Exploiting Semantic Query Context to Improve Search Ranking

Ziming Zhuang
Pennsylvania State University
rickzhuang@psu.edu

Silviu Cucerzan
Microsoft Research
silviu@microsoft.com

Abstract

One challenge for relevance ranking in Web search is underspecified queries. For such queries, top-ranked documents may contain information irrelevant to the search goal of the user; some newly-created relevant documents are ranked lower due to their freshness and to the large number of existing documents that match the queries. To improve the relevance ranking for underspecified queries requires better understanding of users' search goals. By analyzing the semantic query context extracted from the query logs, we propose Q-Rank to effectively improve the ranking of search results for a given query. Experiments show that Q-Rank outperforms the current ranking system of a large-scale commercial Web search engine, improving the relevance ranking for 82% of the queries with an average increase of 8.99% in terms of discounted cumulative gains. Because Q-Rank is independent of the underlying ranking algorithm, it can be integrated with existing search engines.

1. Introduction

Users are increasingly relying on search engines to discover relevant information [4, 35]. During their interactions with the search engine, users typically look at only the first few pages of search results [13]. Thus, from a user's perspective, relevance ranking is a critical factor to gauge the quality of a search engine.

Underspecified queries pose a challenge for relevance ranking. What is an underspecified search query? Consider the following two scenarios:

- **Naïve Queries:** A user submits the query *"hard disk case"* despite the fact that a *more accurate* description of the user's intent (i.e. generally accepted and more frequently used on the web) is *"hard drive enclosure"*. Because search engines usually rank the webpages based on their syntactic match with the query terms (i.e. considering the term frequency, proximity, etc.), the search results for this query could suffer in terms of relevance, even though some of the retrieved pages may actually contain the more accurate descriptive terms (such as *"drive"* and *"enclosure"*).
- **Query Recency:** Before the submission deadline, a user wants to find information about the IEEE ICDM 2008 conference by querying *"ICDM"*. Although we hypothesize that at that time a considerable amount of queries containing both *"ICDM"* and "2008", as well as other terms on the ICDM 2008 site (e.g. "*Call for Papers*"), have been repeatedly submitted to the search engines, these search engines may not retrieve the official ICDM 2008 website in the top-ranked results for the underspecified query *"ICDM"*[1] without explicitly specifying the year "2008".

In the first scenario, an underspecified query is an unarticulated query consists of naive search terms. In the second scenario, an underspecified query is a recency query (i.e. user implicitly favors more recent information). Both cases present a challenge to relevance ranking, and call for relevance ranking methods that take into account not only the overall quality of a webpage and its relevance to the query, but also the match with the users' information need, further referred to as their *real search intents* [4]. Because the query logs of large-scale search engines record the queries issued by a huge number of users, they are believed to be the implicit, collective endorsements about *what typical users are looking for* in a specific time frame. In this paper, we propose a novel method, *Q-Rank*, to leverage the implicit feedbacks from the logs about the users' search intents and to improve the relevance ranking.

The rest of this paper is organized as follows. In Section 2 we briefly review the existing body of work in ranking and search result refinement. In Section 3 we present the rationale, algorithm, and implementation of *Q-Rank*. In Section 4 we describe in detail the experiments for fine-tuning the parameters and the evaluation results. In Section 5 we discuss future work on several interesting scenarios in the framework of employing query log data in ranking. We conclude our paper in Section 6.

2. Related Work

There is a large body of work that investigates methods to rank webpages globally or dependent on a target query. PageRank [26] and HITS (Hypertext Induced Topic Search) [20] are two well-established ranking metrics that make use of the link structure of the Web. Both of them build upon the assumption that the quality of a webpage can be inferred by the number and the quality of pages linking to it. PageRank computes a global ranking score for *all* the pages on the Web independent of user queries and does not take into account the particular topics in which the search engine users are interested. HITS, on the other hand, works on a query-specific subgraph of the Web, so that the ranking scores are biased by the issued query. Hilltop [2] works mostly on popular topics and depends on a set of *expert pages* that are identified to be authoritative in the query domain to rank the other pages. Topic-sensitive PageRank [12] pre-computes a vector for a specific topic, and then uses these topic-sensitive vectors at query-time to bias the final ranking towards the particular topic(s)

[1] For example, when queried for *"ICDM"*, Google returns the official 2008 conference website at position 9; MSN Live Search returns it at position 14; Yahoo returns it at position 17 (as of 3/4/2008).

978-0-7695-3279-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSC.2008.8

denoted by the issued query. In recent years, there are also developments that try to learn the ranking preferences of the users [5, 10, 30].

Query expansion is an effective method to bridge the gap between users' internal information needs [4] and external query expressions. Thesauri-based query expansion [8, 27] generally relies on statistics to find certain correlations between query terms. Pseudo-relevance-feedback query expansion [24] uses the initially retrieved documents as a source for relevance feedback. Cui et al. [7] mine click-through records of search results to establish mappings from query terms to strongly correlated document terms which are then used for query expansion. Kraft and Zien [18] investigate a method to generate lists of weighted expansions for queries from the anchor texts of the retrieved documents. Billerbeck et al. [3] propose an effective method of obtaining query expansion terms from the past queries that retrieved the documents in the collection associated with a target query, reporting 26%-29% relative improvements over unexpanded retrieval on the TREC-9 and TREC-10 collections.

Search results can also be refined in an interactive manner and fulfilled in an iteration of search-and-feedback cycles [29]. Explicit feedback methods require users to make explicit judgments of the documents' relevance, and are easier to spam. In contrast, implicit feedback can be collected unobtrusively by analyzing users' search and click-through patterns [11, 14], and general user browsing actions [1]. Search results can also be refined by employing collaborative filtering algorithms to take into account similar users' preferences [33].

Recently, there have been developments that exploit query logs for search results refinement. Previous queries are selected based on the similarity between their search results and those retrieved by the target query [25], and are used to complement the current query in estimating document relevance [17, 31]. A formal user model is proposed based on the immediate search context for personalized ranking [32].

