrieval, resolving the issue by word sense disambiguation is usually considered to improve the efficiency of information retrieval. So, in recent decades, research has been going on word sense disambiguation. Generally speaking, there are two ways to deal with WSD: methods based on manually creating rules and methods based on formal data source [5].

This paper is based on the research of [6], and further expands on its basis. The biggest difference with [6] is that before re-ranking the repeated results we first establish a polysemy model of concept lattice of queries, and based on this model we arrange the repeated results to one of the meaning of keywords, then we re-rank them. Existing studies did not consider the polysemy of queries or the possible relationship among the repeated results.

Re-rank. Due to the inaccurate keywords of queries and the limitations of the ranking algorithm, the results returned by search engine can't meet the user's personal need. So research of ranking of personalization based on users' interests has received a lot of attention. According to the length of time, users' interests can be divided into long-term interest and short-term interest.

The long-term interest of users' include the history of browsing, the log of search engine, and users' personal information such as documents on their desktop. Teevan et al created a user profile based on documents on their desktop and proved that this kind of information could be used for personalized ranking [7]. Users' short-term interests usually include clicks, habits of users browsing and the change of vision. Joachims et al proposed a method which automatically optimizes the search quality by using the clicks [8].

This paper combines the long-term interests and short-term interest. First, we use the search log which belongs to long-term interests to build a polysemy model, and then we use clicks which belong to short-term interests to re-rank repeated results. The ways we use clicks in this paper could be found in reference [6].

Concept Lattice Model Based on Search Log

In order to re-rank repeated results, we need to figure out the meaning each repeated result related to under the current query, so we need to model the polysemy of keywords. This paper establishes the polysemy model of concept lattice based on a commercial search engine log.

Polysemy Model of Concept Lattice. Formal Concept Analysis is a mathematical analysis tools based on queuing theory, and is first proposed by Rudolf Wille [9]. There are researches based on FCA in informational retrieval. In this paper, we use concept lattice in FCA to model polysemy which representing each meaning of the keyword using nodes in a concept lattice. Concept lattice can describe the relationships among a concept and its child concept, and this is very similar with the characteristic of polysemy. So the features of the concept lattice and the forms of representing concepts of concept lattice itself are basis that we model polysemy.

In a concept lattice, each node represents a concept, and is composed of formal object and formal attribute which express the correspondence between objects and attributes. In this paper, the formal object and formal attribute are vectors of words. And in the theory of FCA, the vertex and the nadir are usually fictional in many cases to make the lattice complete.

A concept lattice C is composed of several related concepts, and each concept corresponds to a node, that, $C_K = \{N_1, N_2, \cdots, N_m\}$

Each node N_i is a four-tuple which defined as follows，

$$N_i =< FO_i, FA_i, P_i, C_i >$$

FO_i represents for formal objects and is composed of vector of words; FA_i represents for formal attribute which is also composed of vector of words; P_i represents for father nodes which is a set of nodes; C_i represents for child nodes which is also a set of nodes. Father nodes and child nodes are defined as follows,

$$Parents = \{Node \mid H - H_N = 1\}$$

$$Children = \{Node \mid H_N - H = 1\}$$

H represents for node depth. In this paper, we specify the node depth of vertex is 0. And we define initial nodes which having node depth 1, that,

$$N_{Initial} = \{N \mid H_N = 1\}$$

It is obviously that the father node of the initial node is the vertex of concept lattice. Initial nodes describe the most basic classification of concept. Except the vertex, the nadir and initial nodes, the following nodes are called lower nodes, that,

$$N_{Lower} = \{N \mid H_N \geq 2\}$$

Given a keyword $Word$ which including n meanings, combing the definition of initial nodes we get that,

$$Word = \{m_1, m_2, \cdots, m_n\} = N_{Initial}$$

So we can combine polysemy with concept lattice, and we have rules in the model as follows:

1. According to the characteristics of the concept lattice, $N_{Initial}$ represents for the broadest classification of a concept, and can be used to classify the basic meanings of polysemy;

2. Each node except the nadir in a concept lattice may have one or more than one child node C_i, this represent the refinement of the current father concept;

3. The lower nodes in a concept may have one or more than one father node P_i, and this represent that the current concept may belong to one or more than one concepts branch.

Based on the model of concept lattice, we can define a polysemous keyword as follows,

Definition 1: if a concept lattice which built from $Word$ of a queryhaving $N_{Initial} \geq 2$, we call the keyword of the query is polysemy.

We take “apple” as an example, and we establish the concept lattice of it in Fig1.

In Fig. 1, we say keyword apple is polysemy because of its $N_{Initial} \geq 2$.

