Personalized Web Search by Using Learned User Profiles in Re-ranking

Jia Hu and Philip K. Chan

Department of Computer Sciences, Florida Institute of Technology
Melbourne, FL 32901, USA
jhu@fit.edu, pkc@cs.fit.edu
WWW home page: http://www.cs.fit.edu/∼pkc/

Abstract. Search engines return results mainly based on the submitted query; however, the same query could be in different contexts because individual users have different interests. To improve the relevance of search results, we propose re-ranking results based on a learned user profile. In our previous work we introduced a scoring function for re-ranking search results based on a learned User Interest Hierarchy (UIH). Our results indicate that we can improve relevance at lower ranks, but not at the top 5 ranks. In this paper, we improve the scoring function by incorporating new term characteristics, image characteristics, and pivoted length normalization. Our experimental evaluation shows that the proposed approach can improve relevance in each of the top 10 ranks.

1 Introduction

Today's search engines usually cannot distinguish different users' needs well. For example, a computer scientist may use the search query "leopard" to locate information on Apple OS X Leopard and a biologist may use the same query for the animal leopard; however, a search engine usually treats the two queries the same way. Alternatively, personalized search provides customized results.

In our previous work we introduced a scoring function for personalizing search results [8]. The function uses four characteristics (the depth of a node where a term belongs to, the length of a term, the frequency of a term, and the emphasis of a term) to score a term that matches the user profile (called UIH), which is learned from the user's bookmarks. We use the page scores to re-rank retrieved web pages. Based on precision and recall, we showed that our personalized re-ranking approach outperformed a search engine at lower ranks, but not at the top 5 ranks. In this paper we improve the scoring function by modifying the features/characteristics used in the function and adding document length normalization. The main contributions of this paper are:

1. modifying the node depth characteristic to node specificity,
2. adding two new characteristics: term span and term specificity,
3. incorporating image term characteristics by extracting image terms from `img` tags,

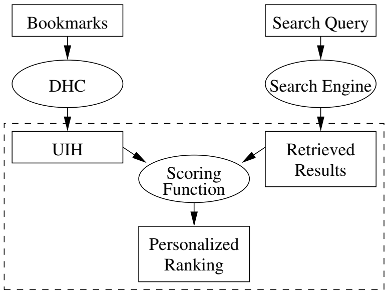


Fig. 1. Personalized Search with UIH

4. utilizing pivoted normalization to reduce bias due do document length, and
5. showing our proposed approach can outperform previous work and a search engine at all top 10 ranks, based on 11 users' 22 search queries.

We will discuss related work in Sec. 2. Our proposed re-ranking approach is detailed in Sec. 3 and evaluated in Sec. 4. Sec. 5 summarizes our findings.

2 Related Work

Jeh and Widom [5] proposed a personalized web search by modifying the global PageRank algorithm [2]. Instead of starting from random pages on the web, the "random surfer" starts from a set of preferred pages (such as bookmarks). For re-ranking results returned by a search engine, the scoring function in Compass Filter [9] favors results in web communities [10] that were also visited by the user. In [19] user profiles are built based on browsing history. Teevan et al. [20] investigated using files stored on the user computer as implicit relevant feedback. Using click-through data on search results, UCAIR [14, 15] extracts words from snippets of the clicked links for query expansion. Agichtein et al. [1] investigated click-through, browsing, and query-text features to construct user profiles and used different ranking methods re-rank the search results.

A few studies explored re-ranking by building user profiles that have different abstraction levels (general to specific), which can provide different contexts for personalization. Pretschner and Gauch [11] presented a system that allows for the automatic creation of structured user profiles, which are built based on an existing category hierarchy. Speretta and Gauch [18] proposed to build a user profile as a weighted concept hierarchy, which is created from the Open Directory Project (ODP). Sieg et al. [16] also used ODP to learn user profiles for personalized web search. ODP currently contains more than 590,000 concepts/nodes, so they only use a few top levels of categories in the ODP hierarchy. Hence, the user profiles do not cover the low-level categories, which are more specific. Consequently, this may reduce the ranking quality for individuals with more specific interests, not represented as high-level categories in ODP. Also, using an existing hierarchy can make the user profile contain many irrelevant categories