Our contribution is a novel re-ranking algorithm that uses distributional information about the query context, as extracted from search engine logs, which effectively improves the ranking of search results. Although our approach shares certain common grounds with existing studies, there are significant differences between the proposed method and previous approaches. Additionally, we have evaluated the performance of our method on a dataset that contains real-world editorial judgments and is much larger than those used in most existing studies, and carried out a series of comprehensive experiments to select the best parameters.

3. The *Q-Rank* Method

3.1. Rationale of *Q-Rank*

Q-Rank is based on a straight-forward yet very effective rationale, that the most frequently seen *query extensions* of a target query (terms extracted from queries that contain the target query as an affix) and *adjacent queries* (queries that immediately precede or follow a query in a user search session) provide important hints about users' search intents[2]. For example, for the target query "*aquarium*", the most frequent extensions, as observed in a real search engine query log, are "*fish*", "*supplies*", "*screensaver*", "*stands*", and "*plants*", the queries that most frequently follow it are "*aquarium screen saver*", "*aquariums*", "*aquarium supplies*", "*fish tanks*", "*aquarium fish*", "*aquarium screensaver*", and "*tropical fish*", while the most frequent queries that precede it (ignoring misspellings) are "*aquariums*", "*fish*", "*fish tank*", "*zoo*", "*petco*", "*aquariophilie*", "*aquarium screensaver*", and "*marine aquarium*". Intuitively, the distribution over query extensions in search engine query logs at any point in time can be regarded as a snapshot of the typical user interests related to the concept in a target query; thus, when a user submits the target query, it can be assumed that she may be interested in a collection of documents that closely match this distribution. Previous studies [21, 22] have shown that when a user is not satisfied with the current set of search results, the user is very likely to refine or rewrite the query, for the purpose of generalization, specialization, or adding new information, as illustrated in Scenario A in Figure 1. The correlation between a query and another one that frequently follows it in user search sessions can thus be regarded as an important clue about how users try to adjust their lexical choices to better match their information need to the Web content and the Web search process. Hence, both queries that frequently follow and queries that frequently precede a target query contain useful lexical information about documents that would satisfy the user's information need (the former capture better the attempt to disambiguate or match the Web content, the latter capture better the original user intent, especially when the users subtract terms from their original queries).

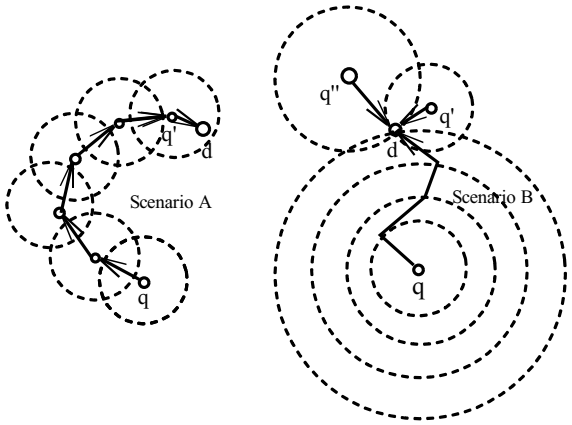


Figure 1. User behavior in the search space, in which a target query *q* is followed by a sequence of actions that eventually lead to a relevant document *d*. *Scenario A*: the user continuously reformulates the query until reaching a relevant document. Such reformulations (made by previous users) are exploited by *Q-Rank* to better compute the match between documents retrieved by a search engine for the target query *q* and the typical intent(s) of the users who have submitted *q*. *Scenario B*: circles represent different tiers in the ranked list; the user keeps on browsing the search result pages until finding a relevant document. This document may have been retrieved by various other queries related to the initial query and/or contain terms from such queries.

3.2. Extracting the query context

We formalize the definition of query context extracted from the query logs as follows. Let Q denote the set of queries in a search engine query log for a given time frame. We use the notation $Q_{next}(q)$ for the set of queries that were seen immediately

[2] Formal definitions of query extensions and adjacent queries are given in Section 3.2.

following a query q in user search sessions, and $Q_{prev}(q)$ for queries that were seen immediately preceding it. The union of these two sets will be referred to as *adjacent queries* $Q_{adj}(q)$:

$$Q_{adj}(q) \doteq \{q_{adj} \mid q_{adj} \in Q_{next}(q) \cup Q_{prev}(q)\}$$

The *user-based expansions* of a query q are all logged queries that contain q as an affix. In this work, we employ only expansions in which q is a prefix for efficiency reasons, and we define the *query extensions* of query q as being all such expansions from which the common prefix q is removed. Formally,

$$Q_{ext}(q) \doteq \{q_{ext} \mid q." \ ".q_{ext} \in Q\},$$

where " " is an empty space and "." denotes the operation of concatenating strings. In what follows, we will omit the argument q from the notation whenever no ambiguity results by doing so.

When computing Q_{ext} and Q_{adj}, we also employ some pre-processing steps to normalize the queries and aggregate statistics over *near duplicates*: basic stemming, punctuation removal, word order normalization, and spell checking [6].

In practice, when $Q_{ext}(q)$ is empty, we use instead $Q_{ext}(q_-)$, where q_- is the longest prefix query of q for which there exist two or more query extensions, but not more than a predefined small number P of extensions (in this way, backing-off from queries such as "*john malkovich*" to very general prefixes such as "*john*" is avoided). Considering again the "*hard disk case*" example, we retrieve adjacent queries such as "*portable hard disk*" and "*usb external hard drive*" from user search sessions. By backing off to the query "*hard disk*" because of the lack of extensions for the target query, we identify the popular extensions "*drive*", "*data recovery*", "*repair*", "*utilities*", "*failure*", "*problems*", "*eraser*", "*enclosure*", etc. The words in these adjacent queries and extensions will be used to re-rank the search results retrieved for the original query, under the assumption that such additional information can help to better represent the typical user's real search intent, in concordance with previous findings.