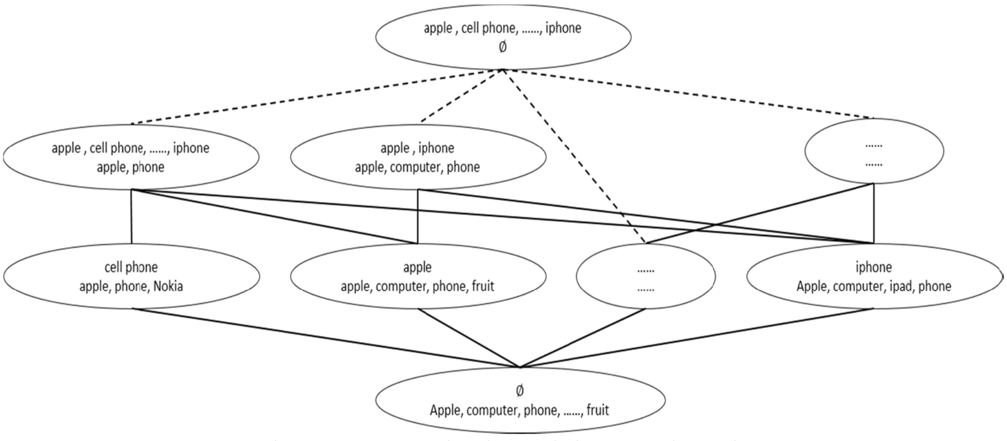
apple , cell phone, ..., iphone
Ø
apple , cell phone, ..., iphone
apple, phone
apple , iphone
apple, computer, phone
......
......
cell phone
apple, phone, Nokia
apple
apple, computer, phone, fruit
......
......
iphone
Apple, computer, ipad, phone
Ø
Apple, computer, phone, ..., fruit

Fig 1 Concept lattice of the word apple

Algorithm of Building the Model of ConceptLattice. Our concept lattice is based on search logs, and it is very time-consuming to build concept lattice for each keyword, however this process could be done offline which do not affect the efficiency of re-ranking repeated results online, so we could ignore the efficiency of this part. The algorithm of building concept lattice based on search log is as follows.

```
ALGORITHM 1：BCOL(Building Concept Lattice Based On Logs)
 Input：search logs，core word
```

```
Output：polysemy of concept lattice
Process：
for each query Q in logs
    if  word∈Q
        column ← split Q;
        get last URL of Q;
        get document of the URL;
        V ← split document;
        row ← remove stopwords of V;
    End if
End for
Pcbo(column,row)
return concept lattice of word
```

Pcbo[10] is an existing algorithm for building concept lattice which needs rows and columns of the matrix as input parameters, and in this paper, we specify the number of CPU is 10 and the recursion depth is 3. Algorithm 1 calculates with only one core word, and the whole process is a loop of Algorithm 1 based on the specified core word. In Algorithm 1, we first match all the queries in search log based on the current core word, if a query contains the core word, we segment the query and get the result as column for Pcbo. And then, according to the query, we extract the last clicked URL to get the corresponding document. After removing stopwords, we do segmentation and statistic of the word frequency, and get the result as rows for Pcbo. After circulating all the queries in the log, we get the inputs of Pcbo, then we use this algorithm to build the concept lattice based on the core word.

During extracting the document of a URL, we suppose that during a search session, the last document the user clicked is the document the user interested in which we have mentioned above.

In summary, we have built polysemy model of concept lattice for each core word based on the search log. These models contain each meaning of a core word and related concepts. After this, we need a user interest model to get the users' interests for re-ranking repeated results.

Re-ranking Repeated Results Based on Concept Lattice

Based on the polysemy model of concept lattice, we can get each meaning of a polysemous word, and in order to combine the users' interests, we need to build a user interest model to select the concept which users' interested in from the concept lattice and design an algorithm of re-ranking.

The User's Interest Model. For the purpose of finding out the user's interest and the change of interest, we need to build a user interest model. The user interest model we propose include two parts: initial interest and click interest. Initial interest represents the interest the user usually interested in corresponding to the user's long-term interest; click interest is expressed during the searching which is corresponding to the user's short-term interest.

The initial interest can be gained from user's search history, and we can find out the user's common interest from the initial interest. Considering that the search history of user is difficult to get, this paper only take the registered users of our search system into account, so we can record the history of the user, and when registering ,we require the user to fill personal interest which is another source of user's long-term interest.