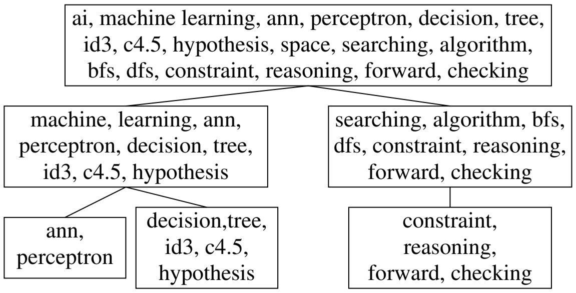

ai, machine learning, ann, perceptron, decision, tree, id3, c4.5, hypothesis, space, searching, algorithm, bfs, dfs, constraint, reasoning, forward, checking
machine, learning, ann, perceptron, decision, tree, id3, c4.5, hypothesis
searching, algorithm, bfs, dfs, constraint, reasoning, forward, checking
ann, perceptron
decision,tree, id3, c4.5, hypothesis
constraint, reasoning, forward, checking

Fig. 2. User Interest Hierarchy

since **all** high-level categories are in the user profiles. To avoid these disadvantages of using an existing taxonomy/hierarchy, Kim and Chan [6] proposed a method to construct a user interest hierarchy (UIH) by learning from implicit user behavior. We also proposed a scoring function [8] for personalized ranking with the UIH learned from bookmarks. We can score a page based on the user profile and the results returned by a search engine as shown in Fig. 1.

To build the user profile, called UIH, we used the web pages in user's bookmarks and the Divisive Hierarchy Clustering (DHC) algorithm [8]. As shown in Fig. 2 , a UIH organizes a user's interests from general to specific. Near the root of a UIH, general interests are represented by larger clusters of terms while towards the leaves, more specific interests are represented by smaller clusters of terms. The term refers to a phrase that has one or more words. The root node contains all distinct terms in the bookmarked web page. The leaf nodes contain more specific terms of interests to the user. The strength of relationship between terms is estimated based on their co-occurrence in the same web page. To re-rank pages in the search results, we proposed four characteristics to calculate the score for each term that matches the UIH. The experimental results show that our approach can be more accurate than Google below the top 5 ranks. In this paper we propose improved features in the scoring function, which can be more accurate than Google at all top 10 ranks.

3 Personalized Re-ranking

Given a web page from the search results and a UIH, we identify **matching terms** (words/phrases) that reside both in the web page and in the UIH. The scoring function for personalized re-ranking is based on characteristics of the matching terms.

3.1 Term Characteristics

Each **matching term** is analyzed according to four characteristics: term frequency, term span, term specificity, and node specificity.

Term Frequency More frequent terms are more significant than less frequent terms. A document that contains a matching term a number of times will be more related to a user's interest than a document that has the matching term only once. We estimate the probability, $P(F_{t_i})$, of a matching term t_i at frequency F_{t_i} in a web page to measure the significance of the term. In general, frequent terms have a lower probability of occurring. For example, in a web page most of the terms occur once, some terms happen twice, and fewer terms repeat three times or more. A term, t_i, is more significant when $P(F_{t_i})$ is lower. We estimate $P(F_{t_i})$ as:

$$P(F_{t_i}) = \frac{\text{number of distinct matching terms with frequency } F_{t_i} \text{ in a web page}}{\text{total number of matching terms in a web page}}. \tag{1}$$

Term Span Although a term with higher term frequency is more significant to a document, it may not be specific to the whole document if the term occurs only in certain part of the document. We consider a term is more relevant to a document if it appears in more diverse locations in the document. We estimate the probability, $P(S_{t_i})$, of a matching term t_i by measuring the span S_{t_i}, from the first occurring position to the last in a web page. When a term occurs only once, S_{t_i} is zero. In general, terms with a larger span are more significant and have lower $P(S_{t_i})$, which is estimated as:

$$P(S_{t_i}) = \frac{\text{number of distinct matching terms with span } S_{t_i} \text{ in a web page}}{\text{total number of matching terms in a web page}}. \tag{2}$$

Term Specificity A term which occurs in many documents is not a good discriminator, and has less significance than one which occurs in few documents—the same reasoning for Inverse Document Frequency (IDF). We measure term specificity by estimating the likelihood $P(E_{t_i})$ of the matching term, t_i, appearing in the documents:

$$P(E_{t_i}) = \frac{\text{number of web pages } E_{t_i} \text{ that contain the term } t_i}{\text{total number of returned web pages}}. \tag{3}$$

Node Specificity A UIH represents general interests in large clusters of terms near the root of the UIH, while more specific interests in smaller clusters of terms near the leaves. Terms in more specific interests are harder to match and a term matching a more specific node has more significance than matching a more general node. A term can appear in multiple nodes; we consider the most specific node that a term matches. We measure node specificity by estimating the likelihood $P(N_{t_i})$ of matching term t_i at node N_{t_i} in the UIH as:

$$P(N_{t_i}) = \frac{\text{number of distinct terms in node } N_{t_i}}{\text{total number of distinct terms in the UIH}}. \tag{4}$$

N_{t_i} is more specific when $P(N_{t_i})$ is lower.

