Mining Query Subtopics from Search Log Data

Yunhua Hu*, Yanan Qian†, Hang Li*, Daxin Jiang*, Jian Pei‡, and Qinghua Zheng†

*Microsoft Research Asia, Beijing, China
†SPKLSTN Lab, Xi'an Jiaotong University, China
‡Simon Fraser University, Burnaby, BC, Canada
{yuhu,hangli,djiang}@microsoft.com, yanan.qian@stu.xjtu.edu.cn, jpei@cs.edu.ca, qhzheng@mail.xjtu.edu.cn

ABSTRACT

Most queries in web search are ambiguous and multifaceted. Identifying the major senses and facets of queries from search log data, referred to as query subtopic mining in this paper, is a very important issue in web search. Through search log analysis, we show that there are two interesting phenomena of user behavior that can be leveraged to identify query subtopics, referred to as 'one subtopic per search' and 'subtopic clarification by keyword'. One subtopic per search means that if a user clicks multiple URLs in one query, then the clicked URLs tend to represent the same sense or facet. Subtopic clarification by keyword means that users often add an additional keyword or keywords to expand the query in order to clarify their search intent. Thus, the keywords tend to be indicative of the sense or facet. We propose a clustering algorithm that can effectively leverage the two phenomena to automatically mine the major subtopics of queries, where each subtopic is represented by a cluster containing a number of URLs and keywords. The mined subtopics of queries can be used in multiple tasks in web search and we evaluate them in aspects of the search result presentation such as clustering and re-ranking. We demonstrate that our clustering algorithm can effectively mine query subtopics with an F1 measure in the range of 0.896-0.956. Our experimental results show that the use of the subtopics mined by our approach can significantly improve the state-of-the-art methods used for search result clustering. Experimental results based on click data also show that the re-ranking of search result based on our method can significantly improve the efficiency of users' ability to find information.

Keywords

Search Log Mining, User Behavior, Query Subtopics, Clustering, Search Result Clustering

1. INTRODUCTION

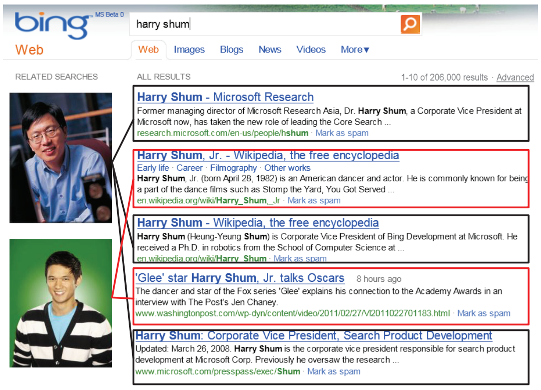
Understanding the search intent of users is essential for satisfying a user's search needs. How to best represent query intent is still an ongoing research problem. One consensus among the researchers is that the intents of queries can be characterized along multiple dimensions. The intents of a query can be represented by its search goals, such as informational, navigational, and transactional [7]. It can also be represented by semantic categories or topics [8, 23, 24]. Furthermore, it can be represented by subtopics (cf., [14]), denoting multiple senses or multiple facets of the query.

Most queries are ambiguous or multifaceted [14]. For example, 'harry shum' is an ambiguous query, which may refer to an American actor, a vice president of Microsoft, or another person named Harry Shum. 'Xbox' is a multifaceted query. When people search for 'xbox', they may be looking for information on different facets of an Xbox, such as 'online game', 'homepage', and 'marketplace'. Note that a query can be both ambiguous and multifaceted. The more frequent a query is, the more likely that it has multiple senses or facets. The major difference between the topics and subtopics of a query is that the former is more coarse-grained and related to other queries, while the latter is fine-grained and is only about the query in question.

Identifying the major subtopics of a query is very important for many search tasks such as personalized search, query suggestion, and search result presentation including clustering, re-ranking, and diversification. In this paper, we aim to automatically mine the major subtopics (senses and facets) of queries from the search log data. Although there is some related work, the subtopic mining problem as defined in this paper does not seem to have been studied previously.

We performed a comprehensive study of the two phenomena, referred to as 'one subtopic per search' (OSS) and 'subtopic clarification by additional keyword' (SCAK) respectively. We show that we can mine the subtopics of queries from search log data by effectively using the phenomena. We then represent the senses or facets of a query by a number of URLs and keywords. We observe that this can be done with high accuracy for head queries. Although the phenomena have been observed or mentioned in previous work (cf., [26]), there has not been a detailed study conducted on them, as far as we know.

If a user clicks multiple URLs after submitting a query (the multiple URLs are also referred to as multi-clicks in this paper), then the clicked URLs tend to represent the same subtopic, which is called one subtopic per search (OSS). Figure 1 shows an example of search results for the query 'harry shum' on a web search engine. The result contains URLs about two Harry Shum's. The search log data indicates that URLs 1, 3, and 5 are usually clicked together in individual searches, and URLs 2 and 4 are clicked together. Each group of URLs actually corresponds to one sense, i.e., one subtopic. Gale and Church discovered a similar phenomenon in natural language texts, called 'one sense per discourse' [19], which means an ambiguous word usually only has one sense in a specific context. Users often add additional keywords (in most cases, one additional keyword) to a query to expand the query in order to clarify its subtopic. This phenomenon is called subtopic clarification by additional keyword (SCAK). As a result, the URLs clicked after searching both with the original and the expanded queries tend to represent the same subtopic and the keyword also tends to be indicative of the subtopic. For example, people may submit 'harry shum microsoft' as an expanded query to specify the subtopic. The URLs clicked in searches for both 'harry shum' and 'harry shum microsoft' usually represent the same subtopic, Harry Shum from Microsoft, and therefore 'microsoft' becomes a keyword of the subtopic.



Harry Shum - Microsoft Research
Former managing director of Microsoft Research Asia, Dr. **Harry Shum**, a Corporate Vice President at Microsoft now, has taken the new role of leading the Core Search ...
research.microsoft.com/en-us/people/**hshum** · Mark as spam

Harry Shum, Jr. - Wikipedia, the free encyclopedia
Early life · Career · Filmography · Other works
Harry Shum, Jr. (born April 28, 1982) is an American dancer and actor. He is commonly known for being a part of the dance films such as Stomp the Yard, You Got Served ...
en.wikipedia.org/wiki/**Harry_Shum**,_Jr · Mark as spam

Harry Shum - Wikipedia, the free encyclopedia
Harry Shum (Heung-Yeung **Shum**) is Corporate Vice President of Bing Development at Microsoft. He received a Ph.D. in robotics from the School of Computer Science at ...
en.wikipedia.org/wiki/**Harry_Shum** · Mark as spam

'Glee' star **Harry Shum**, Jr. talks Oscars 8 hours ago
The dancer and star of the Fox series 'Glee' explains his connection to the Academy Awards in an interview with The Post's Jen Chaney.
www.washingtonpost.com/wp-dyn/content/video/2011/02/27/VI2011022701183.html · Mark as spam

Harry Shum: Corporate Vice President, Search Product Development
Updated: March 26, 2008. **Harry Shum** is the corporate vice president responsible for search product development at Microsoft Corp. Previously he oversaw the research ...
www.microsoft.com/presspass/exec/**Shum** · Mark as spam

Figure 1: Search result for query 'Harry Shum'.

