Multi-dimensional Search Result Diversification*

Zhicheng Dou[1], Sha Hu[2,4], Kun Chen[3,4], Ruihua Song[1], and Ji-Rong Wen[1]
[1]Microsoft Research Asia; [2]Renmin University of China; [3]Xi'an Jiaotong University
[1]{zhichdou, rsong, jrwen}@microsoft.com; [4]{sallyshahu,cs.kunchen}@gmail.com

ABSTRACT

Most existing search result diversification algorithms diversify search results in terms of a specific dimension. In this paper, we argue that search results should be diversified in a multi-dimensional way, as queries are usually ambiguous at different levels and dimensions. We first explore mining subtopics from four types of data sources, including anchor texts, query logs, search result clusters, and web sites. Then we propose a general framework that explicitly diversifies search results based on **multiple** dimensions of subtopics. It balances the relevance of documents with respect to the query and the novelty of documents by measuring the coverage of subtopics. Experimental results on the TREC 2009 Web track dataset indicate that combining multiple types of subtopics do help better understand user intents. By incorporating multiple types of subtopics, our models improve the diversity of search results over the sole use of one of them, and outperform two state-of-the-art models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Algorithms, Experimentation

Keywords

Search Result Diversity, Multi-dimensional, Subtopic, Intent

1. INTRODUCTION

Studies show that the vast majority of queries to search engines are short and vague in specifying a user's intent [12, 23, 10, 20, 18]. Search result diversification is one way to solve the problem when there is uncertainty in the information needs for these queries. It aims to provide a list of results that cover as many aspects as possible, so that most users can be satisfied by the top results.

Although many algorithms have been proposed to diversify search results, no consensus has been reached on the diversity concept. Most previous work [2, 35, 36, 1] diversifies search results from a specific perspective and deals with all queries in the same manner. For example, the MMR algorithm is based on document content, and the IASelect algorithm [1] relies on the topical categories of documents.

We argue that diversification is a multi-dimensional problem, as people expect different things by diversity from different angles for different queries. In this paper, a dimension corresponds to a data source. It provides the information to differentiate the aspects of a query or the related documents. We explore subtopics from four types of dimensions: anchor texts, query logs, clusters of search results, and the web sites of search results. They view the uncertainty of a query from different perspectives. Query logs reflect the popular requirements of real-world users, whereas anchor texts give an overview of the possible meanings of a query that is less biased by users and search engines. Search result clusters group some near-duplicated documents together, whereas web sites are used to identify the documents with different functionalities.

One dimension is not enough for satisfying the various requirements from different users. Query logs are not available for new queries and they have bias toward background rankings. Anchor texts can conquer these shortcomings instead. Query logs and anchor texts are applicable for short and popular queries; whereas, search result clusters and web sites work for both popular and tail queries. The first three types of subtopics (query logs, anchor texts, and search result clusters) can reflect the topicality of the query; while the fourth dimension, web sites of search results, can reflect the functionality aspect of queries. For instance, web pages from Wikipedia are usually good for information seeking queries, while pages on a particular software website are suitable for downloading tasks. As different types of subtopics are complimentary to each other, and can benefit different types of queries, combining them together can potentially help search result diversity. Experimental results in Section 6.2 confirm that combining subtopics from multiple dimensions does help discover more user intents.

Furthermore, user intents vary in different levels, even for a same query. For example, the query "defender" has multiple interpretations. It can stand for "Windows Defender", an

*The work was done when the second and third author were visiting Microsoft Research Asia

WSDM'11, February 9–12, 2011, Hong Kong, China.
Copyright 2011 ACM 978-1-4503-0493-1/11/02 ...$10.00.

anti-virus computer software, or “Land Rover Defender”, an off-road utility vehicle. Whereas, for the specific interpretation “Windows Defender”, users are also interested in various aspects covered by the topic, such as finding the homepage of the software, downloading the software, or looking for user reviews about the problems with the software. Combining subtopics from multiple dimensions is helpful to view documents from different perspectives, and hence is able to diversify results better than using one specific dimension.

We propose a general framework of diversifying search results based on multiple dimensions of subtopics. The framework combines the results of the original query and those of mined subtopics to get comprehensive subtopic information for documents. It diversifies search results by considering both relevance of documents and novelty or richness of their subtopics. A greedy algorithm is used to iteratively select the next best document to generate a diversified ranking list. We implement two systematic approaches, a topic richness model and a topic novelty model, based on this framework. Experimental results on the ClueWeb09 web page collection [7] and the TREC 2009 Web track query set [6] show that by using the subtopics, our proposed result diversification algorithms improve diversity over the sole use of one category of subtopics. The models also outperform two state-of-the-art models, the MMR model [2] based on document content, and the IASelect model [1] based on topical categories.

The remainder of this paper is organized as follows. We briefly discuss related work in Section 2, and describe the dataset in Section 3. We introduce our methods of mining subtopics in Section 4. Following this, the search result diversification framework is proposed in Section 5, and the experimental results are analyzed in Section 6. We then conclude our work in Section 7.

2. RELATED WORK

The problem of improving search result diversity has been well discussed in IDR 2009 [16], and there has been much work on this area. Carboness and Goldstein [2] proposed the MMR algorithm, which diversifies search results or document summaries based on content similarities between documents or sentences. Zhai and Lafferty described a general risk minimization framework for subtopic retrieval, in which relevance and novelty can be modeled together within a loss function [33]. Yue et al. [31] formulate the learning problem of predicting diverse subsets and derive a discriminative training method based on structural SVMs. Zhang et al. [35] proposed the Affinity Ranking (AR) algorithm to rerank search results by optimizing diversity and information richness of search results. Zhu et al. [36] introduced a novel ranking algorithm called Grasshopper, which ranks items with an emphasis on diversity based on random walks in an absorbing Markov chain. Recently, Agrawal et al. [1] proposed a diversification objective function that maximizes the likelihood of ranking relevant documents in the top results based on topical categories of queries and documents, and used a greedy algorithm named IASelect for the objective. Coyle and Smyth [9] investigated the use of case-based reasoning (CBR) diversity-enhancing techniques in Web search, showing that result diversity can be significantly enhanced without compromising result precision and recall. Santos et al [20, 21] proposed the xQuAD framework which models an ambiguous query as a set of sub-queries, and diversified search results based on query reformulations from Web search engines. Rafiei et al. [18] modeled the problem as expectation maximization and presented algorithms to estimate the optimization parameters.