3.2 Image Term Characteristics

As images can speak a thousand words, images can contribute to the significance of a web page. A meaningful image should be large enough to convey content. If an image is too small, it might be just an icon that has no relevance to the content. Hence, we only consider images that satisfy one of these two conditions:

1. Both the image width and height are larger than 50;
2. Either the image width or height is larger than 50, and there is no icon or arrow term included in the src, name or alt parameter.

From the `img` tags, we extract the image file name from the `src` parameter, and terms from the `alt` and `name` parameters. For example, from this `img` tag:

```
<img src="graphics/florida.gif" name="florida scene" alt="World
| United States | South | Florida" width="200" height="105" >
```

we extract: florida, scene, world, united, states, south. But from this `img` tag:

```
<img src="graphics/florida.gif" name="icon image" alt="World
| United States | South | Florida" width="40" height="50" >
```

we extract nothing since “icon” is in `name` and `width` is smaller than 50.

After extracting terms from all the qualified `img` tags, we filter these image terms by a stop list and a stemmer, then match these terms to the UIH. Each matching image term, g_i, is analyzed according to the same four characteristics we discussed in the previous section: term frequency (F_{g_i}), term span (S_{g_i}), term specificity (E_{g_i}), and node specificity (N_{g_i}).

3.3 Scoring a Web Page

Based on the term and image term characteristics, we calculate a score for each term. Let $P(F_{t_i}, S_{t_i}, E_{t_i}, N_{t_i})$ be the joint probability of all four characteristics for the matching term t_i. The four characteristics are generally independent, except for term frequency and term span (higher frequency could imply larger span). For simplicity, we assume the four characteristics are independent: $P(F_{t_i}, S_{t_i}, E_{t_i}, N_{t_i}) = P(F_{t_i}) \times P(S_{t_i}) \times P(E_{t_i}) \times P(N_{t_i})$ and calculate the negative $\log_2$ likelihood:

$$-\log_2 P(F_{t_i}, S_{t_i}, E_{t_i}, N_{t_i}) = -\log_2 P(F_{t_i}) - \log_2 P(S_{t_i}) - \log_2 P(E_{t_i}) - \log_2 P(N_{t_i}). \tag{5}$$

In information theory [12] $-\log_2 P(e)$ is the number of bits needed to encode or the amount of information in event e. Hence, Eq. 5 yields the amount of information in the four characteristics. We also consider that some characteristics are more important than the others. Term frequency F_{t_i}, term span S_{t_i}, and term specificity E_{t_i} represent the term relevance to a web page; however, node specificity N_{t_i} represents the term relevance to a user's interests. A simple heuristic used in this paper assumes N_{t_i} is twice as important as the other characteristics. Thus the weights $w_1 = 0.2$, $w_2 = 0.2$, $w_3 = 0.2$, and $w_4 = 0.4$ are assigned to:

$$ST_i = -w_1 \log_2 P(F_{t_i}) - w_2 \log_2 P(S_{t_i}) - w_3 \log_2 P(E_{t_i}) - w_4 \log_2 P(N_{t_i}). \quad (6)$$

Similarly, we calculate the image term score SG_i for image term g_i. Consider n is the number of terms and m is the number image terms that match the UIH, the score S_{p_j} for page p_j, combining the scores from term and image term characteristics, is:

$$S_{p_j} = \sum_{i=1}^{n} ST_i + \sum_{i=1}^{m} SG_i. \quad (7)$$

The user profile (UIH) contains user preference, but it does not know the importance of a document among all documents on the web. We use the rank order $R(Google_{p_j})$ returned by Google as our "public" score for page p_j. The combined personal and public page score (PPS) for page p_j is:

$$PPS_{p_j} = c \times R(S_{p_j}) + (1 - c) \times R(Google_{p_j}), \quad (8)$$

where R returns the reverse rank number such that a smaller rank number (higher rank) yields a higher score. The personal page score and the public page score are weighted by parameter c. According to [8], $c = 0.5$ shows the highest performance and is used in our experiments.