We employ a clustering algorithm to group URLs as well as keywords into clusters, where each cluster represents one subtopic of a query. For example, one cluster for query 'harry shum' may contain the home page of the Microsoft Harry Shum, his wiki profile, and the keywords 'microsoft' and 'bing'. The clustering is performed based on the two phenomena of user behavior, as described above. Specifically, for each query and its expanded queries (the original query plus keywords), the related click log data is collected. Clustering is then carried out on all the clicked URLs. If two URLs are clicked together many times in the click data, then they are likely to be clustered together. If two URLs are clicked both under the original and expanded queries, then they are likely to be clustered together. We employ a special data structure consisting of a prefix tree and a suffix tree to facilitate recursive execution of the clustering algorithm on the log data. In this way, the mining of subtopics can be conducted very efficiently.

We conducted experiments to measure the accuracy of our clustering method on a TREC data set and a new data set in which the subtopics of both queries are labeled. For ambiguous subtopics, our method can achieve a B-cubed F1 from 0.925 to 0.956. For multifaceted subtopics, our method can achieve a B-cubed F1 from 0.896 to 0.930 in terms of B-cubed F1. The results indicate that our method can effectively perform query subtopic mining.

We also evaluated our method in search result clustering. We compared our method with a state-of-the-art method of search result clustering. Experimental results show that our method can significantly improve accuracy. The improvements are 5.4% in terms of B-cubed precision and 6.1% in terms of B-cubed recall. When compared side-by-side with human generated results, we also find that our method is significantly better than the baseline.

We further evaluated our method on search result reranking, in which the user is confirmed with the subtopic she has in mind and re-ranking of search results based on user feedback is performed. We used the average click position in the log data as the evaluation measure. The results show that our method can boost the average click position 0.61 points higher, which implies a promising improvement in the user experience.

There are three major contributions in our work:

1. We have analyzed two phenomena in user search behavior that can be utilized to identify query subtopics (senses and facets).

2. We have developed a clustering method that can effectively and efficiently mine the subtopics on the basis of the phenomena and use them to represent query subtopics.

3. We have successfully applied the mined results to two tasks: search result clustering and re-ranking.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 describes the detailed analysis of user behavior in the log data for subtopic mining. Section 4 explains our approach to subtopic mining and Section 5 presents the experimental results. Section 6 presents two applications of our subtopic mining algorithm. Section 7 concludes the paper.

2. RELATED WORK

2.1 Search Log Mining

The intent of a query can be characterized along several dimensions, including search goals [7], semantic classes [5], topics [4], and subtopics [14]. The search goals can be informational, navigational, or transactional. The semantic classes of a query can be represented by ODP categories[1]. The topics or subtopics of a query can be represented by a number of queries or URLs. Topics are usually more coarse-grained and can cover multiple queries, while subtopics are more fine-grained and associated with a specific query [14]. For example, for the query 'xbox', all aspects related to Xbox including online game, market place, and even other game consoles are usually included in the topic of the query. In contrast, each aspect of Xbox is represented by one subtopic of the query.

Mining topics from search log data has been intensively studied[2]. Click-through bipartite graph data can be used for clustering queries and URLs. Specifically queries which share the same clicked URLs are considered similar. Methods for performing the task have been proposed (e.g., [4, 10, 15, 18, 21, 25, 27]). Beeferman et al. [4], for example, proposed conducting clustering on a click-through bipartite graph and viewing the obtained clusters as topics covering multiple queries. Radlinski et al. [25] proposed first using search session data to find similar queries, and then using a

[1] Open Directory Project: http://www.dmoz.org/.

[2] The topics or subtopics of queries cannot be obtained by simply clustering the search results (documents) of queries [25].

Table 1: Multi-clicks in search logs of 'harry shum'

Multi-clicks	Frequency
"http://research.microsoft.com/en-us/people/hshum", "http://en.wikipedia.org/wiki/Harry_Shum", "http://www.microsoft.com/presspass/exec/Shum/"	50
"http://en.wikipedia.org/wiki/Harry_Shum,_Jr", "http://www.imdb.com/name/nm1484270/"	95

click-through bipartite graph to refine the discovered queries that are similar, and finally grouping the similar queries into the same clusters. The clusters containing the same query are then regarded as topics of the query.

There are some similarities but also significant differences between our work and Radlinski et al.'s work. 1) The problem settings are different. In their work, a subtopic is represented by multiple queries, while in our work a subtopic is represented by keywords and URLs. 2) The information used for subtopic mining is different. They used search sessions to find frequently co-occurring query pairs, considered them candidates of similar query pairs, and finally used a click-through bipartite graph to filter unlikely queries. 3) The goals are also different. The mined results in their work were utilized for manual creation of TREC data. In this paper, we quantitatively evaluated our method on two search tasks, namely search result clustering and re-ranking.

Methods for utilizing a user's click behavior in individual searches have been developed, e.g., [12, 15]. Existing work aimed at modeling and predicting a user's click behavior while our work attempts to find the subtopics of queries. The exploitation of the prefix and suffix relationship between queries was also considered in the previous work, e.g., [26]. In our work, we not only use the prefix and suffix relationship between queries, but also the clicked URLs of the queries and our goal is to conduct query subtopic mining.

2.2 Search Result Presentation

When a query is ambiguous or multifaceted, presenting its search result by sense or facet to users would improve the search experience [9]. Many methods have been proposed, including search result clustering and dynamic ranking.

Search result clustering attempts to cluster the search results according to semantic classes, topics, or subtopics [11]. For example, Chen and Dumais [13] proposed using text classification technologies to classify search results into predefined categories. Zeng proposed using text clustering techniques to cluster a research result based on the titles and snippets of web pages [29]. See also [2, 16, 17]. The major characteristic of these methods is that they use the content of documents in search result clustering.