Most of the above work focuses on specific diversity models. They usually do not explicitly mine subtopics, or simply use one specific type of subtopics (e.g., document content in [2], topical categories in [1]) to measure the similarity of documents. In this paper, we explicitly mine subtopics from different data sources to better predict user intents, and design result diversification models based on explicit multiple types of subtopics. A relevant work is [14], in which the authors discussed the multi-dimensional diversity problem in image search; whereas we focus on general web search.

The problem of intrinsic diversity is also receiving much attention in IR community [16]. Intrinsic diversity means that diversity is a part of the information need in some cases (either in web search, or in other applications such as recommendation systems). Users prefer to get a list of diverse results instead of a large number of similar or duplicated results. Related applications include but not limit to the approaches as follows. Ziegler et al. [37] proposed diversifying recommendation lists which cover more user interests to improve user satisfaction, rather than specifically focusing on the accuracy of individual recommendations. Celma and Herrera [3] proposed new approaches to evaluate novel recommendations. Vee et al. studied the problem of result diversification in online shopping applications. They developed query processing techniques that guarantee diversity [28]. Strohmaier et al. addressed the problem of the diversity of query suggestions [27]. Radlinski and Dumais proposed to improve personalized web search by mining diverse related queries from query logs [17]. El-Arini et al. [11] presented an approach for picking a set of diverse posts that best covers the important stories in the blogosphere. Song et al. [24] presented a re-ranking method based on topic richness analysis to enrich topic coverage in retrieval image search results. In this paper, we mainly discuss the problem of diversifying web search results. Our proposed models, which diversify results based on associated multiple types of attributes, may also be applicable for the above scenarios.

Several metrics have been proposed to evaluate result diversity. Zhai, Cohen and Lafferty [34] proposed S-recall, S-precision, and WS-precision for evaluating subtopic retrieval. Clarke et al. [4] proposed α-nDCG in which information needs and documents are treated as sets of nuggets. Agrawal et al. [1] proposed several Intent-Aware (IA) metrics including MAP-IA, nDCG-IA, and Precision-IA. Clarke, Kolla and Vechtomova proposed Novelty and Rank-Biased Precision (NRBP) [5] which also measures the novelty and diversity of search results. Sakai et al. [19] propose metrics called Idiv-nDCG and Idiv-Q for evaluating diversified search results. These metrics favor the documents that are highly relevant to more popular intents. Details of these evaluation metrics are not the main focus of this paper. In this paper, we use the α-nDCG metric, which is the primary measurement used in diversity task of TREC 2009 Web track, to evaluate the effectiveness of the proposed result models.

There have been many approaches on mining topics for search result organization, which are also relevant to our work. For example, Lawrie et al. [13] proposed to automatically generate topic hierarchies by applying a graph-theoretic algorithm. In this paper, we preliminary mine subtopics from four data sources and combine them together to diversify search results. We try to investigate whether combining multiple types of subtopics can better predict user intents and improve result diversity. We plan to integrate existing topic mining algorithms into our framework, and detailed analysis of these algorithms is beyond the scope of the paper.

3. DATASETS AND SETTINGS

Before presenting our methods of mining subtopics and diversifying search results, we first briefly introduce the dataset in this section.

We use the public available ClueWeb09 data collection [7] to experiment with our algorithms. The collection consists of one billion web pages in ten languages, collected in January and February 2009. We use WebStudio [30], a flexible indexing and ranking system, to index all 500M English pages using 40 servers. Each server has two 2.50 GHz Xeon CPUs, 16G memory, and a 3TB hard disk. Given a query, top search results are first retrieved from each server, and then merged in an aggregation server.

Our proposed models are evaluated using the query set of TREC 2009 Web track [6]. It includes 50 queries, each of which has three to eight manually edited subtopics. There are 243 subtopics in total, and 199 of them have at least one judged relevant document. To the best of our knowledge, it is the first public query set with explicit diversity-aware relevance judgments.

We use the query log data that include all queries issued to Bing Search [15] in April 2009. We process the log data and group queries into query sessions. A query session includes a sequence of queries issued by one user within a period of search activity terminated by 30 minutes or more of inactivity.

As some of our subtopics mining methods (clusters and web sites of top results in Section 4) and proposed search result diversification models (in Section 5) need an initial set of search results, we implement the MSRA2000 model [25] as our baseline ranking function. The model is an augmented BM25 function that combines four different fields of Web pages including title, body, URL, and anchor text. In addition to term frequency, inverted document frequency, and document length, it further considers the proximity between query term occurrences. Detailed descriptions of the model can be found in [25]. In the ad-hoc task of TREC 2009 Web track, the ranking function (named MSRANORM in [6]) generates reasonably good results [6].

4. SUBTOPIC MINING

We mine multiple types of subtopics that may match potential user information needs or intents. We conjecture that different types of data sources contain complementary information from different perspectives. Leveraging the subtopics from these data sources may help to better understand user intents. In this paper, we extract subtopics from anchor texts, query logs, clusters of search results, and their corresponding web sites. We will investigate other data sources, such as online dictionaries, which include the official interpretations, in future work.

For each subtopic c, we estimate a weight $w(q, c)$ to measure how important the subtopic is for a given query q. A subtopic with a higher weight potentially infers that more likely users are searching for the subtopic by issuing the query. In the following subsections, we will introduce our methods of mining subtopics and corresponding weights.

4.1 Anchor texts

Anchor texts created by web designers provide meaningful descriptions of destination documents. They are usually short and descriptive, which share the similar characteristics with web queries. Give a query, anchor texts that contain the query terms usually convey the information about the query intents, hence we use these kinds of related anchor texts as subtopics.

For a given query q, we first get all anchor texts containing all query terms of q, weight them, and select the most important ones as subtopics. We observe that the importance of an anchor text is usually proportional to its popularity on the Web, i.e., how many times it is used in web sites or pages. However, a shorter anchor text usually matches the query better than a longer anchor text. The subtopic of the longer anchor text may be over-specified or drifted from the original query. Based on these observations, we design the following ranking function to evaluate the importance of an anchor text c:

$$\begin{aligned} f(q,c) &= \text{freq}(c) * \text{rel}(q,c) \\ &= [\text{nsite}_c + \log(\text{npage}_c - \text{nsite}_c + 1)] * \frac{1 + \text{len}(q)}{\text{len}(c)} \end{aligned}$$

The first term $\text{freq}(c) = \text{nsite}_c + \log(\text{npage}_c - \text{nsite}_c + 1)$ evaluates the popularity of anchor text c, in which npage_c denotes the number of source pages that contain the anchor text c, and nsite_c denotes the number of unique source sites of these links. As it is easy to create a large number of source pages within a same source site to boost the anchor text, we just count each source site once. For the additional pages containing the anchor text (totally $\text{npage}_c - \text{nsite}_c$ pages) from these sites, we discount their votes using the log function. Obviously an anchor text used by a larger number of different web sites will get a high value of $\text{freq}(c)$.