3.4 Document Length Normalization

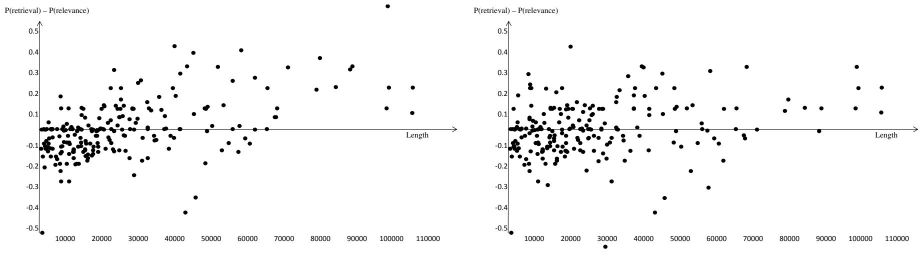
Since longer documents have more terms, they are likely to have more matching terms. Thus longer documents might have a bias of getting higher scores and are more likely to be retrieved. Cosine normalization [13] is commonly used to reduce the bias. The cosine normalization factor is computed as:

$$C_{ST} = \sqrt{ST_1^2 + ST_2^2 + ST_3^2 + \cdots + ST_n^2}. \quad (9)$$

Similarly, C_{SG} for image terms is calculated. Page score in Eq. 7 is adjusted to:

$$S_{p_j} = \sum_{i=1}^{n} ST_i / C_{ST} + \sum_{i=1}^{m} SG_i / C_{SG}. \quad (10)$$

Singhal et al. [17] showed that retrieval is more effective when a normalization strategy retrieves documents with probability similar to their probability of relevance. When the probability of retrieval is larger than the probability of relevance, some non-relevant documents can be retrieved and we ned to decrease the probability of retrieval. On the contrary, when the probability of retrieval is smaller than the probability of relevance, some relevant documents may not be retrieved and we need to increase the probability of retrieval. When the probability of retrieval is similar with the probability of relevance, all the relevant documents may be retrieved, which is the most effective.



(a) (b)

Fig. 3. $P(retrieval) - P(relevance)$ based on (a) term and (b) image term scores

We analyzed the data set from 22 searches collected in Section 4. For each search query, we ordered the top 100 retrieved web pages by their byte lengths and divide them into 10 equal sized bins and each bin contains 10 web pages, thus there are a total of 220 bins. Then by using cosine normalization, we calculate the total term scores and combine them with the public score to get a rank order for the 100 web pages (Eq. 8). We calculate total image term scores and combine them with the public score to get another rank order. After that we compute the probability of relevant/retrieved web pages belonging to a certain bin based on term scores and image term scores separately. The probability of relevant/retrieval, $P(relevance)/P(retrieval)$, are estimated as:

$$P(relevance) = \frac{\text{number of relevant web pages in a bin}}{\text{total number of relevant web pages in query results}} \tag{11}$$

$$P(retrieval) = \frac{\text{number of web pages at top 10 rank in a bin}}{10} \tag{12}$$

We plot $P(retrieval) - P(relevance)$ obtained from the 220 bins against the median web page byte length in each bin based on term scores in Fig. 3(a) and based on image term scores in Fig. 3(b). From Fig. 3(a), we found for web pages longer than about 70000 bytes, $P(retrieval)$ is higher than $P(relevance)$, and for the pages shorter than about 20000 bytes, $P(retrieval)$ is usually smaller than $P(relevance)$. That is, even after cosine normalization has been applied, longer web pages still have a bias to be ranked higher and shorter web pages to be ranked lower. From Fig. 3(b), we can also make a similar, but less prominent, observation for image terms.

Pivoted Normalization From Equation 10, we know that a higher normalization factor decreases the score. Thus the probability of retrieving a web page is inversely related to the normalization factor. Since we observe longer web pages have a bias to be ranked higher than shorter web pages with cosine normalization factor, we should **further** increase the normalization factor for longer web pages and decrease it for shorter web pages.