Wang and Zhai further extended the idea and proposed exploiting the search results of the current query as well as the search results of similar queries [26]. They first build query relationships. For each query, similar queries are found from previous search sessions. They then perform clustering on the clicked URLs of the query and its similar queries. They use the titles and snippets of web pages for similarity calculation. The major difference between their work and our work is that they did not consider mining and utilization of query subtopics. In this paper, we use Wang and Zhai's method as the baseline for search result clustering.

Recently, Brandt et al. proposed a new search UI called dynamic ranking model [6]. The mined subtopics by our method can also be used in the dynamic ranking model. Specifically, when the user hovers the mouse on an URL in the UI, we can dynamically show the other URLs in the same subtopic under the mouse-hovered URL.

3. TWO PHENOMENA

3.1 One Subtopic per Search

One subtopic per search (OSS) means that the jointly clicked URLs in a specific search are likely to represent the same subtopic. We conducted analysis on the phenomenon with a search log data set from a commercial web search engine (DataSetA as described in Table 8) and validated the correctness of this hypothesis. Our explanation of the phenomenon is: 1) Search users are rational and they do not randomly click on search results (cf., [1, 20]). 2) Users *usually* have one single subtopic in mind for a particular search.

Table 1 shows two groups of multi-clicked URLs (i.e., the multiple URLs clicked by a user in a query) from search logs for the query 'harry shum'. The three URLs in the first group are about one sense and the two URLs in the second group are about another sense. Users may accidently click URLs belonging to two different groups, but such cases are rare. This is reasonable, because if a user searches for a specific person then she usually would only click the URLs about the person and skip the URLs about other people. We call such user behavior 'one subtopic per search'.

We examined the accuracy of using one subtopic per search as a rule for subtopic identification. By accuracy we mean the percentage of the multi-clicks that share the same subtopics among all multi-clicks. We sampled 10,000 groups of multi-clicks of individual queries in DataSetA, which are labeled based on the sense or facet. When all the URLs within the multi-clicks are about the same sense or facet, we consider that the rule has correctly identified a subtopic. Table 2 shows the accuracy versus the number of multi-clicks. We can see that the average accuracy of one subtopic per search is about 84%, suggesting that it is very helpful for identifying the subtopics of queries. Table 2 also shows that the accuracy will decrease when the number of URLs in a group of multi-clicks increases. That is to say, the more URLs there are clicked in a search, the less likely that the URLs share the same subtopic. Clicking many URLs in one query may indicate that the user is uncertain about what she is looking for or wants to learn about all aspects of the query. We also observe that the higher the frequency a group of multi-clicks has, the more likely they share the same subtopic. Figure 2 shows the relation between accuracy and frequency. When a multi-click is only observed twice in the log data, the average accuracy is 0.779. In contrast, when it happens more

Table 2: Accuracy of rule v.s. number of clicks

# of Clicks	2	3	4	≥ 5	Avg.
Accuracy	0.902	0.824	0.741	0.683	0.842

Table 4: Search logs of 'harry shum' ignoring click frequency

Query	Clicked URLs
harry shum	"http://research.microsoft.com/en-us/people/hshum", "http://en.wikipedia.org/wiki/Harry_Shum", "http://www.microsoft.com/presspass/exec/Shum/"
harry shum	"http://en.wikipedia.org/wiki/Harry_Shum,_Jr", "http://www.imdb.com/name/nm1484270/"
microsoft harry shum	"http://research.microsoft.com/en-us/people/hshum", "http://www.microsoft.com/presspass/exec/Shum/"
harry shum jr	"http://en.wikipedia.org/wiki/Harry_Shum,_Jr"
harry shum glee	"http://en.wikipedia.org/wiki/Harry_Shum,_Jr", "http://www.imdb.com/name/nm1484270/"

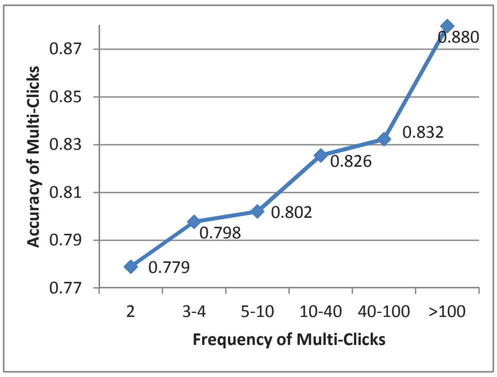


Figure 2: Accuracy of rule vesus frequency.

than 100 times, the accuracy can be as high as 0.880. We investigated the relationship between the click positions and the accuracy of the rule. Table 3 shows the results. 'Same subtopic' means that the rule is correct and the results represent the same subtopic. 'Different subtopic' means that the rule is incorrect and the results represent different topics. We can see that click positions do not affect the one subtopic per search phenomenon.

Table 3: Accuracy of rule v.s. click position

	Same subtopic	Different subtopic	All
# of Multi-clicks	8,421	1,579	10,000
Ave. Position	3.57	3.63	3.50
Ave. Click Intervals	3.23	3.27	3.20

Figure 3 shows the percentages of queries with multi-clicks. The queries with higher frequencies in search log data are more likely to have multi-clicks. For example, for the top 1% frequent queries in the search logs (corresponding to about 50% of search traffic), more than 90% of them have multi-clicks. Therefore, the phenomenon of one subtopic per search can help query subtopic mining for head queries.

3.2 Subtopic clarification by Additional Keyword

Many short queries are simply noun phrases, and long queries often contain short queries plus additional keywords, in which the keywords clarify the subtopics of the short queries. We refer to a short query as the original query, and longer queries containing the short query as expanded queries. The clicked URLs after searching with the original query and the expanded queries tend to represent the same subtopic. The keywords can also become labels of the subtopic. This is the phenomenon of subtopic clarification by additional keyword. The phenomenon can be explained in the following ways: 1) Search users are rational users. 2) Sometimes users tend to add additional keywords to specify the subtopics in their minds. We have also empirically validated the correctness of this hypothesis.

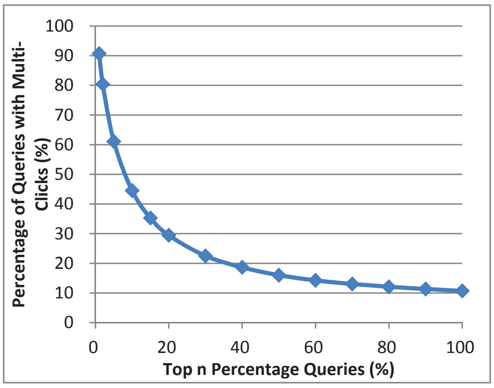


Figure 3: Distribution of top n% queries having multi-clicks.