The second term $\text{rel}(q,c) = \frac{1+\text{len}(q)}{\text{len}(c)}$ punishes the anchor texts that contain too many words. Note that $\text{len}(q)$ is the count of query terms, and $\text{len}(c)$ is the number of terms contained in c. For the query q, an anchor text $q + t_1$ with an additional term t_1 gets as high $\text{rel}(q,c)$ as one, because it is a perfect subtopic of the query; whereas, another one $q+t_1+t_2$ containing two additional terms gets lower $\text{rel}(q,c)$.

As $f(q,c)$ is unbounded, we further normalize it to (0, 1) using the sigmoid function:

$$w(q,c) = \frac{1}{1 + e^{-[f(q,c) - avgFreq]/avgFreq}}$$

where $avgFreq$ is the average of anchor text frequency $\text{freq}(c)$ over all anchor texts in the corpus. An anchor text with average frequency and one additional term will get the average score 0.5.

To efficiently retrieve anchor texts containing all query terms, we use the WebStudio [30] system to index all anchor texts. We treat each anchor text c as a document, and build its statistics (such as nsite_c and npage_c) as its attributes. We use $f(q,c)$ as ranking function to obtain top anchor texts for the query q. Using the configuration introduced in Section 3, we can retrieve top 100 anchor texts for a query within one second.

Table 1: Subtopics for query "defender" mined from different data sources

(a) anchor texts

Subtopic	Weight	Subtopic	Weight
castle defender	0.999	reputation defender	0.822
public defender	0.997	star defender	0.784
cosmic defender	0.979	chicago defender	0.724
windows defender	0.962	base defender	0.637
brewery defender	0.935	doodle defender	0.593

(b) query logs

Subtopic	Weight	Subtopic	Weight
windows defender download	0.458	defender marine supply	0.270
defender arcade game	0.273	install microsoft defender	0.270
defender antivirus	0.273	defender for xp	0.270
land rover defender	0.270	microsoft defender review	0.270
free windows defender beta	0.270	defender pro	0.210

(c) search result clusters

Subtopic	Weight	Subtopic	Weight
office	1.000	land rover	0.264
national,center,juvenile	0.850	otterbox,description	0.228
free encyclopedia	0.471	appellate,osad	0.233
wikipedia,redirected,video	0.421	federal public	0.113
arcade game	0.320	dedicated,life	0.092

(d) sites of search results

Subtopic	Weight	Subtopic	Weight
en.wikipedia.org	1.000	www.pixelparadox.com	0.500
www.otterbox.com	0.997	www.state.co.us	0.500
www.state.il.us	0.731	www.nlada.org	0.500
www.bigfuntown.com	0.731	www.sdcounty.ca.gov	0.500
www.wn.com	0.731	expertisegames.com	0.500

Table 1(a) shows the top 10 anchor texts with their weights for the query "defender" mined from the ClueWeb09 collection [7]. Note that we do not diversify retrieved anchor texts in this paper, because the top mined anchor texts look quite different from each other.

4.2 Query logs

Query log data contain much useful information about user intents, as queries are directly issued by real-world users. When a user issues the query that may be ambiguous or underspecified, and does not get expected results, he/she often refines the query and re-submits a new query to search engines. So by analyzing the follow-up queries in sessions, we can identify user intents covered by the original query.

Suppose for each query q_i, n_i is the number of times the query was issued. For a pair of queries (q_i, q_j), let n_{ij} be the number of times q_i was followed by q_j. $p_{ij} = \frac{n_{ij}}{n_i}$ is the empirical probability of q_i being followed by q_j. The problem of directly using the empirical follow-up probability p_{ij} is that follow-up queries are usually dominated by top user intents. For example, top three follow-up queries for query "defender" are "windows defender download," "microsoft defender," and "windows defender" according to the log data introduced in Section 3. These queries are actually talking about the same interpretation related to "windows defender". To avoid this problem, we use an MMR-like [2] measure to greedily select the set of queries that are related to the given query yet different from each other. Suppose $R(q_i)$ is the set of queries already selected, the next best query, namely q^n, is selected by:

$$q^n = \arg\max_{q_j} \left[\lambda \cdot p_{ij} - (1-\lambda) \cdot \max_{q_k \in R(q_i)} sim(q_j, q_k) \right] \quad (1)$$

where λ=0.5, and $sim(q_j, q_k)$ is the similarity between two queries q_j and q_k. We assume that the two queries q_j and q_k are similar if:

- q_j and q_k are frequently co-issued in the same query sessions. We use the measurement $p^*_{jk} = \sqrt{p_{jk} p_{kj}}$ proposed by Radlinski and Dumais [17] to evaluate the probability of two queries being issued together in the same query sessions. A high p^*_{jk} value means that q_j and q_k are frequently issued in the same sessions.
- The results by searching q_j and q_k are similar; Suppose $Docs(q_j)$ and $Docs(q_k)$ are top ten search results returned for query q_j and q_k. We use $\frac{|Docs(q_j) \cap Docs(q_k)|}{|Docs(q_j) \cup Docs(q_k)|}$ to evaluate the result similarity of these two queries.
- The words contained in q_j and q_k are similar. We use $\frac{|q_j \cap q_k|}{|q_j \cup q_k|}$ to measure the text similarity between these two queries.

We use the linear combination of these factors as follows.

$$sim(q_j, q_k) = \frac{1}{3} \left\{ p^*_{jk} + \frac{|Docs(q_j) \cap Docs(q_k)|}{|Docs(q_j) \cup Docs(q_k)|} + \frac{|q_j \cap q_k|}{|q_j \cup q_k|} \right\}$$

When the top subtopics are selected, we use the following sigmoid function to calculate the normalized weights for selected subtopics (queries):

$$w(q, c) = 1/\left(1 + e^{n*(avgProb - p_{qc})}\right)$$

where p_{qc} is the empirical probability of q being followed by c. The subtopic c with a higher follow-up ratio will get a larger weight $w(q, c)$. $avgProb$ is the estimation of average follow-up ratio between all queries and their top follow-up queries. We empirically set it as 0.1 in this paper. n is a normalization constant and we set it to 10.

