Artif Intell Rev (2014) 41:81–96
DOI 10.1007/s10462-011-9299-6

Optimizing ranking method using social annotations based on language model

Kunmei Wen · Ruixuan Li · Jing Xia · Xiwu Gu

Published online: 3 January 2012

Abstract Recent research has shown that more and more web users utilize social annotations to manage and organize their interested resources. Therefore, with the growing popularity of social annotations, it is becoming more and more important to utilize such social annotations to achieve effective web search. However, using a statistical model, there are no previous studies that examine the relationships between queries and social annotations. Motivated by this observation, we use social annotations to re-rank search results. We intend to optimize retrieval ranking method by using the ranking strategy of integrating the query-annotation similarity into query-document similarity. Specifically, we calculate the query-annotation similarity by using a statistical language model, which in a shorter form we call simply a language model. Then the initial search results are re-ranked according to the computational weighted score of the query-document similarity score and the query-annotation similarity score. Experimental results show that the proposed method can improve the NDCG score by 8.13%. We further conduct an empirical evaluation of the method by using a query set including about 300 popular social annotations and constructed phrases. More generally, the optimized results with social annotations based on a language model can be of significant benefit to web search.

Keywords Social annotation · Language model · Query-annotation similarity · Query-document similarity

1 Introduction

Along with the rise of Web 2.0 technology, it is now customary for social annotations (Cattuto et al. 2009) to be created by web users as a summary of the content of a web page. Some

K. Wen (✉) · R. Li · X. Gu
Intelligent and Distributed Computing laboratory of the HuaZhong, University of Science and Technology, WuHan, China
e-mail: kmwen@hust.edu.cn

J. Xia
709th Research Institute China Shipbuilding Industry Corporation, WuHan, China

scholars argue that the annotation is a term used to augment the text content with additional information (Ball 2009). It is useful for us to examine the annotations of a web page in order to understand the meanings of a web page. Nowadays the technology of social annotations is becoming a new web service that can help users share, classify, and discover the web resources that interest them. The result is that there have been more and more studies on the social annotations. So far these studies have mainly concentrated on four areas: the concept of social annotations, the development and the working principles of social annotation systems, the core elements—tags, and the users of community tagging systems (Hammond et al. 2005). Moreover, in recent years, social annotation has gained increasing popularity in many web-based applications (Zhou et al. 2008). The best way of utilizing social annotations to produce effective web search is becoming more and more popular.

As is generally known, ranking methods play an important role in enhancing the quality of search results. Moreover, ranking methods evaluate the similarity between the query and the content of web page. So computing the similarity score between the query and web page annotations that are tagged by annotators is a good way of optimizing the similarity between the query and the web page. At present, there are many ranking methods that calculate the query-document similarity (Baeza-Yates and Ribeiro-Neto 2005), which is a common ranking strategy. By contrast, there are few methods that integrate social annotations into the similarity computation. This lack of methods has motivated us to integrate social annotations into the common ranking method so as to re-rank the search results. More specifically, we adopt a language model (Ponts and Croft 1998) that computes the query-annotation similarity. In our method, we combine the query-document similarity and the query-annotation similarity to establish the weighted score. We optimize web search by re-ranking the initial search results according to the weighted scores. This approach is different from the methods that currently exist.

The goal of this paper is to improve the quality of search results by utilizing social annotations. In this paper, language model is used to calculate the query-annotation similarity. Because the social annotations of each web page consist of a sequence of terms, we do not need to parse social annotations generally. Instead, we can easily construct a corpus. The similarity measures how well the queries match with the web page annotations. Specifically, the initial search results are re-ranked according to the computational weighted score of the query-document similarity score and the query-annotation similarity score. That is, we attempt to make use of the annotations that contain rich information to benefit the web search. This method makes the search results more satisfactory for the searchers. Naturally, the precision of query (Baeza-Yates and Ribeiro-Neto 2005) is improved thanks to this integration of social annotations.

Our approach is mainly divided into three parts in this paper. First, we collect some web pages from the social bookmarking site (Delicious: http://del.icio.us), which organizes and shares social annotations. Secondly, we standardize these social annotations and remove redundant annotations. Finally, the query-annotation similarity is computed using a language model and the weighted score of the query-document similarity score and the query-annotation similarity score is calculated. Then the initial search results are re-ranked according to the weighted score. The main contributions of this paper are the following. (1) The language model of the social annotations of a web page can be evaluated effectively. (2) The method proposed can calculate the query-annotation similarity. (3) The method proposed, which re-ranks the initial search results according to the computational weighted score of the query-document similarity score and the query-annotation similarity score, performs better than the methods that do not use the social annotations.

The rest of the paper is organized as follows. Section 2 reviews related work on social annotations and their combination with web search. The core of our paper, Sect. 3, proposes a method that is used to re-rank the initial search results. The query-annotation similarity is computed using a language model. For the sake of simplicity, such a method is called ranking with social annotations. Section 4 shows the experimental results. Finally, in Sect. 5, we draw some conclusions and express our thanks in Acknowledgments.

2 Related work

Some work has been done on applying social annotations to web search and applying language models to information retrieval. In the section that follows immediately, we give a brief review of the related work.