3.3. Calculating the re-ranking scores

Let q denote the original query issued by a user, referred to as the *target query*, and $D(q)$ (D when no ambiguity arises from omitting the query) denote a set of candidate documents for ranking. Here, we assume that D contains the n top-ranked documents returned by a search engine for the target query, but other scenarios for obtaining the set D can be envisioned. We assign a ranking score for each document d in D based on its lexical overlap with a set of most popular query extensions and adjacent queries to q, as in Definition 1. The numerator of this formula has two terms, which correspond to query extensions and adjacent queries, respectively. The tf·idf scores are assigned to each query-document pair. They are then weighted by the natural logarithm of the normalized query log frequency in order to account for the difference in query popularity. Each of the terms is weighted by the dampen factor, summed, and then divided by the initial rank of the document. This latter step is done to account for the initial ranking calculated by the search engine, which makes use of many features that are not available at time of re-ranking, such as the static rank of a document, the anchor text of links that point to the document, etc.

The impact of biasing the re-ranking process towards the initial ranking of the search engine is evaluated in the development experiments (Section 4.3). As expected (because of the extra ranking features used by the search engines), we find out that *Q-Rank* works better with such a bias.

$$RS(d,q) \doteq \frac{\gamma \cdot \sum_{i=1}^{|Q_{ext}|} tf(q_i,d) \cdot \ln\frac{|D|}{|D_{q_i}|} \cdot \ln\frac{qf(q_i)}{\sum_{j=1}^{|Q_{ext}|} qf(q_j)}}{R(d)} + \frac{(1-\gamma) \cdot \sum_{i=1}^{|Q_{adj}|} tf(q_i,d) \cdot \ln\frac{|D|}{|D_{q_i}|} \cdot \ln\frac{qf(q_i)}{\sum_{j=1}^{|Q_{adj}|} qf(q_j)}}{R(d)}$$

Definition 1. ***Q-Rank* re-ranking score. Q_{ext} and Q_{adj} denote the query context sets: extensions and adjacent queries to q, respectively. $\gamma \in [0, 1]$ denotes a dampen factor leveraging the contribution of each type of query context. $tf(q_i, d)$ denotes the frequency of the query context q_j in document d. D_{qi} contains all documents d for which $tf(q_i, d) > 0$. $qf(q_j)$ denotes the logged frequency of query q_j. $R(d)$ denotes the initial rank of d.**

3.4. The *Q-Rank* algorithm

Algorithm

Input: Initial query q,
Query log Q,
Options op, c, n, and u
Ranked document set $D=\{d_1,\ldots,d_{c'}\}$ retrieved for q, where $c' \leq c$

Output: Re-ranked document set $(d_{\pi(1)},\ldots, d_{\pi(c')})$, where π is a permutation of $1,\ldots,c'$

Construct $Q_{ext}(q)$ and $Q_{adj}(q)$ from Q, given q and op

foreach document $d \in D$ // calculate the re-rank scores
  $score_d \leftarrow RS(d, q)$
if $u = 0$
  Sort (descending) the documents d by $score_d$
else // the top u documents are not re-ranked
  Sort (descending) the documents $d_{u+1}, d_{u+2}, \ldots, d_{c'}$ by $score_d$
endif
return $(d_{\pi(1)},\ldots, d_{\pi(c')})$

The *Q-Rank* Algorithm

In the Q-Rank algorithm, op specifies whether Q_{ext} and/or Q_{adj} are used to construct the query context, as well as the size of these sets ($|Q_{ext}|$ and $|Q_{adj}|$); c specifies how many document candidates to consider and n specifies the output ranking range (obviously, $n \leq |D| \leq c$); u is the number of top-ranked search results to be left unchanged; i.e. if u is larger than zero, only the bottom n-u of the top n results will be re-ranked.

In the current implementation of *Q-Rank*, D consists of search-engine-generated snippets. An arguably better strategy is to use the full-text of the Web pages retrieved. However, the latter approach requires access to the full-text of these pages and a considerably larger computational effort and thus, it is not easily applicable in practice. We discuss this in detail in Section 4.

Collapsing [9] is a common practice for major Web search engines. While it provides navigational context for users, it can also mislead *Q-Rank* to incorrectly interpret the initial ranking of

the retrieved results. To minimize such negative impacts, we removed the duplicate websites in the *Q-Rank* experiments. That is, if two or more websites with the same domain (top level URL) are found in *D*, only the one with the highest rank is retained in the process. All other duplicates are removed.

4. Experimentation

4.1. Data collection

We were granted access to a data set comprising several tens of thousands queries associated with several million web search results. Each query, search result pair was scored by a team of professional editors on a scale from 0 to 5. The rating reflects the web page's relevancy to the corresponding query, 0 being completely irrelevant and 5 being extremely relevant. From this dataset, we randomly selected two sets of 1,000 queries and the associated search results as our development datasets. Another 2,000 queries were randomly selected from the remaining data for evaluation. We also had access to a two-month query log of the same search engine, which contained aggregated frequencies of queries and also pairs of queries that were sent in succession by users of the search engine.

4.2. Evaluation metrics

A popular commercial Web search engine (identity withheld for blind review) is used as the baseline ranking scheme. We measure the ranking quality using the discounted cumulative gain (DCG) metric introduced in [14] and compared to other metrics in [15]. DCG assigns more weight to highly ranked documents, and allows us to define various levels of subjective relevance judgment for the human editors. For a given query *q*, DCG is defined as:

$$DCG(q) = \sum_{d=1}^{n} \frac{\left(2^{R(d)} - 1\right)}{\ln\left(1 + d\right)}$$

where $R(d)$ is the editorial rating of the *d*-th webpage. A higher DCG reflects a better ranking. DCG for the top *n* results generated by the search engine and *Q-Rank* are computed and compared. To preserve the confidentiality of business information, we cannot present the absolute DCG values on this data set, so we will quantify the performance changes in relative terms.