Table 4 shows the expanded queries and their clicks with respect to the query 'harry shum' in the search log data. From the table, we can see that the clicked URLs of query 'microsoft harry shum' also appear in the former group of URLs corresponding to Harry Shum of Microsoft. The clicked URLs of 'harry shum jr' can be found in the latter group of URLs corresponding to the actor Harry Shum. The keywords 'microsoft' and 'jr' can be used to represent the two groups (subtopics) respectively. Therefore, each expanded query covers one subtopic of the original query with the keyword as the label of the subtopics.

We conducted analysis on the patterns of queries. We classified queries into four types. If the query is a single phrase, usually a noun phrase, then the type is 'Q'. The other three types are '$Q + W$', '$W + Q$', and 'Others', where 'W' denotes a keyword and 'Q' denotes a phrase. We estimated the distribution of the four types. We randomly selected 1,000 queries from the search log data. For each query, we checked whether it might be an expansion of another query. More specifically, we enumerated all queries that could be expanded from the considered query. The most reasonable query-expanded query pair is selected and then the query type is labeled. The clicked URLs were also used to help make the judgments. The distribution of query types is presented in Table 5. From the table, we can see that the types '$Q + W$' and '$W + Q$' consist of about 42% of the queries. '$Q+W$' is more popular than '$W+Q$'. We further examined whether the subtopics of the expanded queries are covered by those of the original queries. In other words, given two queries with the types 'Q' and '$Q + W$', we judged whether the subtopics of the latter are contained in the subtopics of

Table 5: Distribution of Query Types

Query Type	‘Q’	‘$Q+W$’	‘$W+Q$’	‘Others’
Percentage	0.455	0.255	0.165	0.125

the former. We randomly selected 500 pairs of queries with the forms ‘Q’ and ‘$Q + W$’, where the queries in Q form are from DataSetA, and studied the relationship between the subtopics of the original query and expanded query. If subtopics of an expanded query are contained in subtopics of the original query, then we say there is ‘subtopic overlap’ between the two. We also checked whether two queries share identical clicked URLs, and if so we call it ‘URL overlap’. In the investigation we found there is no significant difference between the results of queries in the forms ‘$Q+W$’ and ‘$W + Q$’. Thus, we merged the two types. Table 6 shows the results. From the results, we can see that 81.4% of expanded queries have URL overlap with the original queries, and 18.6% of expanded queries do not. For the expanded queries with URL overlap, 80.1% of them have subtopic overlaps with the original queries. For the expanded queries without URL overlap, 81.7% of them do not have subtopic overlap with the original queries. For example, ‘beijing’ and ‘beijing duck’, ‘fast’ and ‘fast food’, and ‘computer science’ and ‘computer science department’ do not have URL overlap, and they do not have subtopic overlap either. Hence we can exploit URL overlap to filter out unrelated ‘expanded queries’ (they are not ‘true’ expanded queries), and use the subtopics of the remaining expanded queries to help identify the subtopics of the original queries. This becomes another rule for subtopic mining.

Table 6: Relation of subtopic overlap and URL overlap between query and expanded query pair

	URL overlap	None URL overlap
All query pairs	0.814	0.186
Subtopic overlap	0.801	0.199
None subtopic overlap	0.183	0.817

We also investigated how many queries there are for which the rule can be applied. Figure 4 shows that the more popular (frequent) a query is, the more likely the rule is applicable. For the top 1% of frequent queries (about 50% of search traffic), about 88% of them have expansions, and on average there are 89 expanded queries for each query. The results indicate that the identification by keyword rule can be used for subtopic mining.

4. CLUSTERING METHOD

We employ a clustering method to mine subtopics of queries leveraging the two phenomena and search log data. The flow of the method is shown in Figure 5. In the preprocessing stage, we build an index to store all the queries and their clicked URLs. False expanded queries are then pruned from the index. In the clustering stage, the URLs associated with a query and its expanded queries are grouped into clusters, each representing one subtopic. In the post-processing stage, keywords are assigned to the clusters (subtopics) to enrich the representations of subtopics.

4.1 Preprocessing

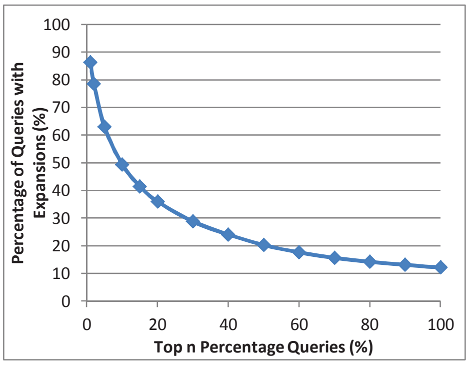


Figure 4: Distribution of top n% queries having expanded queries.

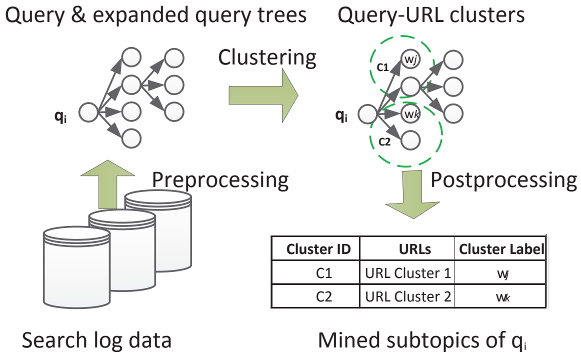


Figure 5: The flow of clustering method.

4.1.1 Indexing

We first index all the queries in an index consisting of a prefix tree and a suffix tree to facilitate efficient clustering. We only consider queries in three forms (‘Q’, ‘$Q + W$’ and ‘$W + Q$’), as discussed in Section 3.2.

We then segment queries and index them. In the prefix tree, query ‘Q’ and its expanded queries ‘Q+W’ are indexed in a father node and child nodes respectively. Search log data of each query is also stored in its node. With the prefix tree, we can easily find the expanded queries of any query. In the suffix tree query ‘Q’ and its expanded queries ‘W+Q’ are indexed as a father node and child nodes respectively. Figure 6 illustrates the data structure. For query ‘harry shum’, we can easily find its expanded queries ‘Q+W’ from the prefix tree, such as ‘harry shum jr’, ‘harry shum glee’, and ‘harry shum bing’. We can also easily find its expanded queries ‘W+Q’ from the suffix tree, such as ‘microsoft harry shum’ and ‘actor harry shum’.