2.1 Application of social annotations in web search

In recent years, much work has been done in this area of research. Some scholars have concentrated on the utilization of social annotations to improve web search. HeyMann et al. (2008) assert that social bookmarking is useful for web search, though it may currently lack the size and distribution of tags necessary to make a significant impact.

Existing research focuses mainly on integrating social annotations into the web search ranking. Bao et al. (2007) explore the use of social annotations to improve web search by proposing two methods: the first is called SocialSimRank, which is used to find the latent semantic association between queries and annotations; the second is called SocialPageRank (Bao et al. 2007), which takes into account the popularity of web pages. Using the two methods given above, Bao et al propose a dynamic ranking method with social annotations. In addition, the general method of many scholars is not to use the similarity between queries and annotations to calculate the query-document similarity. Yan et al. (2009) have proposed an method called Query-Tag-Gap (QTG) to re-rank search results for better user satisfaction. They show that the gap between query and social annotation distribution correlates inversely with the search user's satisfaction. However, none of the methods proposed above adopts a language model that is used to calculate the similarity between the query and social annotations. This is the principal difference between the approach of these other researchers and the method proposed in the present paper. We have adopted a language model because annotations have semantic features and because the social annotations of web pages consist of a sequence of terms, we can easily construct a corpus without parsing the social annotations. Therefore the query-annotation similarity can easily be calculated.

Lately, scholars have researched the relationship between social annotations and the intentions of user queries. A method referred to as TagQV has been proposed by Liu et al. (2009) to identify a user's vertical search intention as expressed in the query using social annotation propagation. Also Noll and Meinel (2008) have analyzed and compared the two different but related types of "metadata" about web documents: social annotations provided by readers of web documents and search queries made by users trying to find web documents. So we have also focused on the relationship between queries and annotations. If we find a similarity between the two, then we could optimize web search by integrating social annotations. In the present paper, we propose a method of re-ranking initial search results according to the computational weighted score of the query-document similarity score and the query-annotation similarity score. This use of annotations is the main difference between our method and that of the works above.

2.2 Application of language models in information retrieval

Recently, language model has been successfully applied to many information retrieval problems. Everett and Borgatti (2008) reviews the existing work about applying language model to information retrieval, summarizes their contributions, and takes note of outstanding challenges.

Since a statistical language model (SLM) has been applied to information retrieval for the first time, many studies have been continued in the field, such as Hiemstra (1998) and the BBN group (Miller et al. 1999). Although the details differ between these methods, the basic idea (Song and Croft 2006) is the same. That is, each document is viewed as a language sample. A query is estimated by computing the probabilities of generating the query in a large corpus of documents, which is the query-document similarity actually. As pointed out by Zhai and Lafferty (2004), the proposed method is to estimate a language model for each document and then to rank the documents according to the query-document similarity. However, our method in the present paper is to build a language model for the web page annotations, but not for the whole web page. Then we re-rank the initial search results which ranked by the query-document similarity according to the computational weighted score of the query-document similarity score and the query-annotation similarity score. Recently, methods to improve the language model by using positional information have been proposed. Lv and Zhai (2009) have proposed a novel positional language model (PLM), which defines a language model for each position of a document and evaluates the document by using the scores of its PLMs.

Combinations of the language model and social annotations are attractive and promising. Using social annotations to improve language model for information retrieval was proposed by Xu et al. (2007a). Also the same paper explored how to utilize social annotations to smooth the language model (Xu et al. 2007b). Unlike the works mentioned above, we make use of language model to compute the query-annotation similarity, instead of using social annotations to improve a language model.

3 Proposed method

In this section, we propose a method to optimize web search ranking using social annotations based on language model.

We first describe the overview of proposed method. Then how to evaluate the language model of social annotations is explained. Then, based on the language model, calculation of query-annotation similarity is explained. Finally, we re-rank the initial search results according to the computational weighted score of the query-document similarity score and the query-annotation similarity score in order to optimize the web search ranking method.

3.1 Overview of proposed method

The basic idea of our method is to incorporate social annotations into the content of web page for improving the web search ranking. The detailed descriptions are explained as follows:

1. Basically, the input of our method is a collection of top K initial search results denoted by returned by a general search engine, where R_i is a resource (e.g., document or web page) and A_i is a set of tags assigned to a specific resource R_i. We denote $V_A = \{w_j | j = 1, L\}$ as a temporary tag corpus with size L and w_j is a social annotation.

2. Further we adopt the Maximum Likelihood Estimation (MLE) (Pratt 1976) and the smoothing method (Jeffreys 1948), which was proposed by Lidstone, Johnson and so on, to establish the language model for the social annotations in this paper. Formally, we denote a vector $z = (P(w_1|A_i), \ldots, P(w_L|A_i))$ as the language model of the social annotations of a specific web page A_i.
3. The basic idea of our method is to incorporate the query-annotation similarity into the query-document similarity to optimize web search ranking strategy. Formally, $Q = \{q_1, q_2, \ldots, q_m\}$ denotes a query, where $q_i \in V_A$. We denote the query-annotation similarity score as $P(Q|A)$ and denote the query-document similarity score as $P(Q|R)$. Specifically, the ranking score is computed by $Score = \alpha * P(Q|R) + \beta * P(Q|A)$, where α and β satisfy the condition $\alpha + \beta = 1$ and the weight α and β is experience value determined by a large number of experiments in this paper. Then according to the score, we re-rank the initial search results in decreasing order and we obtain the optimized results denoted by $D' = \{(R_1^{'}, A_1{'}), \ldots, (R_k{'}, A_k{'})\}$. The web search ranking method proposed in this paper is that the higher the $P(Q|A_i)$, the corresponding web page A_i is higher in the optimized results.