Similar to the subtopics mined from anchor texts, we also require that a subtopic coming from query logs must contain all query terms appearing in the query, or it is the abbreviation of the follow-up query. As an example, we show top 10 subtopics for the query "defender" in Table 1(b).

4.3 Search results clusters

Search result clustering is an alternative approach used to solve the problem of query ambiguity. Instead of diversifying a search result list, it groups search results into clusters, so that users can easily navigate into a particular group that is relevant to his/her information need. Existing search result clustering approaches include but are not limited to [32, 29, 8]. In this paper, we use the algorithm presented by Zeng et al. [32] to group the top N original search results into K clusters based on key phrases in snippets. The documents that contain the same top-ranked phrases (n-grams) are grouped together as one cluster, and the cluster that contains more salient phrases will be ranked higher. In this paper, we treat each cluster as an implicit subtopic. As we do not have the original ranking score of each cluster, we assume a cluster (subtopic), denoted by $cluster_1$, is more important than another cluster, denoted by $cluster_2$, if: (1) $cluster_1$ is ranked higher than $cluster_2$ in terms of salient phrases; and (2) the best document within the cluster $cluster_1$ is ranked higher than that in $cluster_2$. We employ the following equation based on the above two assumptions to evaluate the importance of a cluster subtopic:

$$w(q,c) = 0.5 \times \frac{K - \text{clstRank}_c + 1}{K} + 0.5 \times \frac{1}{\text{bestDocRank}_c}$$

where clstRank_c is the rank of the cluster among all clusters, and bestDocRank_c is the highest rank of the documents within the cluster, i.e., $\text{bestDocRank}_c = \min_{d \in c} \text{rank}_d$. We use the same settings N=200 and K=10 as those in [32]. We show the cluster names for the query "defender" in Table 1(c). The descriptions of the subtopics (clusters) are generated based on selected key phrases in the snippets of top results. Note that the descriptions are not used in our diversification framework, as we can directly get the association between clusters and documents.

4.4 Sites of search results

Web pages from the same web site usually contain similar information, while different websites are more likely to provide results with different functionalities. For instance, pages from Wikipedia are usually good for information seeking, while pages on a particular software hosting website are suitable for downloading tasks. It might be better to mix the results from multiple web sites in the top results.

We regard each web site within top N results as an implicit subtopic, and calculate its normalized importance by the sigmoid function:

$$w(q,c) = 1/(1 + e^{\omega - \text{nPagesInSite}_c})$$

Here nPagesInSite_c is the number of results that come from the site c and are returned at the top N. We assume that the more results a site returns, the richer information the site provides. A site gets a medium weight 0.5 when there are ω results from this site. We empirically set N=200 and ω=2 on the ClueWeb09 dataset, and show the top sites for the query "defender" in Table 1(d). Note that web site `en.wikipedia.org` instead of `microsoft.com` has the highest weight because Wikipedia has many entry pages for different meanings of "defender", whereas the Windows Defender's home page and other relevant pages from `microsoft.com` are not in the ClueWeb09 dataset.

4.5 Discussions

In this section, we briefly discuss the characteristics of each type of subtopics and their differences.

Query log-based subtopics can somehow reflect real-world user information needs, but they have the flaws as follows. Firstly, they are only available for old queries. Secondly, they may show some bias toward background rankings. If most subtopics are already retrieved in the top results, users may just click them without issuing a new query, and hence some major subtopics cannot be found by analyzing query sessions. Thirdly, subtopics may be dominated by some popular user intents. For example, for the query "defender", most of the mined subtopics are about "windows defender", while "football defender" is rarely found in user sessions.

Table 2: Subtopics for an example page http://en.wikipedia.org/wiki/Windows_Defender

C	c	w_c	$r(c,d)$
Anchor	windows defender	0.962	0.99
Clustering	free encyclopedia	0.471	0.99
Clustering	wikipedia,redirected,video	0.421	0.99
Site	en.wikipedia.org	1.000	0.70
Query log	windows defender download	0.458	0.50
Query log	free windows defender beta	0.270	0.99
Query log	defender for xp	0.270	0.30
Query log	microsoft defender review	0.270	0.44

Anchor text-based subtopics also favor short and popular queries. Different from query logs, they reflect the distribution of information from the viewpoints of web publishers. They may contain the information that has not been covered by user queries yet, and they are not biased by past users behaviors and background rankings. Anchor texts are also easy to collect, without any problem of user privacy issue.

Different from query log and anchor text-based subtopics, search result cluster-based subtopics can work for both popular and tail queries. They are especially useful to identify duplicated information in the top results in terms of document content. As subtopics are mined from the original search results and no new documents are considered, their effectiveness depends on the quality of the background rankings.

Web sites of search results can also work for tail queries. Different from the subtopics mined from the first three dimensions (query logs, anchor texts, and search result clusters) that reflect the topicality of queries, subtopics mined from web sites of search results can be used to diversify results from the functionality aspect of queries.

As these different types of subtopics are diverse in many aspects and benefit different types of queries, combining them together can potentially help multi-dimensional search result diversity.

5. SEARCH RESULT DIVERSIFICATION

In this section, we present a general framework for diversifying search results based on subtopics, and then describe two different implementations in the framework in Section 5.2 and Section 5.3.

5.1 Framework

We use a general framework to solve the problem of search result diversity based on subtopics. Assume that $\mathbb{C}$ is the set of all subtopics categories and $C \in \mathbb{C}$ (e.g., anchor text) is one of the categories. $c \in C$ is a subtopic within category C.

We first generate a list of search results for the original query and also a result list for each subtopic c using the baseline ranking function described in Section 3. More specifically, we use each anchor text-based subtopic (or each query log-based subtopic) as a new query and retrieve the top results for it. For each result clustering-based and site-based subtopic, we simply keep the associations between documents and the subtopic and do not carry out new retrieval. We use $r(q, d)$ and $r(c, d)$ to represent the normalized relevance score of document d, ranging from 0 to 1, with respect to query q and subtopic c. Different ranking methods may generate different distributions of ranking scores. To avoid heterogeneous score combination, we utilize the ranks instead of the original relevance scores. Suppose rank(q, d) is the rank of document d in ranking list of q, we let $r(q, d) = 1/\sqrt{\text{rank}(q, d)}$ after experimenting with several other alternatives; and so does $r(c, d)$. Note that $r(q, d)$ and $r(c, d)$ will decrease as the document ranks increase.