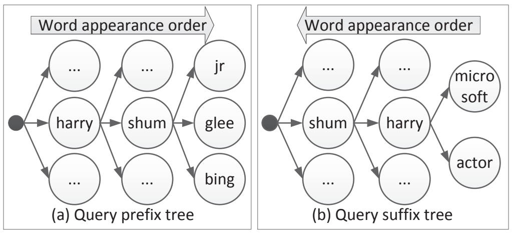


Figure 6: The data structures to index search logs.

4.1.2 Pruning

We then remove the false expanded queries from the prefix and suffix trees by using a heuristic rule: If a query 'Q' does not have URL overlap with its expanded queries 'Q+W' and 'W+Q' (i.e., there is no clicked URL shared by the queries), then those expanded queries will be viewed as false expanded queries and pruned from the trees. For example, 'fast food' and 'fast' do not have URL overlap, and thus 'fast food' will be pruned as a child node of 'fast'. Similarly, 'hot dog' and 'dog' do not have URL overlap so 'hot dog' will be pruned as child node of 'dog'. This heuristics is based on the discussions in Section 3.2. After pruning, only the 'true' expanded queries are stored in the prefix and suffix trees.

4.2 Clustering

We conduct clustering on the clicked URLs of each query and its expanded queries. Since all the queries are indexed in the trees, the clustering can be performed locally and recursively on the trees. The clustering of clicked URLs is guided by the two phenomena described in Section 3. After clustering, each group of clustered URLs is taken as one subtopic of the query in the father node.

4.2.1 Similarity Function

The clustering is repeatedly conducted on the clicked URLs of a query and its expanded queries on the trees. The similarity function between two clicked URLs is defined as a linear combination of three similarity sub-functions.

Specifically, the similarity function between URLs u_i and u_j is defined as

$$S(u_i, u_j) = \alpha S_1(u_i, u_j) + \beta S_2(u_i, u_j) + \gamma S_3(u_i, u_j) \quad (1)$$

Here S_1 is a similarity function based on the OSS phenomenon, S_2 is based on the SCAK phenomenon, S_3 is based on string similarities, with α, β, and γ as weights.

S_1 is defined as

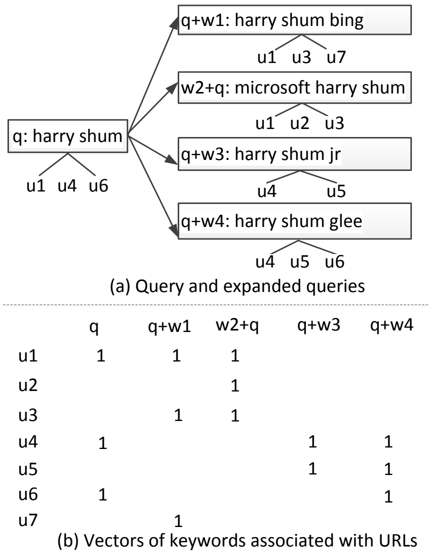
$$S_1(u_i, u_j) = \frac{\vec{m}_{u_i} \cdot \vec{m}_{u_j}}{||\vec{m}_{u_i}||_2 ||\vec{m}_{u_j}||_2} \quad (2)$$

where $\vec{m}_{u_i}$ and $\vec{m}_{u_j}$ denote the vectors of multi-clicks of u_i and u_j respectively, '$\cdot$ ' denotes the dot product, and $|| \cdot ||_2$ denotes the L_2 norm. Each element of the vector corresponds to one multi-click pattern, and its value represents the frequency of the multi-clicks. Intuitively, URLs u_i and u_j will become similar if they frequently co-occur in many multi-clicks.

S_2 is defined as

$$S_2(u_i, u_j) = \frac{\vec{w}_{u_i} \cdot \vec{w}_{u_j}}{||\vec{w}_{u_i}||_2 ||\vec{w}_{u_j}||_2} \quad (3)$$

where $\vec{w}_{u_i}$ and $\vec{w}_{u_j}$ denote the vectors of keywords associated with u_i and u_j respectively, $\cdot$ denotes the dot product, and $|| \cdot ||_2$ denotes the L_2 norm. Note that there is a null keyword, which is associated with the original query (father node). Intuitively, URLs u_i and u_j will become similar if they are clicked in searches of expanded queries containing the same keywords. Figure 7 shows the clustering of URLs u_1, u_2, u_3, etc on query q and its expanded queries $q + w1$, $q + w2$, and $q + w3$. The URLs are represented by vectors with each element corresponding to one keyword. S_3 simply represents the similarity between u_i and u_j as strings. Intuitively, two URLs will be viewed as similar if there are many words shared by the two URLs [22]. To conduct a similarity calculation, we segment a URL into tokens based on the slash symbols, and calculate the cosine similarity of the tokens as S_3.



(a) Query and expanded queries

	q	q+w1	w2+q	q+w3	q+w4
u1	1	1	1		
u2			1		
u3		1	1		
u4	1			1	1
u5				1	1
u6	1				1
u7		1			

(b) Vectors of keywords associated with URLs

Figure 7: Example of clustering using subtopic clarification by keyword.

4.2.2 Algorithm

We employ an agglomerative clustering algorithm to perform clustering. The algorithm has the advantage of ease of implementation. One can also consider employing other clustering algorithms. The specific algorithm is as follows:

Step 1: Select one URL and create a new cluster containing the URL.

Step 2: Select the next URL u_i, and make a similarity comparison between the URL and all the URLs in the existing clusters. If the similarity between URL u_i and URL u_j in one of the clusters is larger than threshold θ, then move u_i into the cluster. If u_i cannot be joined to any existing clusters, create a new cluster for it.

Step 3: Finish when all the URLs are processed.

4.3 Postprocessing

The output of the clustering process is clusters of URLs for each query and its expanded queries. The clusters which consist of only one URL are excluded. Each cluster represents one subtopic of the query. We further extract keywords from the expanded queries and assign them to the corresponding cluster as subtopic labels. As a result, each cluster not only consists of URLs but also retains keywords as cluster labels. The subtopic popularity can be further estimated from the frequency of clicked URLs in each cluster. Table 7 shows the mined search subtopics of query 'harry shum'. The first subtopic has three keywords and two URLs and the second subtopic has three keywords and three URLs.