To this end, we propose a method to optimize web search ranking by using social annotations based on language model. At a high level, tow key points in the proposed method are how to evaluate the language model of social annotations and how to calculate the query-annotation similarity based on the language model.

3.2 Construction of the language model of social annotations

Through an analysis of the features of social annotations crawled from the web site http://del.icio.us, we have found that the social annotations are sparse and have some unrelated words. Because the assighed social annotations on the Web are authored freely, quality of these annotations vary largely depending on many different reasons (e.g. author's expertise). Therefore, in the first place, social annotations need to be standardized and duplicate annotations need to be eliminated. In the present paper, the API interface of WordNet (WordNet: http://wordnet.princeton.edu/) was adopted to standardize the social annotations and a fast algorithm was adopted to remove any duplicate annotations (Chen and Wang 2000). In this paper, we have called the fast elimination of redundant annotations the AFE operation.

Then we adopt MLE to estimate the language model of social annotations. MLE is a method that estimates the largest generating probability of samples. In our method, MLE estimates the language model's parameters of the social annotations of a specific web page. Based on the language model theory, before any specific model estimation we must have a corpus (Robertson 1977). In this paper, a temporary corpus refers to the collection of social annotations in the top K initial search results.

Basically, the input of our method is a collection of the K initial search results denoted by $D = \{(R_1, A_1), \ldots, (R_k, A_k)\}$ returned by a general search engine, where R_i is a resource (e.g., document or web page) and A_i is a set of annotations assigned to a specific resource R_i. We denote $V_A = \{w_j | j = 1, \ldots, L\}$ as a temporary corpus with size L and w_j is a social annotation. $A_i = \{a_i \in V | i = 1, \ldots, n\}$ denotes the set of the social annotations of a specific web page that have been standardized. $P(w_j|A_i)$ is the random generating probability of w_j in the social annotations A_i of a specific web page. $A = \{A_i | i = 1, \ldots, k\}$ denotes the set of A_i , with A_i corresponding to R_i. The vector $(P(w_1|A_i), P(w_2|A_i), \ldots, P(w_L|A_i))$ can be viewed as the language model of the social annotations of a specific web page. So, finally, we need to obtain K vectors as the language models of the K initial search results. Because the social annotations are sparse, the above MLE algorithm surely gives rise to a zero-probability problem. Moreover, the Smoothing method is a key technique for solving the zero-probability problem (Ding et al. 2006). An additive smoothing method is adopted.

Given the K initial search results of a specific query, the social annotations of the K web pages can be modeled by Method 1 named Annotation's Language Model Evaluation (ALME).

Method 1: Language model evaluation of annotations

Function ALME()

Input: $A = \{A_i | i = 1, \ldots, k\}$

Step 1 Init:$Null \longrightarrow V$;

Step 2 For each A_i In $A = \{A_i | i = 1, \ldots, k\}$ do

$V_A = V_A + A_i$; THEN $V_A = AFE(V_A)$;

Step 3 For each A_i In $A = \{A_i | i = 1, \ldots, k\}$do

From $j = 1$ to L do

$P(w_j | A_i) = \frac{C(w_j, A_i)+1}{\sum_W C(w_j, A_i)+L}$;

Step 4 For each A_i In $A = \{A_i | i = 1, \ldots, k\}$ do

Output($(P(w_1|A_i), P(w_2|A_i), \ldots, P(w_L|A_i))$);

End Function

Here, V_A denotes the temporary corpus that have been standardized and processed by the operation AFE given above. In Algorithm 1, $C(w_j, A_i)$ denotes the frequency of w_j in the specific web page social annotations $A_i = \{a_1, a_2, \ldots, a_n\}$.Here L denotes the size of the temporary corpus.

Step 1 initializes the corresponding variable including the temporary corpus V_A. Step 2 introduces how to get the temporary corpus of the social annotations of the K initial search results. Step 3 gives the formula that calculates the generating probabilities of term w_j in the independent random experiment of the social annotation of a specific web page, which is denoted by $P(w_j|A_i)$. Step 4, the output function output the K vectors to show the K language models of the social annotations of the K initial search results. From the output, we can observe the intermediate results for preparing the next algorithm.

The time complexity of the ALME algorithm is $O(K \times L)$. We have a conclusion that the speed of the method 1 is mainly determined by the multiplication of the size of the K initial search results and the size of temporary corpus.

3.3 Query-annotation similarity calculation

Because the social annotations are sparse (Hong and Chi 2009), it is necessary for us to construct queries to evaluate the effectiveness of integrating social annotations into web search. Assume that $Q = \{q_1, q_2, \ldots, q_m\}$ denotes a query, and $q_i \in V_A$. It is clear that for the social annotations $A_i = \{a_1, a_2, \ldots, a_n\}$ of a special web page, a_i also belongs to the temporary corpus V_A. Given the probability of generating Q in A_i's language model, query likelihood is denoted by $P(Q|A_i)$ which is also called query generating probability. Actually the query likelihood reflects the query-annotation similarity.