Then we represent document d with a list of subtopics that are associated with the document. For example, Table 2 shows all the subtopics mined for web page `http://en.wikipedia.org/wiki/Windows_Defender`. The page has eight subtopics that are from all four subtopic categories.

A result diversification model aims to diversify the results in R, which is the set of candidate documents retrieved by the query q or its subtopics, in terms of their subtopic information. We assume that a good diversified set should cover as many subtopics as possible in multiple dimensions, and at the same time the relevance of results should be preserved. We employ a greedy algorithm that can match our objectives to iteratively select the next best document from the remaining documents to generate a diversified ranking list:

$$d_{n+1} = \arg \max_{d \in R \setminus S_n} [\rho \cdot r(q, d) + (1 - \rho) \cdot \Phi(d, S_n, \mathbb{C})] \quad (2)$$

where Φ is the importance of the document d in terms of topic diversity. ρ is the parameter that controls the tradeoff between relevance and topic diversity. Documents will be ranked totally based on diversity when ρ equals to 0, whereas it will approximate to the original ranking when ρ equals to 1.

The framework can be regarded as a general form of the xQuAD framework [20], the MMR model [2], and the IASelect model [1]. It models the diversity of a document d given a set of selected documents S_n. The difference is that $\Phi(d, S_n, \mathbb{C})$ accepts multiple dimensions of subtopics $\mathbb{C}$, rather than a single dimension of subtopics.

The key problem is how to measure the importance of a document in terms of topic diversity over multiple dimensions of subtopics. We explore two models in the following two sections. Note that the weight $w(q, c)$ of a subtopic c to a query q is simplified as w_c in the following sections.

5.2 Topic richness model

Our first approach treats $\Phi(d, S_n, \mathbb{C})$ as a topic richness score, which is similar to the xQuAD framework [20] and the IASelect model [1]. We define the following function to measure the overall topic richness of a document d under the condition that a set of documents S_n have already been selected:

$$\Phi(d, S_n, \mathbb{C}) = \sum_{C \in \mathbb{C}} \mu(C) \cdot v(d, S_n, C) \quad (3)$$

Here we assume the independence between each type of subtopics as the subtopics are mined from four totally different data sources. We use a linear combination and $\mu(C)$ is the importance of subtopic category C among all types of subtopics, and we let $\sum_{C \in \mathbb{C}} \mu(C) = 1$. To simplify the problem, we set an equal weight $\mu(C) = \frac{1}{|\mathbb{C}|}$ for each subtopic category in this paper and will investigate other weighting schemes in future work. $v(d, S_n, C)$ is the subtopic richness score of document d in terms of subtopic category C given a set of selected documents S_n; and we define

$$v(d, S_n, C) = \sum_{c \in C} w_c \cdot \phi(c, S_n) \cdot r(c, d) \quad (4)$$

where w_c is the weight of subtopic c in subtopic category C. $\phi(c, S_n)$ is the discounted importance of subtopic c after documents set S_n has been selected. We assume that the importance of a subtopic should be reduced if previously selected documents have already covered it. Given a subtopic, suppose that documents are independent, we use the following function, which is similar to that in [1], to calculate $\phi(c, S_n)$:

$$\phi(c, S_n) = \begin{cases} 1 & \text{if } n = 0; \\ \prod_{d_s \in S_n} [1 - r(c, d_s)] & \text{else.} \end{cases} \quad (5)$$

Note that in Equation (5), we assume the independence between mined subtopics within a category, just like that in the xQuAD framework [20] and the IASelect model [1]. Actually, we have tried to avoid dependent subtopics when mining the subtopics from query logs in this paper.

5.3 Topic novelty model

$\Phi(d, S_n, \mathbb{C})$ can be regarded as the novelty of document d to the currently selected document set S_n, just like the MMR algorithm [2]:

$$\Phi(d, S_n, \mathbb{C}) = 1 - \max_{d_j \in S_n} Sim(d, d_j, \mathbb{C})$$

We combine multiple types of subtopics, rather than document content, to measure the similarity between two documents as (6).

$$Sim(d, d_j, \mathbb{C}) = \sum_{C \in \mathbb{C}} Sim(d, d_j, C) \quad (6)$$

$Sim(d, d_j, C)$ is the similarity between two documents in terms of subtopic category C. We assume that the documents d and d_j are similar to each other if they are similarly relevant to the subtopics within current category C. Thus, we design $Sim(d, d_j, C)$ as:

$$Sim(d, d_j, C) = 2 \times \left(1 - \frac{1}{1 + e^{-\sum_{c \in C} w_c \cdot \left| r(c,d) - r(c,d_j) \right|}} \right)$$

6. EXPERIMENTS

In this section, we design several experiments to mainly verify: (1) whether combining multiple types of subtopics help discover user intents; (2) whether our proposed models can improve search result diversity; (3) whether the proposed models outperform existing state-of-the-art models. We introduce experiment settings and evaluation metrics in Section 6.1. Statistics about subtopic coverage are discussed in Section 6.2, following which detailed ranking results are presented and analyzed.

6.1 Experiment setup

As introduced in Section 3, we use the TREC 2009 Web track query set and the ClueWeb09 web page collection to evaluate our algorithms. We retrieve the top 1,000 results

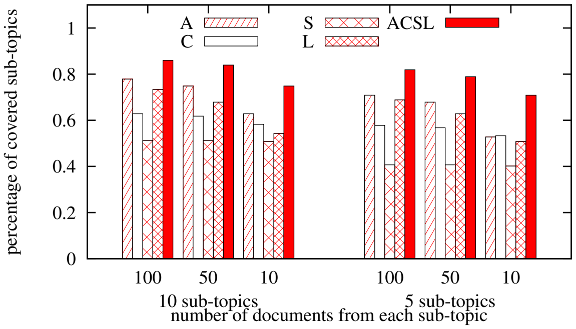


Figure 1: Subtopic coverage statistics

for each query. Four categories of subtopics and their search results are mined as that introduced in Section 4.