5. EXPERIMENTS ON ACCURACY

5.1 Data Sets

We carried out experiments on the accuracy of our method

Table 7: Examples about mined subtopics of 'harry shum'

Subtopic ID	Keywords with frequency	URLs with frequency
1	'harry shum microsoft' : 201 'harry shum bing' : 80 'microsoft harry shum' : 22	"http://en.wikipedia.org/wiki/Harry_Shum": 961 "http://research.microsoft.com/en-us/people/hshum/": 317 "http://www.microsoft.com/presspass/exec/Shum/" : 98
2	'harry shum jr' : 2746 'harry shum glee' : 371 'harry shum junior' : 43	"http://en.wikipedia.org/wiki/Harry_Shum,_Jr." : 2999 "http://harryshumjr.com/" : 845 "http://www.imdb.com/name/nm1484270/" : 327

with three data sets. The first dataset (TREC Data) is from the TREC search result diversification track[3] in 2009. About 7% of URLs in the data set are not accessible on the web now, so we removed them and added new URLs in the search results of the queries at a commercial search engine. The remaining two datasets consist of queries and URLs randomly sampled from the logs of the commercial search engine in the EN-US market from June 1st, 2010 to March 31st 2011. To reduce the data size, we selected queries which occurred at least two times in one week during a year.

Human assessors labeled the subtopics by grouping the URLs associated with the queries. The guideline for the labeling is exactly the same as that in TREC. For ambiguous queries, it is usually easy to make distinctions between the subtopics of the queries. For multifaceted queries, it is sometimes difficult to make distinctions between the subtopics. We asked the assessors to refer to the examples in the TREC data as a spec in their data labeling. Table 8 gives the statistics of the datasets. 'Mixed queries' means that the queries are both ambiguous and multi-faceted.

Table 8: Statistics of three data sets

# of items	TREC	DataSetA	DataSetB
Queries	50	100	50
URLs	6,498	9,657	4,899
Subtopics	243	1,415	/
Ambiguous queries	11	57	20
Multifaceted queries	39	19	24
Mixed queries	0	24	6

5.2 Setting

The quality of a clustering algorithm is usually measured in terms of purity and inverse purity. Recently, Amigo et al. [3] showed that B-cubed precision and recall are more suitable for the evaluation of clustering results. We adopted B-Cubed precision, B-cubed recall, and B-cubed F1 as evaluation measures.

We split DataSetA into three parts and used 1/3 of them for parameter tuning and 2/3 of them for evaluation. The entire TREC dataset was used for evaluation. There are four parameters in our algorithm: α, β, and γ in the similarity function, and θ in the clustering algorithm. We tune the parameters heuristically. First, we set all parameters at the value 0.3. Then we increase or decrease the parameter value in increments of 0.05. After several rounds of tuning, we found the final α, β, γ, and θ were 0.35, 0.4, 0.25, and 0.3, respectively.

5.3 Results

Table 9 shows the accuracy of our methods on the two data sets for ambiguous queries, multifaceted queries, and mixed queries. The experimental results indicate that our approach performs quite well in the subtopic mining task. To understand the effect of each similarity sub-function, we conducted another experiment with the two datasets combined together and Table 10 reports the results. We can see that all three similarity sub-functions can achieve high precision, which indicates that the two phenomena described in Section 3 really exist. The recalls of the three functions are not so high, however, mainly due to the sparseness of the available data. By combining the uses of all the similarity sub-functions, we can significantly improve the recalls while maintaining high precision.

Table 9: Accuracy of subtopic mining for different types of queries

Data Set	Query Type	B-cubed Precision	B-cubed Recall	B-cubed F1
TREC	Ambiguous	0.918	0.932	0.925
	Multifaceted	0.893	0.899	0.896
DataSetA	Ambiguous	0.961	0.951	0.956
	Multifaceted	0.937	0.924	0.930
	Mixed	0.952	0.941	0.946

Table 10: Accuracy of subtopic mining with different similarity functions on the 'Combined' data set

Similarity Function	B-cubed Precision	B-cubed Recall	B-cubed F1
OSS phenomenon	0.945	0.681	0.791
SCAK phenomenon	0.915	0.778	0.841
URL string	0.976	0.517	0.676
All	0.928	0.922	0.925

The clustering accuracy in Table 10 differs from the accuracy in finding subtopics using the two rules in Section 3. There are two main reasons for this. First, different evaluations are made. Table 10 shows the accuracy of subtopic mining using clustering. Section 3 shows the accuracy of individual rules without using clustering. Second, different evaluation measures are utilized.

6. APPLICATIONS OF SUBTOPIC MINING

We consider two applications, search result clustering and re-ranking, and evaluate our method within the applications.

6.1 Search Result Clustering

Search result clustering is about grouping the URLs in the search result of a query by subtopics. Many methods have been proposed for search result clustering. Wang and Zhai's method [26] can be viewed as state-of-the-art and we take it as a baseline. We conduct comparisons in two settings: accuracy comparison and side by side comparison.

[3] http://trec.nist.gov/data/webmain.html

Table 11: Accuracy comparison between our method and baseline

Data set	Query Type	B-cubed Precision		B-cubed Recall		B-cubed F1	
		Our Method	Baseline	Our Method	Baseline	Our Method	Baseline
TREC Data	Ambiguous	0.926	**0.944**	**0.924**	0.792	**0.925**	0.861
	Multifaceted	0.864	**0.872**	**0.905**	0.808	**0.884**	0.839
DataSetA	Ambiguous	**0.963**	0.874	**0.935**	0.878	**0.949**	0.876
	Multifaceted	**0.936**	0.840	**0.913**	0.906	**0.927**	0.877
	Mixed	**0.952**	0.839	**0.933**	0.920	**0.942**	0.878
Improvement		+5.4%		+6.1%		+5.9% (p $\leq$ 4.58E-06 in the sign test)	

Table 12: Accuracy comparison from various perspectives

		Purity	Diversity	Coverage	Granularity	Average
Assessor 1	Baseline	2.88	2.35	2.52	2.73	2.62
	Our method	3.41	3.27	3.60	3.25	3.39
Assessor 2	Baseline	3.15	2.46	2.75	3.08	2.86
	Our method	3.60	3.29	4.27	3.44	3.65
Average	Baseline	3.02	2.41	2.64	2.91	**2.74**
	Our method	3.51	3.28	3.94	3.35	**3.52**

6.1.1 Our Method

Our method conducts query subtopic mining offline and stores the mined results in a database. Once the search results for a query are given by the search system, our method performs search result clustering online. Specifically, our method first refers to the database and finds the subtopics of the query, each of which may contain several URLs. For each search result, our method then takes the subtopics of the query as seed clusters. There might be some URLs which do not belong to any of the mined subtopics. We calculate cosine similarity between them and the URLs in the existing clusters by using the TFIDF of terms in titles and snippets. We then assign those URLs to the existing clusters or create new clusters on the basis of the calculated similarities and regard them as the final search result clusters.

6.1.2 Accuracy Comparison

We compared our method with the baseline method on ambiguous queries, multifaceted queries, and mixed queries, respectively. Table 11 shows the detailed comparison between our method and the baseline.