4.3. Development experiments

4.3.1. Comparison of three sets of parameters. Table 1 (a, b, and c) summarizes the investigated parameter settings when only Q_{ext} are used, only Q_{adj} are used, and both type of query contexts are used. The notation $|\cdot|$ refers to the number of query extensions and adjacent queries employed by *Q-Rank*. When adjacent queries are used, we employ an equal number of preceding and subsequent queries; for example, $|Q_{adj}| = 20$ means that 10 preceding and 10 subsequent queries are employed.

For each of the three main classes of experiments, *c* indicates the number of document candidates considered for re-ranking, *u* indicates that the top *u* ranked documents are kept unchanged, *bias* indicates whether or not the bias towards the initial ranking (the factor $1/R(d)$ in Definition 1) is used.

Figure 2 (a, b, and c) shows the percentage of queries with improved ranking (measured in term of the DCG metric) plotted against various query lengths for each of the three parameter setting for the three main classes of experiments. When computing DCG, the output ranking range *n* was set to 15. In all parameter

Table 1. *Q-Rank* investigated parameter space.

(a) when using Q_{ext} only (γ=1)

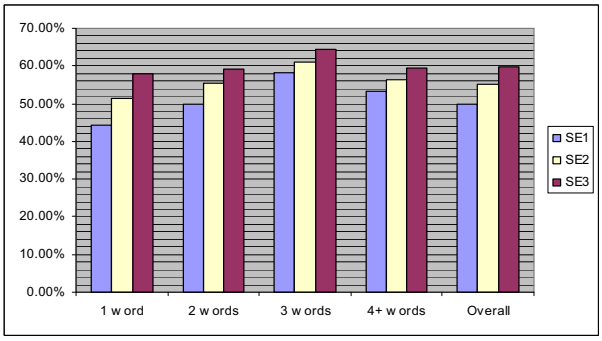
Parameters:	$\lvert Q_{ext}\rvert$	*c*	*u*	*bias*
SE_1	20	50	1	False
SE_2	20	50	2	False
SE_3	20	50	2	True

(b) when using Q_{adj} only (γ=0)

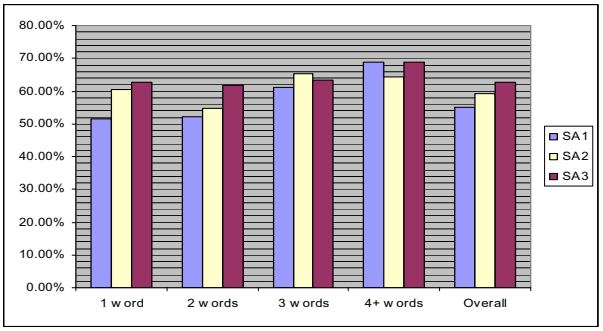
Parameters:	$\lvert Q_{adj}\rvert$	*c*	*u*	*bias*
SA_1	20	50	1	False
SA_2	20	50	2	False
SA_3	20	50	2	True

(c) when using both Q_{ext} and Q_{adj}

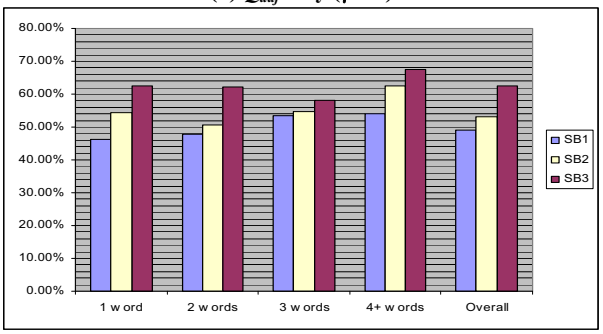
Parameters: γ=.5	$\lvert Q_{ext}\rvert+\lvert Q_{adj}\rvert$	*c*	*u*	*bias*
SB_1	20 + 20	50	1	False
SB_2	20 + 20	50	2	False
SB_3	20 + 20	50	2	True



(a) Q_{ext} only (γ = 1)



(b) Q_{adj} only (γ = 0)



(c) Q_{ext} and Q_{adj}, γ = 0.5

Figure 2. Percentage of queries with improved ranking broken down by query length

settings except SE_1 and SB_1, *Q-Rank* is able to improve the DCG score for more than 50% of the queries with changed rankings, which cover between 64% and 72% of the total number of queries (as shown in Table 2). These improvements can be viewed as major when considering the fact that on average, for each query, less than 20 documents had relevance scores assigned by judges and typically, the top 10 search results were among those judged, and we assigned a score of 0 to all documents without editorial ratings. Consequently, when replacing a document that had a lower but positive score with an unjudged but potentially relevant document, the DCG score was decreased.

The numerical values for this first set of experiments are shown in Table 2(a). One important observation is that when taking into account the initial ranking (i.e. *bias* = true), *Q-Rank* performs better overall (SE_3, SA_3, and SB_3 consistently outperform the other settings). Another observation is that using adjacent queries alone seems to produce the best ranking; this is further verified in a number of experiments discussed in the following sections.

4.3.2. Various query lengths. With one exception (SA setting for query length 3), we observed a consistent pattern in which keeping the top two search results unchanged and using the bias towards the initial ranking outperformed the other settings for all query lengths. While we initially expected that most of the improvements will surface for one-word queries, we observed that due to the richness of the search query logs and to the back-off strategy employed in generating the query context, *Q-Rank* was able to improve the ranking substantially even for longer queries (three or more words).

4.3.3. Various ranking ranges (*n*). Table 2(b) summarizes the performance of *Q-Rank* with various ranking ranges (*n*). Because we have confirmed from previous experiments that *Q-Rank* performs better with bias towards the initial ranking, only the parameter settings SE_3, SA_3, and SB_3 are investigated here. The increase of *n* does not guarantee a better performance. In fact, *Q-Rank* achieves the best ranking constantly for $n = 10$ in all three parameter settings. As discussed above, the main reason for this may be the fact that most results with lower initial ranks do not have editorial ratings.