Singhal et al. [17] proposed pivoted normalization, which is based on cosine normalization. Their observation is **opposite** to ours: $P(retrieval)$ is larger, not

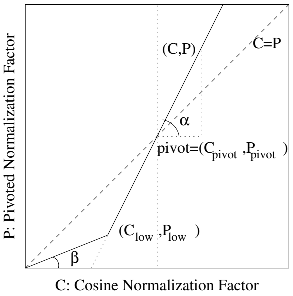

(C,P)
C=P
α
pivot=(C_{pivot} ,P_{pivot})
(C_{low} ,P_{low})
β
P: Pivoted Normalization Factor
C: Cosine Normalization Factor

Fig. 4. Pivoted Normalization

smaller, than $P(relevance)$ for shorter documents and $P(retrieval)$ is smaller, not larger, than $P(relevance)$ for longer documents. Thus when the cosine normalization factor is less than a "pivot," they increase the pivoted normalization factor to decrease $P(retrieval)$ for shorter documents, otherwise they decrease the pivoted normalization factor to increase $P(retrieval)$ for longer documents. However, in our case, we need to decrease the pivoted normalization factor for shorter documents and increase it for longer documents. We illustrate the relationship between our pivoted normalization P (x-axis) and cosine normalization C (y-axis) as a solid line in Fig. 4. The amount of tilting of the solid line at the pivot away from the identity ($C = P$) dotted line is the *slope*, which is a parameter. Since $\mathit{slope} = \tan\alpha = (P - P_{pivot})/(C - C_{pivot})$ and $C_{pivot} = P_{pivot}$,

$$P = C_{pivot} + \mathit{slope} \times (C - C_{pivot}). \tag{13}$$

P is smaller than C on the left side of pivot, and P becomes negative when C is smaller than the x-intercept. In order to avoid a negative value for P, we draw an additional line from the origin to a point (C_{low}, P_{low}) on the solid line:

$$P = \tan\beta \times C = P_{low}/C_{low} \times C, \tag{14}$$

where P_{low} is the smallest **positive** pivoted normalization factor for a search query and C_{low} is the corresponding cosine normalization factor according to Eq 13: $(P_{low} - C_{pivot})/\mathit{slope} + C_{pivot}$. The revised pivoted normalization is:

$$P = \begin{cases} C_{pivot} + \mathit{slope} \times (C - C_{pivot}) & \text{if } C \geq C_{low} \\ P_{low}/C_{low} \times C & \text{if } 0 < C < C_{low}. \end{cases} \tag{15}$$

Consider P_{ST} is the pivoted normalization factor for terms and P_{SG} for image terms, page score in Eq. 7 becomes:

$$S_{p_j} = \sum_{i=1}^{n} ST_i/P_{ST} + \sum_{i=1}^{m} SG_i/P_{SG}. \tag{16}$$

Similar to [17], we choose the average cosine normalization factor as the pivot. For the data set we collected, we found $\mathit{slope} = 1.2$ is the best value for term scores in Equation 15 and $\mathit{slope} = 1.1$ for image term scores. The difference is consistent with the trend being steeper in Fig. 3(a) for term scores than Fig. 3(b) for image term scores.

4 Experimental Evaluation

To measure the ranking quality, we use Discounted Cumulative Gain (DCG) [4]. DCG is a measurement that gives more weight to higher ranked documents by discounting the gain values $G(r)$, where r is the rank, for lower ranked documents. Also, the gain values can be of different magnitude for different relevance levels. $DCG(r)$ is defined as:

$$DCG(r) = \begin{cases} G(1) & \text{if } r = 1 \\ DCG(r-1) + G(r)/\log(r) & \text{otherwise} \end{cases} \tag{17}$$

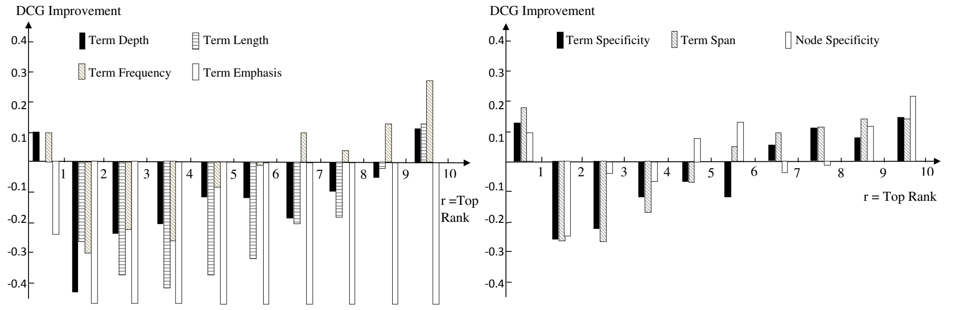
In our experiments, we used $G(r) = 1$ for non-relevant results, $G(r) = 2$ for relevant results, and $G(r) = 3$ for highly relevant results.