From the results, we can see that our method outperforms the baseline method. When combining the two datasets, the improvement of our method over the baseline is 5.4% in terms of B-cubed precision, 6.1% in terms of B-cubed recall, and 5.9% in terms of B-cubed F1 respectively.

6.1.3 Side-by-Side Comparison

We also compared our method with the baseline side-by-side. We created a new dataset referred to as DataSetB. DataSetB contains 50 queries and includes all three types of queries. First we applied both our method and the baseline to each query to obtain two sets of clustering results. Then, we showed the two results to two assessors to perform side by side evaluations. To avoid bias, we randomly swapped the sides of the two results. The assessors could not figure out which result was from which approach.

Several measures were used for evaluation of the results by each method: purity, diversity, coverage, and granularity. Purity is similar to precision and coverage is similar to recall. Diversity shows the difference between clusters. Granularity means how specific the cluster is. An ideal clustering should have high purity, high diversity, high coverage and fine granularity. We asked the assessors to give a score (1 to 5) for each measure.

Table 12 shows the average scores of the methods in terms of the measures for all 50 queries. The results show that our approach is much better than the baseline. The overall improvement is about 28%.

Table 13 shows the side-by-side evaluation results averaged over the 50 queries. When conducting an evaluation, the assessors made judgments on which result was better using three labels: ‘better than,’ ‘equally good,’ and ‘worse than.’ The results show that for about 60% of the queries, our method is better than the baseline; for about 21% of the queries, our method is equal to the baseline. We found that the clusters generated by our method are easier for humans to understand than the baseline.

Table 13: Side-by-Side Comparison

	Better than	Equal to	Worse than
Assessor 1	32	10	8
Assessor 2	30	11	9
Average	31	10.5	8.5

6.1.4 Discussion

When analyzing the results, we found the clusters generated by our approach are more accurate and natural (easy to understand) than the clusters generated by the baseline. This indicates that the clustering of URLs from a user's perspective (log mining based) can achieve better results than from a content creator's perspective (content based).

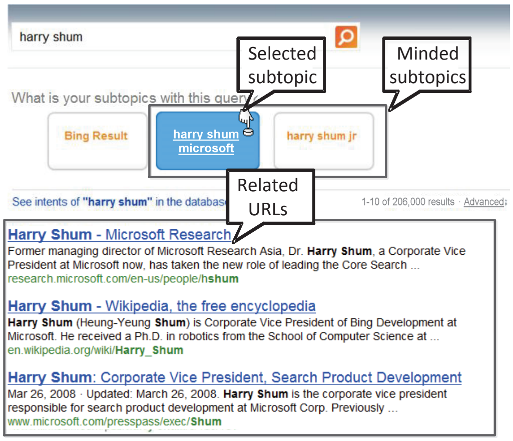
Our approach performs better than the baseline approach in terms of F1 measure for ambiguous, multifaceted, and mixed queries. The improvements of our approach on recall are larger than the improvements on precision. This is because our approach starts from the mined subtopics that cover the main subtopics of queries. On the contrary, the baseline has no such information, and it tends to generate larger clusters.

The improvement of our method over the baseline on the TREC data is not as high as those on DataSetA. Because in the TREC data, each subtopic consists of about 26.7 URLs while in DataSetA each subtopic only consists of about 6.8 URLs. In other words, DataSetA has finer grained subtopics than the TREC data. When the granularity of subtopics is coarser, our method can benefit less from log mining.

Here we show three examples of results given by the two methods. For simplicity, we only show the labels (subtopics) of URLs in each cluster. The first example is an ambiguous query, the second is a multifaceted query, and the third is a mixed query. We can see that our method often gives more specific clusters (subtopics) while the baseline tends to produce more general clusters (subtoipcs).

Table 14: Examples of comparison between our approach and baseline

Query	Our method	Baseline
Black swan (Ambiguous)	C1: black swan movie	C1: black swan movie
	C2: black swan diet	black swan reviews
	C3: black swan inn	C2: black swan yoga
	C4: black swan yoga	black swan theory
	C5: black swan wine	C3: black swan reviews
Video game (Multifaceted)	C1: video game reviews	C1: video game reviews
	C2: video game cheats	video game cheats
	C3: video game stores	video game stores
	C4: video game addiction	C2: video game wiki
	C5: video game trailers	video game history
Manchester (Mixed)	C1: Manchester united	C1: Manchester England
	C2: Manchester college	Manchester tourism
	C3: Manchester history	Manchester news
	C4: Manchester news	C2: Manchester united
	C5: Manchester tourism	Manchester soccer


harry shum
Selected subtopic
Minded subtopics
What is your subtopics with this query
Bing Result
harry shum microsoft
harry shum jr
Related URLs
See intents of "harry shum" in the databas
1-10 of 206,000 results · Advanced

Harry Shum - Microsoft Research
Former managing director of Microsoft Research Asia, Dr. **Harry Shum**, a Corporate Vice President at Microsoft now, has taken the new role of leading the Core Search ...
research.microsoft.com/en-us/people/h**shum**

Harry Shum - Wikipedia, the free encyclopedia
Harry Shum (Heung-Yeung **Shum**) is Corporate Vice President of Bing Development at Microsoft. He received a Ph.D. in robotics from the School of Computer Science at ...
en.wikipedia.org/wiki/**Harry_Shum**

Harry Shum: Corporate Vice President, Search Product Development
Mar 26, 2008 · Updated: March 26, 2008. **Harry Shum** is the corporate vice president responsible for search product development at Microsoft Corp. Previously ...
www.microsoft.com/presspass/exec/**Shum**

Figure 8: Example of search result re-ranking.

6.2 Search Result Re-Ranking

We consider another search task for the evaluation of our subtopic mining method, namely, search result re-ranking. We chose to do so because we can easily use click-through data to conduct quantitative evaluation.

6.2.1 *Our Method*

When the search system provides the results to the user, there might be multiple ways to present it if the query contains multiple subtopics, i.e., it is ambiguous or multifaceted. These include simple ranking, clustering of URLs by subtopics, dynamic ranking of URLs belonging to the same topic, as well as re-ranking of URLs by subtopics.

Re-ranking is conducted in the following way. The user is first asked which subtopic she is interested in, with the subtopics shown at the top of the results page. When the user selects a subtopic, the URLs belonging to the subtopic will be moved to the top (re-ranked). The relative order between URLs inside and outside of the subtopic will be kept. The subtopics are assumed to be represented by keywords of subtopics mined with our method.