Here, we also assume that the size of the temporary corpus is L. Given a constructed query $Q = \{q_1, q_2, \ldots, q_m\}$, and the social annotations $A_i = \{a_1, a_2, \ldots, a_n\}$ of a web page, the query-annotation similarity is computed by Method 2 named Query-Annotation Similarity Evaluation (QASE).

Method 2: Query-annotation similarity evaluation

Function QASE()

Step 1 Input: $Q = \{q_1, q_2, \ldots, q_m\}$

Step 2 From $i = 1$ to K do

GetVector($(P(w_1|A_i), P(w_2|A_i), \ldots, P(w_L|A_i))$);

StoreSimilarity

$(P(Q|A_i) = \prod_{j=1}^{m} P(q_j|A_i) = \prod_{w \in Q} P(w|A_i)^{C(w,Q)})$;

Step 3 From $i = 1$ to K do

Output($(P(Q|A_i))$);

End Function

In the present paper, a multinomial model is adopted to calculate the query likelihood, which represents the query-annotation similarity. The query is viewed as the sequence of results of a multinomial random experiment. In Step 2, the function GetVector is to obtain the language model of social annotations of a specific web page and the function StoreSimilarity is to store the relevance weight between the query and social annotations. $P(w|A_i)$ denotes an element of the vector $(P(w_1|A_i), P(w_2|A_i), \ldots, P(w_L|A_i))$, and the frequency of the term w in query Q denoted by $C(w, Q)$ must be considered in calculating the query-annotation similarity score. The function Output is to output the relevance weight between the query and the social annotations of a web page, which is the query-annotation similarity.

The time complexity of the QASE algorithm is $O(K \times m)$. Generally, the size of Query Q isn't large. So we can approximately consider the time complexity is $O(K)$.

3.4 Search results optimization based on social annotations

Suppose the set of K initial search results with respect to the query Q is $D = ((R_1, A_1), (R_2, A_2), \ldots, (R_K, A_K))$. We re-rank the K initial search results according to the computational weighted score of the query-document similarity score and the query-annotation similarity score decreasingly in a decreasing order. That is, we intend to optimize retrieval ranking method by using the ranking strategy of integrating the query-annotation similarity denoted by $P(Q|R)$ into query-document similarity denoted by $P(Q|A)$. Specifically, the ranking score is computed by $Score = \alpha * P(Q|R) + \beta * P(Q|A)$, where α and β satisfy the condition $\alpha + \beta = 1$ and the weight α and β is experience value determined by a large number of experiments in this paper. Then according to the score, we re-rank the initial search results in decreasing order and we obtain the optimized results denoted by $D' = ((R_1^{'}, A_1{'}), \ldots, (R_k{'}, A_k{'}))$. Because of incorporating the query-annotation similarity into the query-document similarity, the quality of the web search results is optimized.

Suppose that with respect to a specific query denoted by Q, the initial K search results denoted by $D = ((R_1, A_1), (R_2, A_2), \ldots, (R_K, A_K))$ and the optimized results denoted by $D' = ((R_1^{'}, A_1{'}), \ldots, (R_k{'}, A_k{'}))$. The re-ranking strategy is presented as Method 3 named search result optimization based on annotations (SROA).

Method 3: Search result optimization with annotations

Function SROA()

Step 1 Input: $D = ((R_1, A_1), (R_2, A_2), \ldots, (R_K, A_K))$;

Step 2 From $i = 1$ to K do

$Score_i = \alpha * P(Q|R_i) + \beta * P(Q|A_i)$;

Step 3 Output: $D' = ((R'_1, A'_1), (R'_2, A'_2), \ldots, (R'_K, A'_K))$;

End Function

In the method given above, we re-rank the initial search results according to the computed weighted score in decreasing order. The weighted score is calculated by the query-document similarity and the query-annotation similarity. So the higher the score, the higher the corresponding web page is in the optimized results.

4 Experimental results and discussions

4.1 Experimental setup

4.1.1 Datasets

In order to compare with some typical methods in the previous works, we don't use the benchmark TREC datasets. Because the baseline method in this paper also uses the dataset from http://del.icio.us and the TREC datasets do not have social annotations. Therefore, we have conducted experiments on the dataset crawled from the famous bookmarking web site http://del.icio.us. For instance, the social annotations and the corresponding titles of some web pages on http://del.icio.us are shown in Table 1.

As the table shows,social annotations can be viewed as a summary of the content of web page to some extent. However, some social annotations are sparse and insufficient for understanding the web page. So the datasets in our experiment need to be processed by us before implementing the proposed method. The main steps of precessing the datasets are the following.

(1) First, we crawl 474,586 triples of (user, link, tag) from http://del.icio.us.
(2) Because in the experiment we need to obtain all the social annotations of a corresponding link, the initial triples need to be processed. After processing, we obtain 136,492

Table 1 Shown of some social annotations and the corresponding titles of some web pages on http://del.icio.us

Title of web page	Social annotations of web page
Finale music composing	Learn, music, notation
Your music with spotify friends	Play lists, play music
Music	AM, MUSIC, JEN, I
Finale music composing and notation software	Education music Audio Media tv
Cryptomnesia ǀ Music ǀ NerdGazm	nuevo,el,volta,mars,rodriguez

triples of (user, link, tag), which consist of 136,492 different web page links. Also the social annotations of each link are collected and saved for the next processing.