In our experiments we used the same data set in our previous work [8], the data were collected from 11 different users and each user submitted 2 search queries that can contain any Boolean operators. Some examples of the search queries used are: *review forum +"scratch remover", cpu benchmark, aeronautical, Free cross-stitch scenic patterns, neural networks tutorial, DMC(digital media center), artificial intelligence* , etc. Then for each search query we used the top 100 web pages returned by Google. So there are 2200 web pages used in our evaluation. To evaluate the ranking quality, we asked each user to submit relevance ratings for the retrieved web pages. The relevance rating is divided to three scales: highly relevant, relevant and not relevant. For UIHs, the profile data are bookmarks from the 11 users and an UIH is learned for each user using the DHC algorithm [6]. Web pages from both Google and bookmarks were parsed to retrieve only words, which are stemmed and filtered through a stop list [3]. A phrase-finding algorithm [7] was used to identify variable-length phrases. Words in selection boxes/menus were also removed because they did not appear on the screen until a user clicked on them. Irrelevant information such as comments and styles were also removed. To remove any negative bias to Google, broken links that were still ranked high erroneously by Google were excluded from the evaluation, since those web pages are non-relevant to the user for sure. Visual Basic and Java were used for implementation, and the program ran on an Intel Pentium 4 CPU with 1.5GB memory.

4.1 Previous and Proposed Individual Term Characteristics

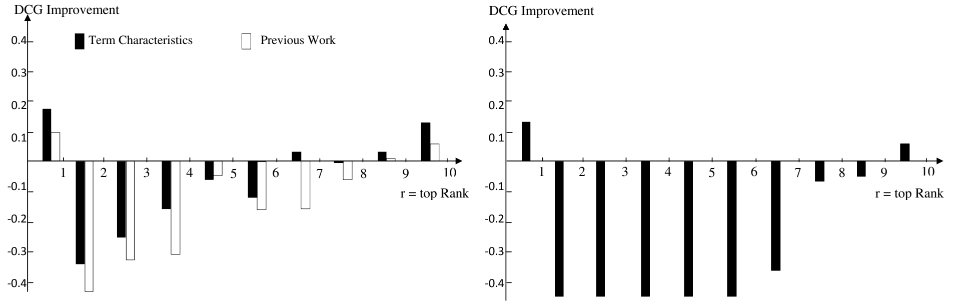
In our previous work [8], we used four characteristics for a term: term frequency, term length, term emphasis, and node depth. We analyze the top 10 DCG scores for each of these four characteristics. The x-axis in Fig. 5(a) is top rank r and the y-axis is the average DCG improvement of personalized ranking over Google ($DCG_{personalization}(r) - DCG_{Google}(r)$). Our personalized ranking can outperform Google at top 1 and top 10 ranks based on term depth, at top 10 rank based on the term length, at top 1, top 7, top 8, top 9 and top 10 ranks based on term frequency, and none based on the term emphasis.

Since the performance of term depth, term length and term emphasis are poor, we introduced term specificity and term span, and modified node depth to



(a) (b)

Fig. 5. DCG improvement based on Previous (a) and Proposed (b) Individual Term Characteristics



(a) (b)

Fig. 6. DCG Improvement based on Term (a) and Image Term (b) Characteristics

node specificity as described in Section 3.1. The experimental results from the three new characteristics are shown in Fig. 5(b). Our approach can outperform Google at top 1, top 7, top 8, top 9 and top 10 ranks based on term specificity, at top 1, top 6, top 7, top 8, top 9 and top 10 ranks based on term span, at top 1, top 5, top 6, top 9 and top 10 ranks based on node specificity, each of these three new characteristics can outperform Google at least half of the 10 top ranks. Including the original tem frequency, we use four characteristics in our scoring function. In the following sections, all the evaluation results are based on these four characteristics.

4.2 Combining Term Characteristics

We next combine these four characteristics to score each page and compare the results with our previous approach and Google in Fig. 6(a). We can see our personalized ranking based on term characteristics can outperform Google at top 1, top 7, top 9 and top 10 ranks. In most top ranks it performs better than our previous work, which can only outperform Google at top 1, top 9 and top

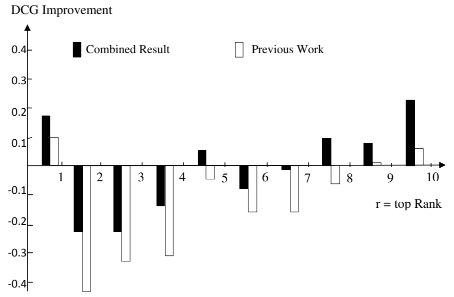


Fig. 7. DCG Improvement based on Combining Term and Image Term Characteristics

10 ranks. However the result is not ideal since we can only outperform Google at 4 top ranks out of 10.