Again, we observe that when using adjacent queries alone *Q-Rank* performs the best (SA_3, $n = 10$). One explanation is that adjacent queries represent the frequent user modifications of the target query when the users are not satisfied with the search results (because either they underspecified their intent or they overspecified it in relation to the lexical composition of Web documents). Such reformulated queries can better represent users' real search intents.

4.3.4. Various numbers of unchanged top results (*u*). Table 2(c) summarizes the performance of *Q-Rank* with various numbers of unchanged top results (*u*), in which the top *u* ranked results remain the same. $SE_3'/SA_3'/SB_3'$ indicates that except for *u*, other parameters remain the same as in $SE_3/SA_3/SB_3$. Here *n* is set to 10 because it has been shown to achieve the best performance in the previous experiments. While *Q-Rank* tends to push up quality and relevant documents that were originally ranked lower, we also observe that keeping unchanged the top $u = 2$ search results produces better performance. There are good reasons to do so. First, there are some websites designed completely in still images or Flash animations with little or no textual information, e.g. the automobile maker Mercedes' page (http://www.mercedes-amg.com/). While the URLs of such websites may contain the query terms, it is extremely hard or even impossible to generate meaningful snippets for them in order to match the query log context; thus, *Q-Rank* would give very low rank scores to such sites. Second, the major commercial search engines often employ lists of *definitives* to be shown as the top results (i.e. Web pages that are editorially matched to queries), especially for well-known organizations and businesses, which also suggests that we should keep the top one or two search results unchanged.

4.3.5. Various numbers of re-rank candidates (*c*). Table 2(d) summarizes the performance of *Q-Rank* with various numbers of document candidates *c* considered in the re-ranking process. Here *u* is set to 2 because the system was shown to achieve the best performance for this setting in previous experiments. An interesting observation is that increasing the number of re-rank candidates does not yield better results, which is consistent across all sets of parameters. An explanation is that there is no sufficient annotated data: it is very likely that most of the rated websites are already in the top 30 results, as explained at the beginning of this section, and thus, taking into account a larger pool of document candidates does not necessarily improve the DCG score of the final ranking.

4.3.6 The best empirical value of γ. We also ran a series of experiments on the development datasets to determine the best value of γ for *Q-Rank*. Figure 3 plots the percentage of queries with improved ranking when γ increases from 0 to 1 in 0.1 increments. On average, *Q-Rank* improves the rankings for 75.8% of the re-ranked queries. The percentile peaks at $\gamma = 0$ (78.5%), which is consistent with our previous findings that using adjacent queries alone achieves the best results. Figure 4 shows the actual increase (in percentage) of the DCG scores when γ grows from 0 to 1 in 0.1 increments.

When we select γ to be 0.5, the DCG scores are increased by an average 6.81% for 76.3% (538 out of 707) of the re-ranked queries; this amounts to more than half (53.8%) of the queries in the development dataset.

4.3.7. Full-text vs. document snippets. To compare the performance using full-text and document snippets, we downloaded the documents retrieved by the search engine and performed *Q-Rank* using the full-texts of these documents. Table 3 summarizes the obtained results. We plot the percentage of queries with improved ranking and the percentage of actual DCG improvement in Figure 5 and 6. The abrupt curves are due to the fact that only the values at $\gamma = 0$, $\gamma = 0.5$, and $\gamma = 1$ are plotted. Not surprisingly, using full-text Q-Rank has achieved the best performance so far for all three sets of parameters. On average, Q-Rank is able to improve the ranking for 81.4% of the queries with an average DCG increase of 8.65%. Once again, we observe that using the adjacent queries alone improves results ranking for the highest percentage of queries, while in practice choosing $\gamma = 0.5$ is a good trade-off.

Table 3. *Q-Rank* using full-text. Percentage of queries with improved ranking is shown in the first row. Percentage of DCG improvement (*Q-Rank* vs. *S*) is shown in the last row.

	***n*=10, *u*=2, *c*=30, γ=1**	***n*=10, *u*=2, *c*=30, γ=0**	***n*=10, *u*=2, *c*=30, γ =.5**
Improved / Total changed	**80.7%** (522/647)	**81.8%** (568/694)	**81.7%** (579/709)
DCG Improvement	8.7%	8.0%	9.3%

Table 2. Comparison of results for various parameter settings. Percentage of queries with improved ranking is shown.

(a) The investigated parameter space (as defined in 4.3.1; n=15).

	Q_{ext} only			Q_{adj} only			$Q_{ext} + Q_{adj}$		
	SE_1	**SE_2**	**SE_3**	**SA_1**	**SA_2**	**SA_3**	**SB_1**	**SB_2**	**SB_3**
All queries (Improved/Total changed)	**49.8%** (319/640)	**55.2%** (354/641)	**59.7%** (384/643)	**54.9%** (386/703)	**59.2%** (416/703)	**62.8%** (443/706)	**49.0%** (351/717)	**53.3%** (383/719)	**62.5%** (451/722)
One-word queries	44.4%	51.6%	57.8%	51.5%	60.4%	62.8%	46.3%	54.5%	62.5%
Two-word queries	49.8%	55.4%	59.3%	52.1%	54.7%	61.6%	47.8%	50.5%	62.1%
Three-word queries	58.3%	60.9%	64.4%	61.2%	65.3%	63.3%	53.3%	54.6%	58.1%
Four+ word queries	53.1%	56.3%	59.4%	68.9%	64.4%	68.9%	54.2%	62.5%	67.4%

(b) Various ranking ranges (n).