(3) Because of the speed of this foreign web site, many of its web pages could not be crawled. At present, we have about 35,000 web pages in the preliminary test. Indeed we have obtained a total number of tags of about 190,000.

We adopt the strategy of the third-party toolkit Lucene to obtain the initial search results and the query-document similarity. This strategy relies on query-document similarity to obtain initial search results.

4.1.2 Query construction

In the experiment, because the social annotations are sparse and because the capacity of the dataset is insufficient at present, we have adopted constructed queries manually to evaluate the re-ranking algorithm. Moreover, some of the most popular social annotations on the web site http://del.icio.us are selected to construct the set of queries denoted by QSet: queries such as "*design*", "*programming*", "*blog*", "*music*". In this paper, we use the query set QSet including about 300 popular social annotations or their constructed phrases.

4.2 Measure the weights in re-ranking method

In this paper, we use the formula $Score = \alpha * P(Q|R) + \beta * P(Q|A)$ to compute the ranking score, where α and β are determined by a large of experiment results. Specifically, we use normalized discount cumulative gain (NDCG) score of optimized results to determine the optimal value of α and β by adjusting the weights α and β. Figure 1 shows the average NDCG score of optimized results with respect to the corresponding weights α and β.

From the Fig. 1, we can observe that because of $\alpha + \beta = 1$, which means that $\beta = 1 - \alpha$. Therefore, given a specific query set, the abscissa is equivalent to the value of (α,β), and the ordinate is the average NDCG score with respect to the corresponding optimized results. From the data in the Fig. 1, we can conclude that when $\alpha = 0.7$ and $\beta = 0.3$, the average NDCG score of optimized results is higher, which means that the quality of results is better.

In this paper, the weights α and β are both experience values, therefore the proposed method has some limitations, such as it depend on the data sets. In the future, we will be make use of machine learning method to determine the weights α and β adaptively.

4.3 Evaluation of proposed method

In order to measure the effectiveness of the proposed method that optimizing the ranking method, the evaluation metrics we used in the experiments are NDCG

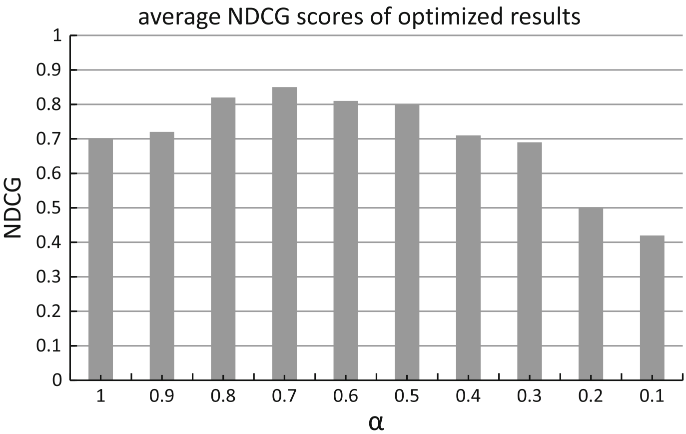

Fig. 1 The average NDCG scores of optimized results under different α weights

Table 2 Relevance of the two types of the top 6 web pages of search results provided by experiment participants

Web page	P_1	P_2	P_3	P_4	P_5	P_6
Results types						
Optimized results	3	2	1	0	3	3
Initial results	2	1	1	0	3	2

(Jarvelin and Kekalainen 2000) at K and Mean Average Precision (MAP) (Everett and Borgatti 2008). Quite a few works have adopted NDCG and MAP to measure web search ranking methods previously. NDCG and MAP are effective measures of web search engine ranking algorithms. Because the NDCG's gain is accumulated cumulatively from the top result to the bottom result, NDCG is suitable for measuring the graded relevance scale of search results. MAP is a measure that evaluates many queries. It is suitable for measuring the average precision of the queries in the QSet.

4.3.1 Measuring the proposed method with NDCG

NDCG is a measure of cumulated gain-based evaluation of information retrieval techniques. In this paper we use NDCG to estimate the cumulative relevance gain by examining the search results with given similarity scores. In reality, the query-document relevance is not of only two types, that is, relevant and irrelevant. Generally it often has a graded relevance , such as from 0 to 3. Indeed, we hope that the higher the relevance of the web page, the higher it will be in the search results. So we adopt NDCG to evaluate the effectiveness of the proposed ranking method. In accordance with the evaluation of NDCG, the higher the score computed by NDCG, the better the performance of the ranking method of the search engine.

Presented with a list of web pages that had already been the subject of queries, the experiment participants are asked to judge the relevance of each web page to the constructed queries mentioned above. Each web page is to be judged on a score of 0–3 with 0 meaning irrelevant, 3 meaning relevant completely, and 1 and 2 representing relevant to some extent. That is, the higher the score, the more relevant the web page to the query.