4.3 Combining Image Term Characteristics

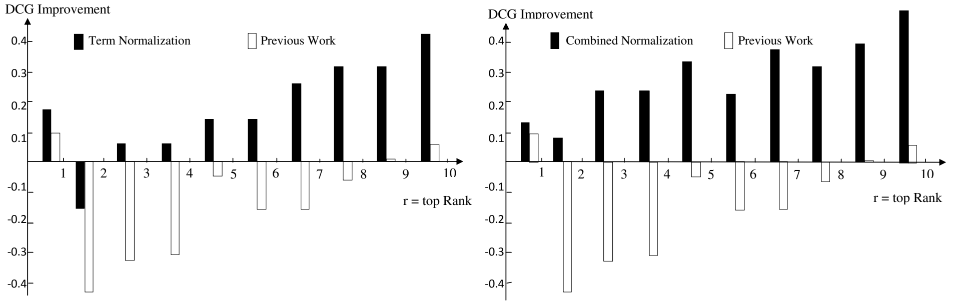
In this section we evaluate the combination of image term characteristics. The average DCG scores at top ranks are illustrated in Fig. 6(b). Our personalized ranking quality based on image terms can only outperform Google at top 1 and top 10 ranks. The results are even worse than the results based on term scores. This is reasonable since images generally provide less information than terms. So we combine these two sources of information to improve the ranking quality in the next section.

4.4 Combining Term and Image Term Characteristics

In Equation 7 the personalized score is based on both term scores and image term scores. After combining with Google rank in Equation 8, we re-rank each search's results, and compare the ranking quality with Google and our previous work. The average DCG scores at top 10 ranks are illustrated in Fig. 7. Our personalized ranking quality based on the combination of term and image characteristics can outperform Google at top 1, top 5, top 8, top 9 and top 10 ranks (half of the 10 top ranks). And it performs better than our previous work at all 10 top ranks. This shows using the combined score is better than using only term scores or image term scores.

4.5 Document Length Normalization

We have found term information is more robust for personalized scoring than image information, and the combination of them produces a better result. But in certain top ranks Google still performs better. This is because longer pages have a bias to obtain higher scores than shorter pages, so the chance for relevant short pages to be ranked high is reduced. In order to remove this bias we utilized



(a) (b)

Fig. 8. DCG Score based on Term Characteristics (a) and Combining Term and Image Characteristics (b) with Pivoted Normalization

pivoted normalization (Equation 16). Fig. 8(a) shows the average DCG results based on only term scores with pivoted normalization. Our personalized ranking can outperform Google at almost all top ranks except top 2, and it performs better than our previous work at all 10 top ranks.

We also combined the term score and image score with pivoted normalization, the results are shown in Fig. 8(b). Our personalized ranking can outperform Google and previous work at all 10 top ranks. Moreover, the improvement over Google increases at 8 of the 10 top ranks. In summary, our results indicate the significance of incorporating image term characteristics and pivoted normalization for improving personalized re-ranking.

4.6 Analysis of Search Queries and Bookmarks

We also investigated which search queries yielded higher DCG score with personalized search than with Google. Out of the 22 search queries (11 users × 2 search queries), our approach outperformed Google in 8 search queries (36%) at all 10 top ranks and partially (over half of the 10 top ranks) outperformed Google in 5 queries (23%). Google outperformed completely in 5 search queries (23%) and partially in 4 queries (18%). The search queries that our approach outperformed completely are: *aeronautical, Caribbean History, Free cross-stitch scenic patterns, XML Repository, ddr2 memory, Australia adventure tours. Australia ecology, java design patters*, and partially are: *boston pics, complex variables, beos operating system, artificial intelligence, sniper rifle.* The queries that Google outperformed completely are: *aerospace, cpu benchmark, review forum +"scratch remover", windows xp +theme +skin, neural networks tutorial*, and partially are: *DMC (digital media center), military weapons, extreme programming principles, woodworking tutorial.*

For the search queries that our algorithm did not outperform Google, we analyzed the search results and found that relevant web pages in the search results are few. For example, when a user searched *review forum +"scratch remover"*, there were only 4 highly relevant web pages rated by the user out of 100 search results. Hence, improving the ranking quality for this search is quite difficult.