We use letters '**A**' (Anchor), '**L**' (Log), '**C**' (Clustering), and '**S**' (Site) in experiments to denote four types of subtopics. An additional letter '**N**' is used to identify the topic Novelty model, and no additional symbols are used for the topic richness model. The baseline model without result diversification is denoted as "**BASE**".

We evaluate result diversity using the α-NDCG measurement proposed in [4]. Given a query, the Discounted Cumulative Gain (DCG) measurement over the top K documents is calculated as:

$$\text{DCG}[K] = \sum_{j=1}^{K} \frac{\sum_{i=1}^{m} J(d_j, i)(1-\alpha)^{C(i,j-1)}}{\log(1+j)}$$

where m is the number of subtopics of the query; and each $J(d_j, i)$ is a binary relevance judgment for subtopic i of result d_j returned at position j. $C(i, j-1)$ is the number of documents ranked up to position $j-1$ that have been judged to be relevant to subtopic i, i.e., $C(i, j-1) = \sum_{c=1}^{j-1} J(d_c, i)$. α is a constant with $0 \le \alpha \le 1$, which reflects the possibility of assessor error. Note that duplicated documents within the same subtopic are treated as totally irrelevant to the subtopic when α equals to 1. The normalized discounted cumulative gain (α-nDCG) for the top K results is computed as: α-nDCG@K = DCG$[K]$/DCG$'[K]$, where DCG$'[K]$ is the ideal gain, which is calculated based on ideal ordering of documents. The final α-nDCG value of a query set is computed by averaging α-nDCG values over all queries. We report α-NDCG@5 and α-nDCG@10 in this paper.

6.2 Subtopic coverage

Before experimenting with the diversification models, we first investigate whether using multiple types of subtopics can help discover real user intents and subtopics. We extract all human-edited subtopics and their relevant documents from the judgment set. For each mined subtopic category C, we count the number of human-edited subtopics that are covered by C. We say that a human-edited subtopic c_r is covered by C if at least one of c_r's relevant documents appears in the results associated with some mined subtopics within C. We have tried the top 10 or top 5 subtopics in each mined subtopic category, and the top 100, top 50, or top 10 results associated with each mined subtopic. The percentages of covered subtopics are plotted in Figure 1. Note that in the query set, 199 of all 243 manually edited subtopics have at least one judged relevant document. We find that **combining four types of subtopics (series ACSL) can cover more manually-edited subtopics over any sole use of one category**, no matter how many mined subtopics and associated results are used. About 86% (i.e., 171) of the subtopics are covered by at least one category when the top 10 subtopics in each category and the top 100 results for each subtopic are used.

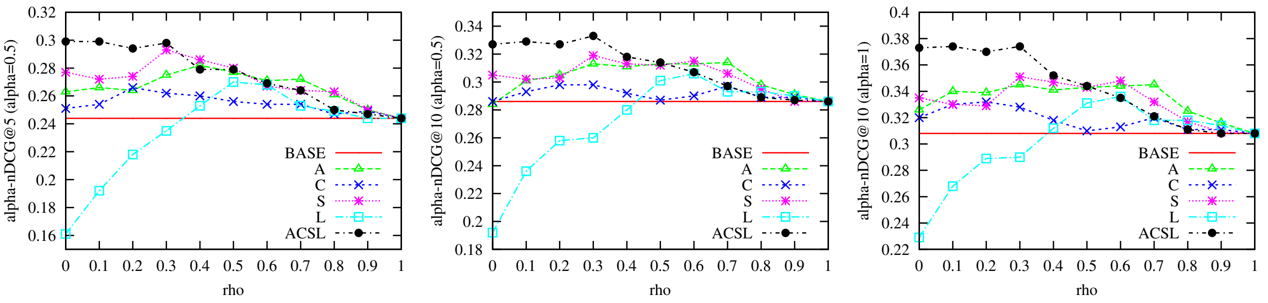
This figure also shows that more human-edited subtopics are covered by each subtopic category when more mined subtopics within the category and more results are used. The site-based method is an exception. It covers similar numbers of human-edited subtopics no matter top 100 or top 10 results from each site are used. This is because that most of the sites return less than 10 results in the original search results. The two settings are actually same in the number of results from each site.

6.3 Results of the topic richness model

Figure 2 shows the results of the topic richness model when different types of subtopics are used. Note that the results are generated based on the top 1,000 results of the original query and no new documents retrieved by subtopics are considered. Experimental results indicate that the sole use of each type of subtopics can improve result diversity with appropriate settings of ρ, **while a combination of them (series ACSL) performs the best**, especially in terms of α-nDCG@10 with α=0.5 or α=1. This is because that: (1) combining multiple types of subtopics can help discover more user intents, as indicated in Section 6.2; and (2) multiple types of subtopics are complementary in better understanding of document topics. For example, for two documents that appear in two different sites but have similar content, the site-based subtopics cannot identify their duplication while the result clustering-based subtopics can.

We also try other combinations of subtopics, for example, just combining anchor text-based and query-log based subtopics. Experimental results show that any combination outperforms the sole use of one type of subtopic. Note that a combination of subtopics from anchor texts, search result clusters, and web sites of search results (named MSRAACSF, with ρ=0.5) performs top 1 in the diversity task of the TREC 2009 Web track, in terms of α-nDCG@10. We skip the details of the experimental results due to space limitation.

Figure 2 also shows that the anchor text-based subtopics (series A) and the query log-based subtopics (series L) can improve diversity but they harm α-nDCG when the final ranking output heavily depends on result diversity while ignoring document relevance (ρ closes to 0). By analyzing detailed ranking results, we find that the two methods harm α-nDCG mainly because some novel but less relevant or even irrelevant results are ranked higher. Sometimes there are few relevant documents retrieved for some subtopics or irrelevant subtopics are wrongly mined, and thus irrelevant documents are returned at the top. In addition, the evaluation of diversity is influenced by the judgments. Sometimes reasonable subtopics are mined yet excluded from the manually created subtopics. For example, "castle defender" and "public defender" are mined as subtopics for query "defender", but they are not listed in the judgments. For the query log-based methods, some top returned documents are not judged and thus treated as irrelevant because of the incomplete judgments.