The click interest means that user expresses during the searching, and the ways we use the click interest just as reference [6]. So we also classify the clicks into three kinds: clicked, not noticed, and noticed and considered but not clicked. When the user clicked a result, we consider that the user is interested in this result, if the user noticed and considered but not clicked a result, we consider that this result doesn't meet the user's interest, if the user no noticed the result we can't certain if this result meets user's need.

Combining with the polysemy model of concept lattice, in order to calculating the similarity between user's interest and the nodes in the concept lattice, we represent user's interest I as follows.

$I = I_b + I_c$

I_b is the initial interest, and I_c is the clicked interest which are just as above.

$I_b = I_H + I_R$

I_H represents interest which is find from the history of the user, and the way we get I_H could reference Algorithm 1. I_R represents interest the user filled in when they registered.

$I_c = I_Y + I_N$

Y represents the interest user is interested in and N represent the interest user is not interested in according to the cases we classify the clicks.

The model of interests we defined is expressed by vectors of words, and we calculate the similarity using cosine similarity. In fact, the model of user interest is a probability distribution on the polysemy model of concept lattice, just as follows.

$$\begin{aligned} Interest_{user} &= P(I_b + I_c \mid C) \\ &= P(H + R + I_Y + I_N \mid \{nodes\}) \end{aligned}$$

User's interest is multidimensional, and distributed on each node in concept lattice, if we suppose α_i as probability that the user's interest fits the nodes, we have the following formula.

$\alpha_i = \alpha_H + \alpha_R + \alpha_{I_Y} + \alpha_{I_N}$

And we can express $Interest_{user}$ as follows.

$Interest_{user} = \sum \alpha_i node_i$

Selection Algorithm of Nodes in Concept Lattice. The polysemy model of concept lattice expressed the meanings which the keywords may have, based on this we need the model of user interest to gain the user's current interest.

```
ALGORITHM 2：SIN(Selecting Interested Node)
Input：concept lattice，user's interest
Output：user's current interest on the nodes
Process：
if $I_c$=null
    for each N in $N_{Initial}$
        compute Sim（$I_b$， N）;
    End for
    get N of max（Sim（$I_b$， N））;
    return N;
End if
else
    for each N in $N_{Initial}$
        compute Sim（$I_N$， N）;
        if Sim（$I_N$， N）>α
            remove N from $N_{Initial}$;
        End if
    End for
    if $N_{Initial}$=null
        return N of min(Sim（$I_N$， N）);
    End if
    else
```

```
          for each N in  $N_{Initial}$
      compute Sim（$I_Y$， N）
          End for
      return N of max(Sim（$I_Y$， N））;
      End else
End else
```

Sim is a function which calculates the similarity, in this paper we use cosine similarity. Through Algorithm 2 we can calculate α_i in the model of user interest.

Algorithm 2 first judges that if there is click interest, if there is no considering that some registered users did not fill in with the interest when they registered, calculates the similarity between the initial interest and the nodes and returns the node which has the maximum similarity. If there is initial interest, the algorithm calculates the similarity between I_N and the nodes, and removes the nodes which are greater than the threshold, and then returns the remaining nodes. Among the remaining nodes, the algorithm calculates similarity between I_Y and the nodes, finally return the node which has the biggest similarity. In this algorithm, because of removing the nodes by I_N, we could greatly improve the efficiency of the algorithm.

Re-ranking Algorithm of Repeated Results. Based on Algorithm 1 and Algorithm 2 we can determine the node which the user is current interested in, so when the user starts to search, we can calculate the similarity between the results and the node, and then we can re-rank with the similarity, and the algorithm is as follows.

```
ALGORITHM 3：RRR(Re-ranking Repeated Results)
Input：node N which the user interested in，the returned repeated results
Output：the result of re-rank
Process：
For each r in Results
    compute Sim (r, N)
End For
    sort Sim(r,N)
return r
```

We can get the finally rank for each search during the search session by Algorithm 3, and this rank is based on the polysemy of keywords with the user’s long-term and short-term interests.

The core of this algorithm is calculating the similarity between the interested node and the result, and rank with the similarity. In the experiment, considering the efficiency and usability factors, we only calculate the first 30 results which is consistent with the user’s browsing habits.

Experiment

In this paper, the experiment mainly contains two parts: establishing the polysemy of concept lattice and test the effective of the algorithms we proposed.

Data Source. In the experiment, we need a search engine log to build the polysemy model of concept lattice. In this paper we use search logs of two months of 2011 which Sogou official released as data source. In the re-ranking test, we extracted 132 queries randomly and submitted these queries to Baidu search engine, then we took the returned results as test objects.

Comparative Experiment. The experiments of this paper compares with the method proposed in reference [6]. We introduce the concept of ambiguity which refers to the degree of polysemy. In this paper, we define ambiguity A as the number of initial nodes of the polysemy model of concept lattice.