Now we show the NDCG's computing process by illustrating with respect to the query "J2EE". The top 6 web pages of the two types of search results are all denoted by P1, P2,

Table 3 Sub-process of IDCG score computation of the top 6 optimized results

i	rel_k	$\log_2 k$	$\frac{rel_k}{\log_2 k}$	$rel_1 + \sum_{k=2}^{i} \frac{rel_k}{\log_2 k}$
1	3	N	N	3
2	2	1	2	5
3	1	1.59	0.629	5.629
4	0	2.00	0	5.629
5	3	2.32	1.293	6.922
6	3	2.59	1.158	8.08

Table 4 IDCG' s scores of the two types of top 6 web pages

Type	$IDCG_1$	$IDCG_2$	$IDCG_3$	$IDCG_4$	$IDCG_5$	$IDCG_6$
Optimized results	3	6	7.887	8.887	9.138	9.138
Initial results	3	5	6.258	6.758	7.189	7.189

Table 5 NDCG' s scores of the two types of top 6 web pages

Type	$NDCG_1$	$NDCG_2$	$NDCG_3$	$NDCG_4$	$NDCG_5$	$NDCG_6$
Optimized results	1	0.883	0.714	0.633	0.757	0.884
Initial results	0.667	0.6	0.580	0.537	0.685	0.792

P3, P4, P5, P6, Table 2 shows the similarity scores of the top 6 web pages of the two types of search results provided by experiment participants.

The $NDCG_i = \frac{rel_1 + \sum_{k=2}^{i} \frac{rel_k}{\log_2 k}}{IDCG_i}$ (Yanbe et al. 2007), where rel_k denote the graded relevance of the result at position k and judged by experiment participants from 0 to 3. $IDCG_i$ is the ideal ordering of the scores of the corresponding results. For instance, the ideal ordering of the top 6 optimized results can be denoted by the vector (3, 3, 3, 2, 1, 0). Table 3 shows the sub-process of IDCG score computation of the top 6 optimized results. The top 6 optimized results scores of $rel_1 + \sum_{k=2}^{i} \frac{rel_k}{\log_2 k}$ are denoted by the vector = (3, 5, 5.629, 5.629, 6.922, 8.08).

Just as in the case of the computation of the top 6 optimized results given above, so scores were given to the top 6 initial results. The initial top 6 web page scores of $rel_1 + \sum_{k=2}^{i} \frac{rel_k}{\log_2 k}$ are denoted by the vector = (2, 3, 3.629, 3.629, 4.922, 5.694). In order to obtain the IDCG scores, we need to compute the IDCG scores until now. Then Table 4 shows the IDCG's scores of the two types of top 6 web pages.

So far, according to the scores in the tables above, we can calculate the NDCG's scores. Table 5 shows the NDCG's scores by illustrating the top six web pages of the two types of search results with respect to the query "J2EE".

From the NDCG computation, we can conclude that the higher the relevance of the web page in the search results, the higher the NDCG scores of the search results. In our experiment, the NDCG values of the top 6 optimized results are larger than the top 6 initial search results because the more relevant web pages in the optimized results are higher than the web pages in the initial search results.

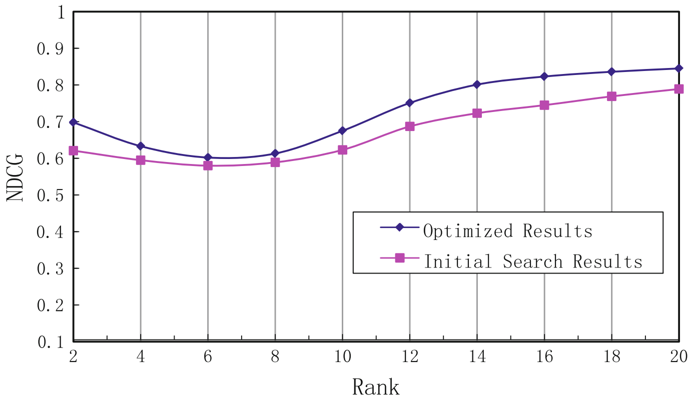

Fig. 2 NDCG scores comparison among two types of ranking results

From the NDCG computation to test the NDCG scores of the top 6 web pages for the two types of ranking results, we can select some key points as the test of the total 50 results for the two types of ranking results. The NDCG scores for all constructed queries can be averaged to obtain a measure of the average performance of a search engine's ranking method. So in the experiment for the query set QSet, we average the NDCG scores of 11 key points.

Figure 2 shows the NDCG scores comparison of the top 30 results on the QSet among the two types of ranking results. We can easily observe that the optimized results can further improve the relevance of the initial top 30 results. Through quantitatively comparing the NDCG scores of the top 30 optimized results and NDCD scores of the top 30 initial search results based on the query set QSet, we can obtain that the former can improve the NDCG score 21.1% than the latter through the average NDCG score on the 11 key points. For example, such as at the point 15, the NDCG score of the re-ranking results is 0.587 and the NDCG score of the initial search results is 0.506, the former improve the NDCG score by $(0.587 - 0.506)/0.506 = 16\%$. So we compute the improving percentage of NDCG scores based on the 11 key points, the improving percentage of the re-ranking results is 8.13%. Actually the improving percentage of NDCG score represents that the proposed ranking method can improve the query-document relevance weight by 8.13%. Because the initial search results only use query-document similarity. By analyzing the data from these tables showed above, we can conclude that the proposed method can optimize the search results.