	SE_3 (c=50, u=2, bias=true)			**SA_3 (c=50, u=2, bias=true)**			**SB_3 (c=50, u=2, bias=true, γ=0.5)**		
	n = 10	**n = 15**	**n = 20**	**n = 10**	**n = 15**	**n = 20**	**n = 10**	**n = 15**	**n = 20**
All queries (Improved/Total changed)	**63.8%** (410/643)	**59.7%** (384/643)	**59.2%** (377/637)	**66.4%** (469/706)	**62.8%** (443/706)	**62.2%** (439/706)	**62.6%** (452/722)	**62.5%** (451/722)	**59.0%** (426/722)
One-word queries	60.8%	57.8%	55.6%	66.0%	62.8%	60.6%	60.3%	62.5%	53.9%
Two-word queries	63.6%	59.3%	58.4%	64.6%	61.6%	59.8%	62.3%	62.1%	58.2%
Three-word queries	69.6%	64.4%	66.1%	69.4%	63.3%	64.6%	65.1%	58.1%	63.2%
Four+ word queries	62.5%	59.4%	64.5%	71.1%	68.9%	77.8%	66.7%	67.4%	72.9%

(c) Various numbers of unchanged top results (u).

	SE_3' (c=50, n=10, bias=true)			**SA_3' (c=50, n=10, bias=true)**			**SB_3' (c=50, n=10, bias=true, γ=0.5)**		
	u = 0	**u = 1**	**u = 2**	**u = 0**	**u = 1**	**U = 2**	**u = 0**	**u = 1**	**u = 2**
All queries (Improved/Total changed)	**53.0%** (339/640)	**62.1%** (398/641)	**63.8%** (410/643)	**50.4%** (355/704)	**62.6%** (441/704)	**62.6%** (469/706)	**52.8%** (380/720)	**61.1%** (440/720)	**62.6%** (452/722)
One-word queries	46.7%	60.6e%	60.8%	41.6%	58.4%	66.0%	47.3%	58.6%	60.3%
Two-word queries	55.7%	61.5%	63.6%	51.6%	61.6%	64.6%	56.2%	61.2%	62.3%
Three-word queries	55.7%	64.4%	69.6%	57.8%	67.4%	69.4%	52.6%	61.8%	65.1%
Four+ word queries	56.3%	68.8%	62.5%	57.8%	73.3%	71.1%	54.2%	68.8%	66.7%

(d) Various numbers of re-rank candidates (c).

	SE_3" (n=10, u=2, bias=true)			**SA_3" (n=10, u=2, bias=true)**			**SB_3" (n=10, u=2, bias=true, γ=0.5)**		
	c = 20	**c = 30**	**c = 40**	**c = 20**	**c = 30**	**c = 40**	**c = 20**	**c = 30**	**c = 40**
All queries (Improved/Total changed)	**75.3%** (490/651)	**78.2%** (507/648)	**73.0%** (473/648)	**77.7%** (543/699)	**80.3%** (557/694)	**75.1%** (521/694)	**76.3%** (545/714)	**79.3%** (562/709)	**71.0%** (503/709)
One-word queries	67.9%	78.2%	67.3%	74.7%	81.0%	68.1%	73.4%	80.2%	66.8%
Two-word queries	78.8%	77.5%	75.2%	76.5%	78.5%	77.3%	75.6%	77.6%	70.9%
Three-word queries	75.3%	80.4%	76.1%	85.5%	80.6%	76.9%	78.6%	80.0%	73.9%
Four+ word queries	87.2%	79.5%	79.5%	81.6%	87.8%	87.8%	88.2%	84.3%	82.4%

5. Evaluation Results

Finally, we test *Q-Rank* on the evaluation dataset which contains 2,000 queries randomly selected from the query logs of a popular commercial search engine, using the set of parameter values that achieved best performance on the development data. Results are displayed in Figure 7 and 8, showing an improvement on 81.8% of the re-ranked queries with an average increase in the DCG scores of 8.99%. The characteristics of these results are very consistent with those from the development dataset.

Figure 9 plots the percentage of queries with improved rankings for various query lengths. There is no consistent pattern in the performance changes that correlates with query length. Interestingly, for long queries (four words or more), the adjacent queries seem to work best. This can be explained on one hand, by the lower number of query extensions and thus, the need to back-off, on the other hand, by the less noisy adjacent query contexts.

6. Discussions and Future Work

Recent studies [19, 28] on identifying and studying users' search goals showed that most queries can be classified as either informational or navigational. Users submit informational queries when they intend to obtain relevant information from the Web and navigational queries when they intend to reach a specific (and typically authoritative) website. The distribution of clicks on the search results of the navigational queries tends to be skewed because the users are often able to recognize the particular website they had in mind [23]. For similar reasons, such queries also have fewer adjacent queries. Thus, *Q-Rank*'s performance with Q_{ext} and Q_{adj} suggests that the proposed system works better for informational queries. We plan to do a careful empirical study to confirm or reject this hypothesis.

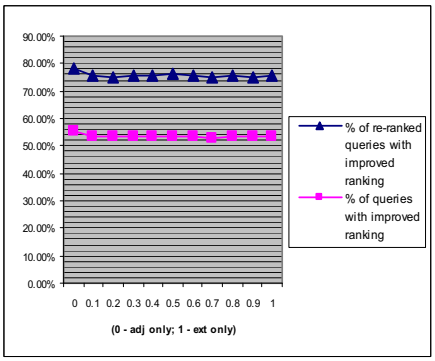


Figure 3. Percentage of queries with increased DCG scores when γ grows from 0 to 1 at a step of 0.1. Using adjacent queries alone (γ = 0) has the highest percentage, which is consistent with previous findings.

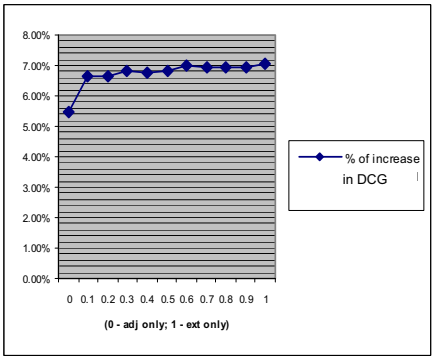


Figure 4. Percentage of increased DCG scores when γ grows from 0 to 1 at a step of 0.1.