Figure 8 shows an example UI in which re-ranking is performed for the query 'harry shum.' The second subtopic of the query is selected and all the URLs belonging to it are re-ranked at the top. This will significantly improve a user's experience compared with the conventional approach (Figure 1). Although re-ranking is a simple, interactive approach to search results presentation (cf., [28]), there seems to be no study before on the approach to the best of our knowledge. Note that re-ranking can only be done for head queries, because it is based on log data mining, which is a shortcoming that any data mining method may suffer from.

6.2.2 *Evaluation*

We collected search log data of 20,000 randomly selected searches at the commercial search engine, for which the query has at least two subtopics mined by our method. The search log data contains the ranking results of URLs as well as the clicks on URLs in the searches. We used the data to estimate how much the cost of users' searches can be saved by re-ranking based on our subtopic mining method.

We first calculated the average position of last clicked URLs in the 20,000 searches. The result is 3.41. The number indicates the average cost of finding information in the conventional search UI based on ranking.

Next, we used the same log data to estimate the cost for users based on re-ranking. We assumed that the cost for the user to check the subtopics and click one of them is 1.0, because the subtopics are shown in one line at the top, and examining them is almost equivalent to examining the title and snippet of a URL. The cost of finding the last click within each selected subtopic can be calculated by using the search log data, because within the subtopic the ranking order is preserved. Note that because of the one subtopic per search phenomenon, the clicked URLs in a search usually belong to the same subtopic. We calculated the average position of last clicked URLs belonging to the same subtopics in the 20,000 searches and found that it is 1.80.

The cost saved in the re-ranking may be calculated as

$$\Delta = 3.41 - 1.80 - 1 = 0.61$$

This is the result for the saved cost per search. Similarly, we can estimate the saved cost per query, which is 0.7. We can conclude, therefore, that the re-ranking method can reduce a user's effort in finding information, and thus our method of subtopic mining is effective.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied the problem of query subtopic mining. Specifically, we have discovered two phenomena of user search behavior that can be used as signals to mine major senses and facets of ambiguous and multifaceted queries. One is called one subtopic per search and the other subtopic clarification by additional keyword. We have developed a clustering algorithm that can effectively and efficiently mine query subtopics on the basis of the two phenomena. We have evaluated the effectiveness of the proposed approach. On the TREC dataset, the F1-measure is 0.925 for finding ambiguous subtopics and 0.896 for finding multifaceted subtopics. When applied to search result clustering, our method can improve precision by 5.4% and recall by 6.1%. The effectiveness of our method has also been verified in search result re-ranking, using click data.

There are several issues we want to further explore to enhance our current work. First, only three types of features were used in the current clustering algorithm. We plan to investigate the use of other features to further improve the accuracy. Second, we only tried one clustering algorithm as an example. Other existing algorithms can be applied as well. Third, the mined subtopics by our method were only applied to two applications, and they can be useful in other applications as well. We also plan to try some other applications such as personalized search and search result diversification. Finally, our method can only be employed when there is enough search log data, which is also a drawback for most log mining algorithms. How to apply the approach in tail queries is also an issue we need to consider.

8. ACKNOWLEDGEMENTS

This research was done when the second author was visiting MSRA, her research was supported by the SPKLSTN Lab (National Science Foundation of China under Grant No. 60825202, 61173112, and Cheung Kong Scholar's Program).

Jian Pei's research is supported in part by an NSERC Discovery Grant, a BCFRST NRAS Endowment Research Team Program project, and a GRAND NCE project. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

9. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of SIGIR'06*, pages 19–26, 2006.

[2] O. Alonso and M. Gertz. Clustering of search results using temporal attributes. In *Proceedings of SIGIR'06*, 2006.

[3] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12:461–486, August 2009.

[4] D. Beeferman and A. L. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of SIGKDD'00*, pages 407–416, 2000.

[5] S. Beitzel, E. Jensen, A. Chowdhury, and O. Frieder. Varying approaches to topical web query classification. In *Proceedings of SIGIR'07*, pages 783–784. ACM, 2007.

[6] C. Brandt, T. Joachims, Y. Yue, and J. Bank. Dynamic ranked retrieval. In *Proceedings of WSDM'11*, pages 247–256, 2011.

[7] A. Z. Broder. A taxonomy of web search. *Sigir Forum*, 36:3–10, 2002.

[8] A. Z. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *Proceedings of SIGIR'07*, pages 231–238, 2007.

[9] M. Burt and C. L. Liew. Searching with clustering: An investigation into the effects on users' search experience and satisfaction. *Online Information Review*, 36, 2012.

[10] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proceeding of KDD'08*, 2008.

[11] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17, 2009.

[12] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of WWW'09*, pages 1–10, 2009.

[13] H. Chen and S. Dumais. Bringing order to the web: automatically categorizing search results. In *Proceedings of CHI'00*, pages 145–152, 2000.

[14] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *Proceedings of TREC'09*, pages 1–9, 2009.

[15] N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of SIGIR'07*, pages 239–246, 2007.

[16] H. Deng, I. King, and M. Lyu. Entropy-biased models for query representation on the click graph. In *Proceedings of SIGIR'09*, pages 339–346. ACM, 2009.

[17] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. *Software: Practice and Experience*, 38(2):189–225, 2008.

[18] S. Fujita, K. Machinaga, and G. Dupret. Click-graph modeling for facet attribute estimation of web search queries. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 190–197, 2010.

[19] W. Gale, K. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics, 1992.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of KDD'02*, pages 133–142, 2002.

[21] R. Jones and K. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *Proceedings of CIKM'08*, pages 699–708, 2008.

[22] M. Kan and H. Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM, 2005.

[23] Y. Li, Z. Zheng, and H. K. Dai. Kdd cup-2005 report: facing a great challenge. *SIGKDD Explor. Newsl.*, 7:91–99, December 2005.

[24] H.-T. Pu, S.-L. Chuang, and C. Yang. Subject categorization of query terms for exploring web users' search interests. *J. Am. Soc. Inf. Sci. Technol.*, 53:617–630, August 2002.

[25] F. Radlinski, M. Szummer, and N. Craswell. Inferring query intent from reformulations and clicks. In *Proceedings of WWW'10*, pages 1171–1172, 2010.

[26] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *Proceedings SIGIR'07*, pages 87–94, 2007.

[27] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of WWW'01*, pages 162–168, 2001.

[28] X. Yuan and N. J. Belkin. Supporting multiple information-seeking strategies in a single system framework. In *Proceedings of SIGIR'07*, pages 247–254, 2007.

[29] H. Zeng, Q. He, Z. Chen, W. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of SIGIR'04*, pages 210–217. ACM, 2004.