(a) α-nDCG@5(α=0.5) (b) α-nDCG@10(α=0.5) (c) α-NDCG@10(α=1)

Figure 2: Results of the topic richness model when different types of subtopics are used

6.4 Results of the topic novelty model

We combine different types of subtopics in the topic novelty model and compare them with results of the topic richness model. We only report α-nDCG@10, the primary measurement used in TREC 2009, due to space limitation. In Figure 3, series NACSL stands for the topic novelty model using all four types of subtopics. Results show that it outperforms the baseline ranking (BASE) and all other models that only use one type of subtopics (NA, NC, NS, and NL). This means that **combining multiple types of subtopics can also improve result diversity** in the topic novelty model.

Figure 3 also shows that the model does not perform as well as the topic richness model, whenever only one category of subtopic or all categories are used. This is mainly because the novelty model does not consider topic coverage information. When selecting next document, a document d_1 that covers five new subtopics and another document d_2 that covers only one new subtopic may get similar novelty scores close to 1. Actually, we would like to rank d_1 higher than d_2 if d_1's relevance is not significantly worse because d_1 may have a higher probability of matching real user intents. The topic richness model considers such topic coverage information; hence it outperforms the topic novelty model.

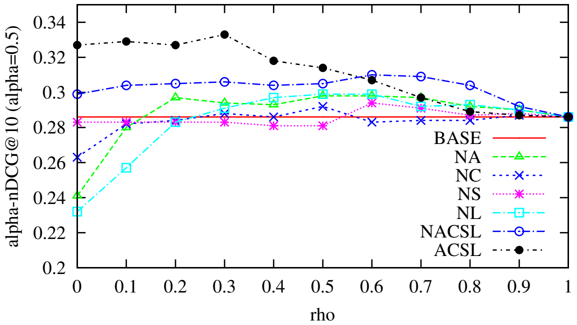


Figure 3: Result of the topic novelty model

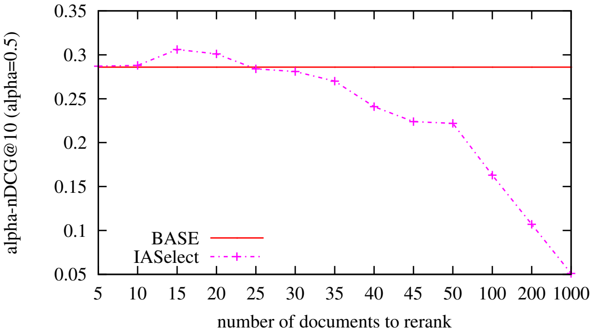


Figure 4: Results of the IASelect algorithm

6.5 Comparison with existing work

We implement two state-of-the-art diversification models to compare with our approaches.

The first model we implement is the IASelect model proposed by Agrawal et al. [1]. The difference between the model and our proposed models is that it diversifies search results based on topical categories (and only based on topical categories) of queries and documents. We use the document and query classification tool introduced in [22] to categorize all queries and their top 1,000 documents (retrieved by the baseline ranking function) into 16 pre-defined topical categories, and then use the IASelect algorithm to re-rank a certain number of top documents. Figure 4 shows the results of the IASelect algorithm when different numbers of top results are re-ranked. The figure shows that the algorithm is less stable. It can improve result diversity based on a certain number of top results (e.g., 15 to 20 in this paper), but its effectiveness decreases when more results are involved in re-ranking. This is because it ranks documents totally based on category vectors without considering document relevance. More irrelevant documents within relevant categories may be ranked to the top when more results are used in re-ranking. Furthermore, since the subtopics in the query set are not labeled based on topical categories, the inconsistence between subtopics definition and topical category also hurts the effectiveness of the IASelect model. We find that a large percentage of manually-judged subtopics in the query set are of the same topical category. For example, for the query "map", the top four labeled subtopics are finding the homepages of Google Maps, MSN Maps, Yahoo Maps, and MapQuest. Obviously these four intents belong to similar topical categories and it is hard to differentiate the documents that are relevant to these subtopics by using the IASelect algorithm.

The second algorithm we implement is the MMR algorithm [2], which diversifies results based on similarity of document content. We generate a TF-IDF term vector based on full document content for each document, and use the cosine similarity of the TF-IDF vectors to measure the similarity of

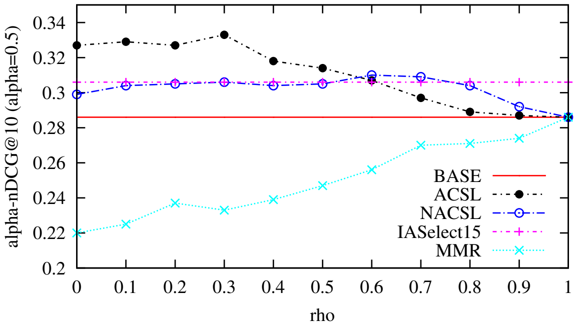


Figure 5: Result comparison of all models

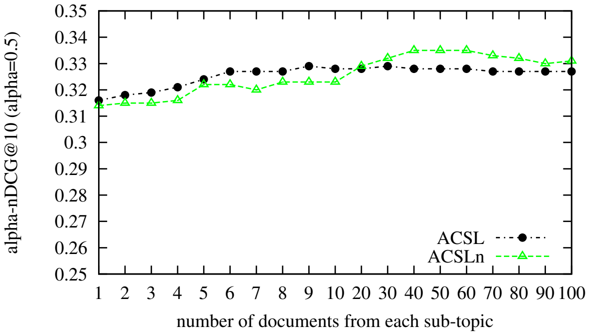


Figure 6: Results of the topic richness model with different numbers of documents from subtopics

two documents. We first experiment with re-ranking a different number of top results when ρ=0.5. We find that results do not significantly change with the number of re-ranked results increasing. This is because the MMR algorithm uses the original ranking score to make sure significantly irrelevant results cannot be ranked to the top.

In Figure 5, we show the results of the MMR algorithm (series MMR) when all 1,000 results are re-ranked, together with the results of the topic richness model (series ACSL), the topic novelty model (series NACSL), and the IASelect model (IASelect15) when the top 15 results are re-ranked. The figure shows that the topic richness model ACSL, which combines four different types of subtopics, performs the best. It (with ρ=0.3) significantly outperforms the topic novelty model (with ρ=0.6) and the IASelect model (p-value<0.05 in the two-tailed t-test for both). The topic novelty model NACSL is slightly better than the IASelect model but the difference is not significant (p-value >0.05). The MMR algorithm performs the worst (significantly worse than NACSL with p-value<0.05), which may be caused by the following reason. The MMR algorithm fully depends on document content, and does not use any (other) subtopics. It may mismatch with the objective of TREC2009 Web track diversity task as some subtopics are not easily identified by content. For example, for the query "obama family tree", most of the top 10 results share similar words, such as "barack", "african", "kenya", "country", and "launch". It is really difficult to identify and diversify the following judged subtopics by comparing their content similarity: (1) where did Barack Obama's parents and grandparents come from; (2) find biographical information on Barack Obama's mother.