We use NDCG as evaluation criteria, which can be calculated as follows.

$$N_n = Z_n \sum_{i=1}^{n} \frac{2^{r(j)} - 1}{log(1+j)} \qquad (1)$$

r represents for related settings, the denominator is conversion factor, Z_n is normalization factor. And we get the experiment results between our algorithms based on FCA and the original method in reference [6] as follows.

Table 1 Values of NDCG

NDCG @ Rank	A=1 (Query=15)		A=2 (Query=22)		A=3 (Query=19)		A=4 (Query=28)		A=5 (Query=27)		A=6 (Query=21)	
	Ori	FCA	Ori	FCA	Ori	FCA	Ori	FCA	Ori	FCA	Ori	FCA
1	0.853	0,849	0.861	0.861	0.853	0.861	0.866	0.875	0.856	0.873	0.852	0.879
2	0.847	0,846	0.853	0.854	0.847	0.853	0.863	0.873	0.852	0.870	0.849	0.877
5	0.836	0.837	0.847	0.844	0.837	0.841	0.858	0.872	0.851	0.869	0.845	0.874
10	0,840	0.838	0.837	0.838	0.834	0.840	0.853	0.869	0.850	0.867	0.846	0.876
20	0,837	0,835	0.838	0.836	0,836	0.837	0.855	0.870	0.847	0.866	0.843	0.875

According to Table 1, we could get Fig. 2 as follows.

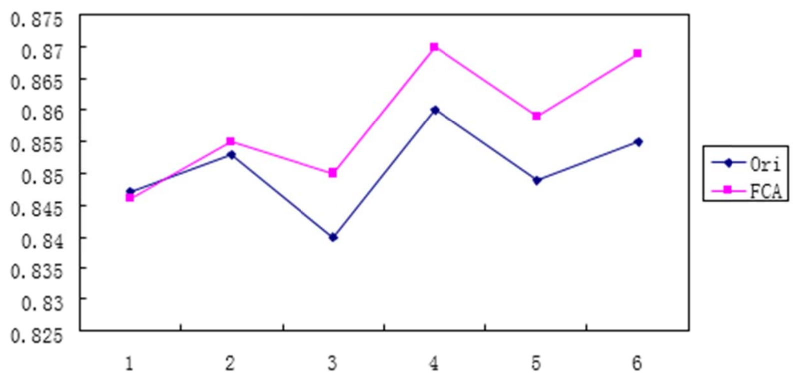

Fig 2 Values of NDCG

It is obviously from Fig. 2 that our models and algorithms are more effective when the ambiguity gets bigger. And this shows that when a keyword is polysemous, we can re-rank repeated results better. The experiment results also illustrate that with the analysis of polysemy and the user's interest, we can get better rank for search engine results.

Summary

This paper studied the re-ranking of repeated results in search session which composed of several related searches. In connection with the lack of considering polysemy of existing methods, we proposed a polysemy model of concept lattice based on search logs. Combining with the user interest model, we designed three algorithms to re-rank repeated results returned by the search engine. In the experiment, we compared our method with one existing method and shows that our method could gain better rank when the keywords of queries are polysemous.

References

[1] Jaime Teevan, Susan T. Dumais, Eric Horvitz: Personalizing search via automated analysis of interests and activities. SIGIR 2005:449-456.

[2] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling

the impact of short- and long-term behavior on search personalization. InProc. SIGIR, pages 185-194, Portland, OR, 2012.

[3] Christopher Stokoe: Differentiating Homonymy and Polysemy in Information Retrieval. HLT/EMNLP 2005。

[4] Christopher Stokoe, Michael P. Oakes, John Tait: Word sense disambiguation in information retrieval revisited. SIGIR 2003:159-166。

[5] Mark Sanderson: Retrieving with Good Sense. Inf. Retr. (IR) 2(1):45-65 (2000)。

[6] Milad Shokouhi, Ryen W. White, Paul N. Bennett, Filip Radlinski: Fighting search engine amnesia: reranking repeated results. SIGIR 2013:273-282.

[7] Jaime Teevan, Susan T. Dumais, Eric Horvitz: Personalizing search via automated analysis of interests and activities. SIGIR 2005:449-456.

[8] Thorsten Joachims: Optimizing search engines using clickthrough data. KDD 2002:133-142.

[9] Rudolf Wille: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. ICFCA 2009: 314-339.

[10] Petr Krajca, Jan outrata, Vilem Vychodil: Parallel Recursive Algorithm for FCA. CLA 2008: 71-82.