We also analyzed the bookmarks, which are used for learning the user profiles. When we compare the bookmarks with the highly relevant retrieved web pages, we found that they are unrelated. For example, a user used *woodworking tutorial* as a search query, but he never bookmarked web pages related to that query. That is, additional sources of interests beyond bookmarks would be useful for building user profiles. However, naturally, the user can always perform a search that is different from any source of interests that can be collected.

5 Conclusion

This paper improves our previous work on personalized ranking by enhancing the accuracy of scoring function. We eliminated two term characteristics, term length and term emphasis, from the previous scoring function because we found they made little contribution to the rank quality. We also modified the node depth characteristic to the node specificity characteristic, which is more effective. Also we proposed two additional term characteristics, term term specificity and term span, which we found are very useful to score a term. Consequently, the four characteristics used in our new scoring function are: term frequency, term span, term specificity, and node specificity.

After re-ranking the search results by our proposed scoring function, we evaluated the performance by comparing with Google search and our previous work. Our previous work showed it could not perform better than Google at top 5 rank. By calculating average DCG scores from a collected data set, we found the improved personalized search based on term score without pivoted normalization factor can outperform Google at 4 top ranks out of 10, and can outperform our previous work at 9 top ranks out of 10. While combining term score and image score without pivoted normalization factor as our personalized search score the result was better, it can outperform Google at 5 top ranks out of 10, and can outperform our previous work at all 10 top ranks. After we added pivoted normalization factor into the scoring function to normalize the document length, our approach can outperform Google at 9 top ranks out of 10, and can outperform our previous work at all 10 top ranks. Our personalized search based on combination of term score and image score with pivoted normalization factor can outperform both Google and our previous work at all 10 top ranks. So the new term characteristics, extracted image terms and pivoted normalization help to improve our personalized ranking.

Although the new scoring function performed well on average for the 22 search queries, our algorithm did not outperform Google for some queries. We found some search queries were too specific that the relevant retrieved results were very few. This makes the scoring function difficult to improve the ranking quality of these search queries. We also found some search queries are not related to the user's bookmarks. Hence, improving ranking quality with only information from bookmarks is not sufficient. Our future work may capture user's recent interested web pages by implicit indicators like mouse movement, mouse click etc, and use these recent interested web pages to construct a short-term UIH to improve our personalized ranking.

References

1. E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. 29th SIGIR*, pages 19–26, 2006.
2. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. Intl. Conf. WWW*, 1998.
3. W. Frakes and R. Baeza-Yates, editors. *Information retrieval: data structures and algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1992.
4. K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proc. SIGIR*, pages 41–48, 2000.
5. G. Jeh and J. Widom. Scaling personalized web search. In *Proc. Intl. Conf. WWW*, pages 271–279, 2003.
6. H. Kim and P. Chan. Learning implicit user interest hierarchy for context in personalization. In *Proc. Intl. Conf. on Intelligent User Interfaces*, pages 101–108, 2003.
7. H. Kim and P. Chan. Identifying variable-length meaningful phrases with correlation functions. In *Proc. IEEE Intl. Conf. on Tools with AI*, pages 30–38, 2004.
8. H. Kim and P. Chan. Personalized ranking of search results with learned user interest hierarchies from bookmarks. In O. Nasraoui, O. Zaine, M. Spiliopolou, B. Mobasher, B. Masand, and P. Yu, editors, *Web Mining and Web Usage Analysis*, pages 158–176. Springer, 2006.
9. A. Kritikopoulos and M. Sideri. The Compass Filter: Search engine result personalization using web communities. In *IJCAI Workshop on Intelligent Techniques for Web Personalization*, 2003.
10. R Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proc. Intl. Conf. WWW*, 1999.
11. A. Pretschner and S. Gauch. Ontology based personalized search. In *Proc. 11th Intl Conf. on Tools with AI*, pages 391–398, 1999.
12. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
13. G. Salton. *Automatic Text Processing*. Addison-Wesley, Reading, MA, 1988.
14. X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proc. SIGIR*, pages 43–50, 2005.
15. X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proc. CIKM*, pages 824–831, 2005.
16. A. Sieg, B. Mobasher, and R. Burke. A large-scale evaluation and analysis of personalized search strategies. In *Proc. CIKM*, pages 525–534, 2007.
17. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proc. SIGIR*, pages 21–29, 1996.
18. M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proc. Intl. Conf. Web Intelligence*, pages 622–628, 2005.
19. K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. Intl. Conf. on WWW*, pages 675–684, 2004.
20. J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. SIGIR*, pages 449–456, 2005.