4.3.2 Measuring the proposed method with MAP

In order to calculate the MAP of the search engine, we first need to compute the average precision denoted by AP for each query in QSet. We then compute the mean AP over all queries. High average precision produces the return of more relevant web pages earlier. The AP for each query is defined as $average\_precision = \sum_{j=1}^{M} p(j) * \Delta r(j)$, where $p(j)$ denotes the precision over the top j results, and $\Delta r(j)$ is the change in recall from $j - 1$ to j.

For instance, given two queries Q_1 and Q_2, the number of relevant results about query Q_1 is 4 and the number of relevant results about query Q_2 is 5, respectively. For the query Q_1, the order of the relevant results returned by the search engine is 1, 2, 4 and 7. For the query Q_2, the order of the relevant results returned by the search engine is 1, 3 and 5. So the AP of Q_1 is $(1/1+2/2+3/4+4/7)/4 = 0.83$. And the AP of Q_2 is $(1/1+2/3+3/5+0+0)/5 = 0.45$. Finally, the MAP of the two queries is $(0.83 + 0.45)/2 = 0.64$.

Table 6 Comparison of MAP among two types of the ranking results

Evaluate metrics	MAP
Types of results	
Optimized results	0.785
Initial results	0.735

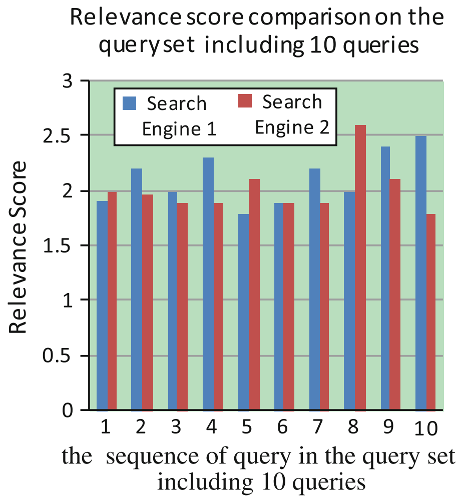

Fig. 3 Performance comparison between the two search engines's results

In accordance with the principle above, we can calculate the MAP of the two types of ranking results based on the QSet. Table 6 shows the comparison of MAP among the two types of the ranking results. The conclusion is that the optimized results can improve the mean average precision. Moreover, these results are closer than the initial results to the MAP of the standard results.

4.4 Comparison with existing method

There is no doubt that the method proposed, which re-ranks the initial search results according to the query-annotation similarity, performs better than the method that only use the query-document similarity. Nevertheless, it is necessary to have a comparison between the proposed method and the existing method SocialSimRank, used to find the latent semantic association between queries and annotations, which has been viewed as the baseline method in this paper. Here, we called the search engine adopted the proposed method in this paper as search engine 1 and called the search engine adopted the method SocialSimRank as search engine 2 in briefly. We select about 10 queries. Each query is submitted to the two search engines adopted the two methods respectively. The search results are compared between the two search engines accordingly. In summary, the overall performances of the two search engines differentiate little. We thus could not judge which one is better than the other. The mainly differences is that the search engine 1 outperforms the search engine 2 with regard to some popular annotations such as *java*, *design*, and *music*. However, with regard to some not frequently used annotations such as *dublin*, *fun*, and *php*, the situation is the opposite.

Figure 3 shows the average relevance score mentioned above which is judged by experiment participants manually through understanding the content of the web page, comparison of the top 30 results on the query set including 10 queries between the ranking results of the two search engines. Therefore these statistics can reflect the performance comparison of the two search engines to some extent.

4.5 Discussions

The results presented in the preceding section show the performance of the proposed ranking method in this paper. We have found that social annotations can actively benefit the web search through integrating query-annotation similarity into query-document similarity. We also found that using a language model to evaluate the query-annotation similarity is convenient (Zhai and Lafferty 2004). Because of the social annotations' own feature, that is, that the social annotation consists of a sequence of terms (Golder and Huberman 2006), we need not parse social annotations generally and we can construct a corpus easily. At present, combinations between social annotations and language models are becoming more and more effective with the steady increase of social annotations.

However, for further improving the performance of the proposed method in this paper, there are still several problems to further address. First, we must consider the sparsity of social annotations. We may be correspondingly propagating the annotations between web pages to solve this problem. Secondly, as the semantic similarity of social annotations, we can incorporate the semantic similarity to ranking method in the future work. Finally,In the future, we will be make use of machine learning method to determine the weights α and β adaptively.

5 Conclusions and future work

In this paper, we intend to optimize retrieval ranking method by using the ranking strategy of integrating the query-annotation similarity into query-document similarity. Specifically, we calculate the query-annotation similarity by using a language model. Then the initial search results are re-ranked according to the computational weighted score of the query-document similarity score and the query-annotation similarity score. An experiment on the QSet showed that the method proposed in this paper could significantly optimize the ranking method.

However, there are many problems that should be considered. In the future we plan to consider the following aspects. On the one hand, in the follow-up study, the annotation semantic similarities (Wu et al. 2006) will be integrated to improve calculating the relevance weight based on the language model. We hope that the annotation semantic similarities will benefit ranking method. On the other hand, we intend to adopt a third-party toolkit about machine learning to determine the influence on the final results of the relevance weight of re-ranking results and the initial search results respectively.