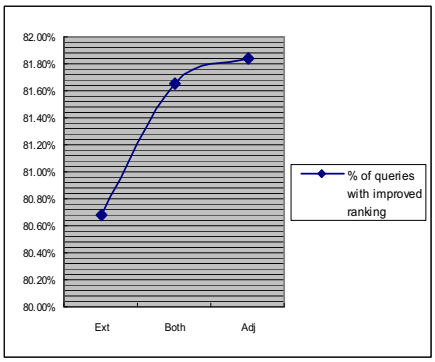


Figure 5. Percentage of queries with increased DCG scores at γ = 1, γ = 0.5, and γ = 0, using full-text of the retrieved documents. Again, using adjacent queries alone (γ = 0) achieves the best performance.

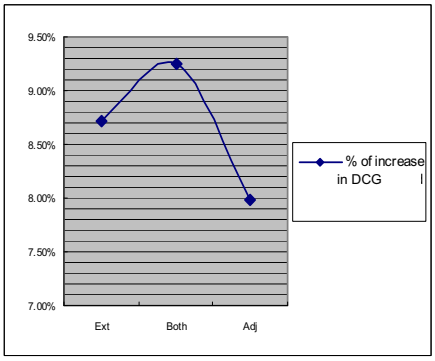


Figure 6. Percentage of increased DCG scores for γ = {1, 0.5, 0} when using the full-text of the documents.

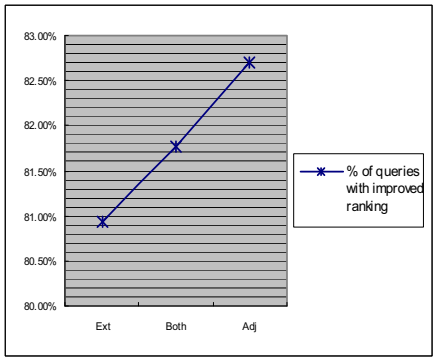


Figure 7. Percentage of queries with increased DCG scores at γ = 1, γ = 0.5, and γ = 0, using document snippets, on the test set.

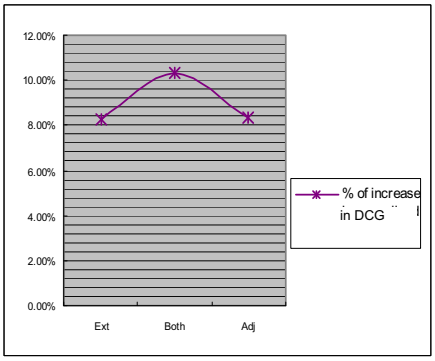


Figure 8. Percentage of increased DCG scores at γ = 1, γ = 0.5, and γ = 0 on the test set.

Another interesting future direction is to investigate whether the variety in the top search results introduced by *Q-Rank* is truly beneficial to the users. While query logs contain aggregated knowledge about collective preferences, each user may have his or her preference as for what should rank higher. For example, Google's top 10 results[2] for the query *"cats"* are mostly about *cat* the animal. On the contrary, the top 10 results from *Q-Rank* also include websites about the popular Broadway musical and the Charlotte Area Transit System (CATS). Apparently, the results produced by *Q-Rank* are more diversified, and potentially capture a wider set of interests for a particular query. Such advantage may not be obvious when the results are evaluated with the DCG metrics, because different topics for the same query are not distinguished as long as they are all relevant. We plan to extend the current work to include a user study which will be more effective to measure such preferences. Following this line, an interesting modification to *Q-Rank* is to consider personal query logs instead of the aggregated query logs, which could return ranking tailored particularly to the tastes of the user. Similar research directions were presented recently in [34]. We plan to investigate such a *personalized Q-Rank* in the near future.

While the length of the time frame used to construct Q_{ext} and Q_{adj} is currently 2 months, a smaller time frame may reflect the more *up-to-date* trends of interests for the same query. On the other hand, query extensions and adjacent queries can be weighted individually according to the distance (in time) between their appearances and that of the target query, and their semantic similarities. Given the importance of ordering, which type of the query contexts performs better, those that precede or follow the initial query? We also plan to investigate these temporal issues in our future work.

Finally, we conclude our discussion with a comment about the current implementation of *Q-Rank*, which uses only document snippets (rather than the full-text of the retrieved documents). This choice makes it very efficient in terms of the overhead added to an existing ranking system, while paying a relatively small cost in its effectiveness compared to using the whole documents. Yet, this introduces a dependency on the quality of the snippets generated by a search engine and may increase the vulnerability to web

[2] A screenshot captured at the time of writing is available at http://static.flickr.com/38/91887416_ff469496ab_o.jpg

spam, even if limited by the fact that only the top *N* search results are used as re-rank candidates.

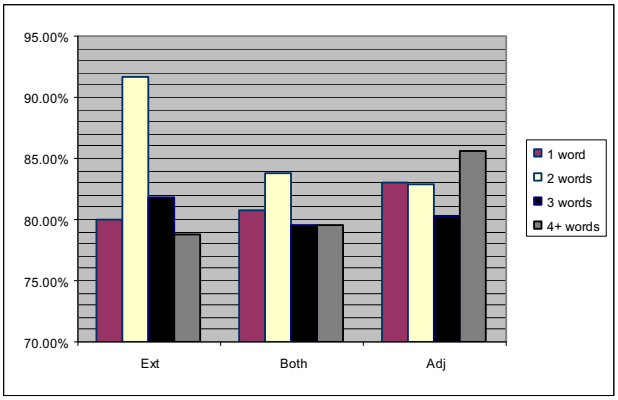


Figure 9. Percentage of queries with increased DCG scores in the test set, broken down by query length.