6.6 Experiments on the number of documents

Using different numbers of documents associated with each subtopic may impact the coverage and richness of subtopics. In this experiment, we use 1,000 results from the original query and use different numbers of documents corresponding to each subtopic in the topic richness model with ρ=0, and plot the results in Figure 6. Note that series ACSLn stands for the experiment in which *new* documents retrieved by anchor-based or query log-based subtopics are incorporated. This figure indicates that:

(1) Using more documents associated with subtopics (for example, from 10 to 50) improves search result diversity, no matter whether the new documents are used in ranking or not. However, for the ACSL method, the performance goes flat when more than ten documents are used. Using more documents can help discover new subtopics and enrich the representation of documents. That is why the diversity is improved. But when most relevant documents have already been covered by the top results, using more documents cannot bring much benefit.

(2) When comparing two methods, we find that adding new documents (series ACSLn) retrieved by subtopics in ranking outperforms the ACSL method when more than 20 documents for each subtopic are used, but the difference is not statistically significant (p-value>0.05 in t-test). It is risky to use the ACSLn method when less than 20 documents from each subtopic are used. Although some new subtopics can be discovered by new documents, some irrelevant document may also be ranked higher, especially when subtopic information are not enough.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to diversify search results in a multi-dimensional way.

We first mine subtopics, which are defined as pieces of information that a query contains, from four different types of data sources, namely anchor texts, query logs, search result clusters, and the web sites of search results. We reveal that combining these subtopics together can help discover more user intents and finally benefit result diversity, because they cover query information from different dimensions, and favor different queries.

We then propose a general framework, which incorporates multiple types of explicit subtopics to diversify search results. We implement two models, namely the topic richness model and the topic novelty model, which emphasize topic coverage or document novelty in terms of the subtopics of different categories that are associated with a document.

Experimental results on TREC 2009 Web track data indicate that combining four types of subtopics improves result diversity over using only one type of subtopics in terms of α-NDCG, and the topic richness model is more effective than the topic novelty model. The combination of subtopics plus the topic richness model also performs significantly better than two state-of-the-art models, the MMR model based on document texts and the IASelect model based on topical categories.

In the future, we plan to incorporate topical categories and raw content to our existing subtopics. We also plan to mine subtopics from more data sources (such as Wikipedia and WordNet). The problem of search result diversity becomes more and more interesting when a large number of subtopic categories are available. For instance, should we simply combine all subtopics using our proposed framework, or select the most suitable dimension(s) to the query? We will continue investigating this problem in future work.

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09*, 2009.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, 1998.

[3] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *RecSys '08*, 2008.

[4] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR '08*, 2008.

[5] C. L. Clarke, M. Kolla, and O. Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *ICTIR '09*, 2009.

[6] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In TREC 2009 Proceedings, 2009.

[7] The clueweb09 dataset. `http://boston.lti.cs.cmu.edu/Data/clueweb09/`.

[8] Clusty the clustering search engine. `http://clusty.com/`.

[9] M. Coyle and B. Smyth. On the importance of being diverse: analysing similarity and diversity in web search. *Int. Inf. Proc. II*, 2005.

[10] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07*, 2007.

[11] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *KDD '09*, 2009.

[12] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *IPM*, 36(2), 2000.

[13] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, New York, NY, USA, 2001. ACM.

[14] T. Leelanupab, M. Halvey, and J. Jose. Application and evaluation of multi-dimensional diversity. In *Theseus/ImageClef 2009 Workshop*, 2009.

[15] Microsoft. Bing search engine. `http://www.bing.com`.

[16] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims. Sigir workshop idr '09, 2009. `http://ir.cis.udel.edu/IDR-workshop/idr-workshop-2009.pdf`.

[17] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *SIGIR '06*, 2006.

[18] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 781–790, New York, NY, USA, 2010. ACM.

[19] T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C.-Y. Lin. Simple evaluation metrics for diversified search results. In *Proceedings of the 3rd International Workshop on Evaluating Information Access (EVIA)*, 2010.

[20] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 881–890, New York, NY, USA, 2010. ACM.

[21] R. L. T. Santos, J. Peng, C. Macdonald, and I. Ounis. Explicit search result diversification through sub-queries. In *ECIR*, 2010.

[22] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Q2C at UST: our winning solution to query classification in kddcup 2005. *SIGKDD Explor. Newsl.*, 7(2), 2005.

[23] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1), 1999.

[24] K. Song, Y. Tian, W. Gao, and T. Huang. Diversifying the image retrieval results. In *MULTIMEDIA '06*, 2006.

[25] R. Song, J.-R. Wen, S. Shi, G. Xin, T.-Y. Liu, T. Qin, X. Zheng, J. Zhang, G.-R. Xue, and W.-Y. Ma. Microsoft research asia at web track and terabyte track of trec 2004. In *TREC*, 2004.

[26] Z. Dou, K. Chen, R Song, Y. Ma, S. Shi, and J.-R. Wen. Microsoft Research Asia at theWeb Track of TREC 2009. In *TREC*, 2009.

[27] M. Strohmaier, M. Kröll, and C. Körner. Intentional query suggestion: making user goals more explicit during search. In *WSCD '09*, 2009.

[28] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *ICDE '08*, 2008.

[29] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *SIGIR '07*, 2007.

[30] J.-R. Wen and W.-Y. Ma. Webstudio: building infrastructure for web data management. In *SIGMOD Conference*, pages 875–876, 2007. `http://research.microsoft.com/en-us/projects/webstudio`.

[31] Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. In *ICML '08*, 2008.

[32] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04*, 2004.

[33] C. Zhai and J. Lafferty. A risk minimization framework for information retrieval. *IPM*, 42(1), 2006.

[34] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03*, 2003.

[35] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *SIGIR '05*, 2005.

[36] X. Zhu, A. Goldberg, J. Van Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. *HLT-NAACL*, 2007.

[37] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05*, 2005.