Meanwhile, because the social annotations were sparse and because the capacity of the experiment dataset was insufficient, thereby impacting the effectiveness of the proposed method in the paper, we will research the automatic tagging of the web pages. We will also make use of the ROM proposed by Zeng (2008) to achieve the automatic tagging for the web page.

However, the present paper is just a beginning study of query-annotation similarity. In the future, we would like to optimize the proposed method by integrating the annotation semantic similarities and by improving the language model itself.

Acknowledgments We would like to thank Dr. Yong Zeng at Concordia University for his suggestions on this paper. We also thank Yanan Jin for his help with the analysis of http://del.icio.us data. This paper is supported by National Natural Science Foundation of China under Grant No. 60873225, 60773191, 70771043, National High Technology Research and Development Program of China (863 Program) under Grant No. 2007AA01Z403, Natural Science Foundation of Hubei Province (NSF-HB) under grant No. 2009CDB298, Open Foundation of State Key Laboratory of Software Engineering under Grant No. SKLSE20080718 and Innovation Fund of Huazhong University of Science and Technology under Grant No. Q2009021.

References

Baeza-Yates R, Ribeiro-Neto B (2005) Modern information retrieval. China Machine Press, Beijing
Ball EC (2009) Annotation an effective device for student feedback: a critical review of the literature. Nurse Educ Pract 10(3):138–143
Bao S, Wu X, Fei B, Xue G, Su Z, Yu Y (2007) Optimizing web search using social annotations. In: WWW, pp 501–510
Cattuto C, Barrat A, Baldassarri A, Schehr G, Loreto V (2009) Collective dynamics of social annotation. PNAS 106(29):10511–10515
Chen G, Wang Y (2000) Research on fast algorithms for removing redundant strings in a string set. Mind 19(3):255–257
Ding G, Bai S, Wang B (2006) A survey of statistical language modeling for text retrieval. Chin Acad Sci 43(5):769–776
Everett MG, Borgatti SP (2008) Statistical language models for information retrieval: a critical review. Found Trends Inf Retr 2(3):137–213
Golder S, Huberman BA (2006) Usage patterns of collaborative tagging systems. J Inf Sci 32(2):198–208
Hammond T, Hannay T, Lund B, Scott J (2005) Social bookmarking tools. (i): A general review. D-Lib Mag 11(4). Available at: http://www.dlib.org/dlib/april05/hammond/04hammond.html
HeyMann P, Koutrika G, Garcia-Molina H (2008) Can social bookmarking improve web search. In: WSDM, pp 195–206
Hiemstra D (1998) A linguistically motivated probabilistic model of information retrieval. In: Research and advanced technology for digital libraries, pp 569–584
Hong L, Chi EH (2009) Annotate once, appear anywhere: collective foraging for snippets of interest using paragraph. In: Human factors in computing systems, pp 1791–1794
Jarvelin K, Kekalainen J (2000) Ir evaluation methods for retrieving highly relevant documents. In: SIGIR, pp 41–48
Jeffreys H (1948) Theory of probability. Clarendon Press, Oxford
Liu N, Yan J, Fan W, Yang Q, Chen Z (2009) Identifying vertical search intention of query through social tagging propagation. In: WWW, pp 1209–1210
Lv Y, Zhai C (2009) Positional language models for information retrieval. In: SIGIR, pp 299–306
Miller D, Leek T, Schwartz RM (1999) A hidden markov model information retrieval system. In: SIGIR, pp 214–221
Noll MG, Meinel C (2008) The metadata triumvirate: social annotation, anchor texts and search queries. In: Web intelligence (WI), pp 640–647
Ponts JM, Croft WB (1998) A language modeling approach to information retrieval. In: SIGIR, pp 275–281
Pratt JW (1976) F.Y. Edgeworth and R.A. Fisher on the efficiency of maximum likelihood estimation. Ann Stat 4(3):501–514
Robertson SE (1977) The probability ranking principle in IR. J Doc 33(4):294–304
Song F, Croft WB (2006) A general language model for infromation retrieval. In: WAIM, pp 97–108
Wu X, Zhang L, Yu Y (2006) Explore social annotations for the semantic web. In: WWW, pp 417–426
Xu S, Bao S, Cao Y, Yu Y (2007a) Using social annotations to improve language model for information retrieval. In: CIKM, pp 1003–1006
Xu S, Bao S, Yu Y, Cao Y (2007b) Using social annotations to smooth the language model for IR. In: PAKDD, pp 1015–1021
Yan J, Liu N, Chang EQ, Ji ZL (2009) Search result re-ranking based on gap between search queries and social tags. In: WWW, pp 1197–1198
Yanbe Y, Jatowt A, Nakamura S, Tanaka K (2007) Can social bookmarking enhance search in the web? In: JCDL, pp 107–116
Zeng Y (2008) Recursive object model(rom)-modeling of linguistic information in engineering design. Comput Ind 59(6):612–625

Zhai C, Lafferty J (2004) A study of smoothing methods for language models applied to information retrieval. In: TOIS, pp 179–214

Zhou D, Bian J, Zheng S, Zha H, Giles CL (2008) Exploring social annotations for infromation retrieval. In: WWW, pp 715–724