7. Conclusion

We proposed *Q-Rank*, a re-ranking algorithm which uses distributional information of the query contexts extracted from search engine query logs, to effectively and efficiently improve the relevance ranking of Web search results. We evaluated our proposal with a series of comprehensive experiments to empirically determine the impact of various factors (e.g. query length, re-ranking range, interpolation coefficient between different types of query contexts), and to select the optimal parameters for the re-ranking algorithm. *Q-rank* consistently outperformed the baseline ranking algorithm, demonstrating an 9% improvement in relevance ranking quality for 81.8% of the re-ranked queries.

8. References

[1] Agichtein, E., Brill, E., and Dumais, S. Improving web search ranking by incorporating user behavior information. In Proc. of the 29th ACM SIGIR Conference (SIGIR '06), pp. 19-26. 2006.

[2] Bharat, K., and Mihaila, G. When Experts Agree: Using Non-Affiliated Experts to Rank Popular Topics. In Proc. of the 10th International World Wide Web Conference, pp. 597-602, 2001.

[3] Billerbeck, B., Scholer, F., Williams, H. E., Zobel, J. Query Expansion using Associated Queries. In Proceedings of CIKM, pp. 2-9. 2003.

[4] Broder, A. A Taxonomy of Web Search. SIGIR Forum, pp. 3-10, 2002.

[5] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G. Learning to Rank using Gradient Descent. In Proc. of the 22nd International Conference on Machine Learning, 2005.

[6] Cucerzan, S. and Brill, E. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In Proceedings of the EMNLP Conference, pp. 293-300. 2004.

[7] Cui, H., Wen, J. Nie, J., and Ma, W. Probabilistic Query Expansion Using Query Logs. In Proc. of the 11th WWW Conference, 2002.

[8] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T.K., and Harshman, R. Indexing by Latent Semantic Analysis. Journal of American Society of Information Sciences, 41 (6), pp. 391-407. 1990.

[9] Dumais, S., Cutrell, E., and Chen, H. Optimizing Search by Showing Results in Context. In Proc. of SIGCHI Conference. 2001.

[10] Freund, Y., Iyer, R., Schapire, R., and Singer, Y. An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research, 4:933-969. 2003.

[11] Fu, X., Budzik, J., and Hammond, K. Mining Navigation History for Recommendation. In Proc. of the 5th International Conference on Intelligent User Interfaces, pp. 106-112. 2000.

[12] Haveliwala, T. Topic-Sensitive PageRank. In Proc. of the 11th International World Wide Web Conference, pp. 517-526, 2002.

[13] Jansen, J., and Spink, A. An Analysis of Web Documents Retrieved and Viewed", In Proc. of the 4th International Conference on Internet Computing, 2003.

[14] Jarvelin, K., and Kekalainen, J. IR Evaluation Methods for Retrieving Highly Relevant Documents. In Proc. of SIGIR '00, pp. 41-48. 2000.

[15] Jarvelin, K., and Kekalainen, J. Cumulated Gain-based Evaluation of IR Techniques. In ACM Transactions on Information Systems, 20: 4, 422-446. 2002.

[16] Joachims, T. Optimizing search engines using clickthrough data. In Proc. of ACM Conference on Knowledge Discovery and Data Mining. 2002.

[17] Jones, R., Rey, B., Madani, O., and Greiner, W. Generating Query Substitutions. In Proceedings of the 15th international conference on World Wide Web, pp. 387-396, 2006.

[18] Kraft, R. and Zien, J. Mining Anchor Text for Query Refinement. In Proc. of the World Wide Web Conference, pp. 666 - 674. 2004.

[19] Kang, I., and Kim, G. Query Type Classification for Web Document Retrieval. In Proc. of the 26th ACM SIGIR Conference, 2003.

[20] Kleinberg, J. Authoritative Sources in a Hyperlinked Environment. In Proc. ACM SIAM Symposium on Discrete Algorithms, 668-677, 1998.

[21] Koenemann, J. & Belkin, N.J. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In Proc. of the Conference on Human Factors in Computing Systems, 1996.

[22] Lau, T. and Horvitz, E. Patterns of search: Analyzing and modeling web query refinement. In Proc. of the 7th International Conference on User Modeling, pp. 119-128. 1999.

[23] Lee, U., Liu, Z., and Cho, J. Automatic Identification of User Goals in Web Search. In Proc. of the 14th World Wide Web Conference. 2005.

[24] Mitra, M., Singhal, A., and Buckley, C. Improving Automatic Query Expansion. In Proc. of the 21st ACM SIGIR Conference, 206-214. 1998.

[25] Nambiar, U., and Kambhampati, S. Providing Ranked Relevant Results for Web Database Queries. In Proc. of the WWW Conference, pp. 314-315. 2004.

[26] Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank Citation Ranking: Brining Order to the Web. Technical Report, Stanford University Database Group, 1998.

[27] Qiu, Y. and Frei, H. Concept-based Query Expansion. In Proc. of the 16th ACM SIGIR Conference, pp. 160-169. 1993.

[28] Rose, D., and Levinson, D. Understanding User Goals in Web Search. In Proc. of the World Wide Web Conference, pp. 13-19. 2004.

[29] Salton, G., and Buckley, C. Improving Retrieval Performance by Relevance Feedback. Journal of the American Society for Information Science. 41:288-297, 1990.

[30] Shashua, A., and Levin, A. Ranking with Large Margin Principle: Two Approaches. In Proc. of the NIPS, 2002.

[31] Shen X., and Zhai, C. Exploiting Query History for Document Ranking in Interactive Information Retrieval. In Proc. of SIGIR, 377-378. 2003.

[32] Shen, X., Tan, B., Zhai, C. Implicit User Modeling for Personalized Search. In Proceedings of CIKM, 2005.

[33] Sun, R, Ong, C., and Chua, T. Mining dependency relations for query expansion in passage retrieval. In Proc. of SIGIR, pp.382-389, 2006.

[34] Teevan, J., Dumais, S., Horvitz, E. Personalizing Search via Automated Analysis of Interests and Activities. In Proc. of SIGIR, 449-456, 2005.

[35] Pew Internet Report 2007, Retrieved at http://www.pewinternet.org/reports.asp.