Microsoft Research Asia at the Web Track of TREC 2009

Zhicheng Dou*, Kun Chen†, Ruihua Song*, Yunxiao Ma*, Shuming Shi*, and Ji-Rong Wen*
*Microsoft Research Asia, †Xi'an Jiongtong University
*{zhichdou, rsong, yunxiaom, shumings, jrwen}@microsoft.com, †cs.kunchen@gmail.com

Abstract

In TREC 2009, we participate in the Web track, and focus on the diversity task. We propose to diversify web search results by first mining subtopics, and then rank results based on mined subtopics. We propose a model to diversify search results by considering both relevance of documents and richness of mined subtopics. Our experimental results show that the model improves diversity of search results in terms of α-NDCG, and combining subtopics from multiple data sources helps further improve result diversity.

1. Introduction

We propose mining subtopics to diversify search results. A subtopic approximates to a piece of information or user intent covered by a query. Different from the previous work that focuses on one source, e.g., search logs or page content, we argue that mining subtopics from multiple complementary sources can help better understand user intents. We preliminarily mine subtopics from the following data sources: (1) anchor texts that represent the opinions of web annotators; (2) clusters of search results that show the perspective of page content; (3) websites of search results that reflect web information organization.

We propose a search result diversification model that diversifies search results based on our mined subtopics. As shown in Figure 1, given a query, we first mine subtopics together with their importance. We then combine the original query and the subtopics to retrieve documents for ranking. We diversify retrieved search results by considering both their relevance and their subtopic richness. A greedy algorithm is employed to iteratively select the next best document from the remaining documents.

We design experiments and submit official runs to answer the following questions: (1) can our search result diversification algorithms improve diversity of search results? (2) can subtopics from multiple data sources help predict user intents and further improve search result diversity? (3) can diversification algorithms improve or harm adhoc ranking effectiveness?

The remaining parts of the report are organized as follows. In Section 2, we briefly introduce our retrieval platform. We then introduce our methods of mining subtopics in Section 3, and propose a search result diversification model in Section 4. We report our experimental results in Section 5, and then conclude our work in Section 6.

2. Retrieval Platform

2.1. WebStudio

We use the WebStudio[1], an experimental search system for facilitating large-scale, end-to-end search experiments, to index and retrieve documents. Using the WebStudio, users can easily do search experiments, by implementing pre-defined interfaces or inheriting existing implementations. Multiple users can run different search experiments simultaneously. As an end-to-end search system, the WebStudio also provides search interfaces, via which we can input queries and get search results.

We use 40 machines, each of which has 4 CPU's, 16GB memory, and 4 1T IDE disks, to deploy an instance of WebStudio. We index all English pages in the ClueWeb09 dataset (about 500M pages) using this WebStudio instance. Note that web pages are distributed into these machines. There are about 12.5M web pages on each machine. It takes about 20 hours for each machine to index the data. Before indexing the data, anchor texts are

[1] http://research.microsoft.com/en-us/projects/WebStudio

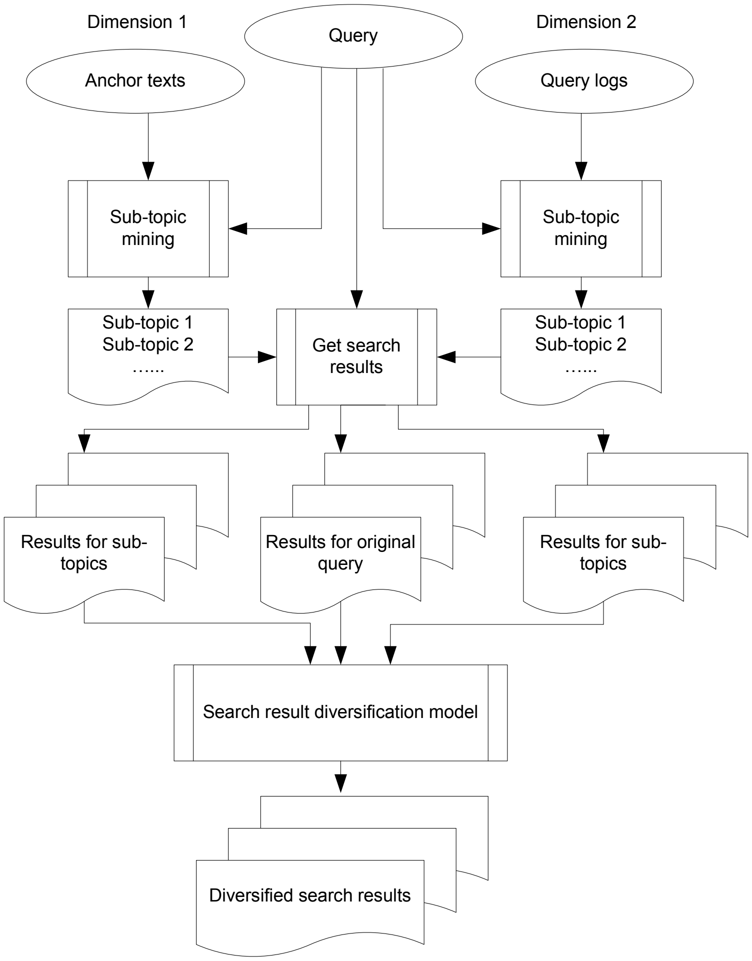


Figure 1. Framework of search result diversification based on two types of subtopics

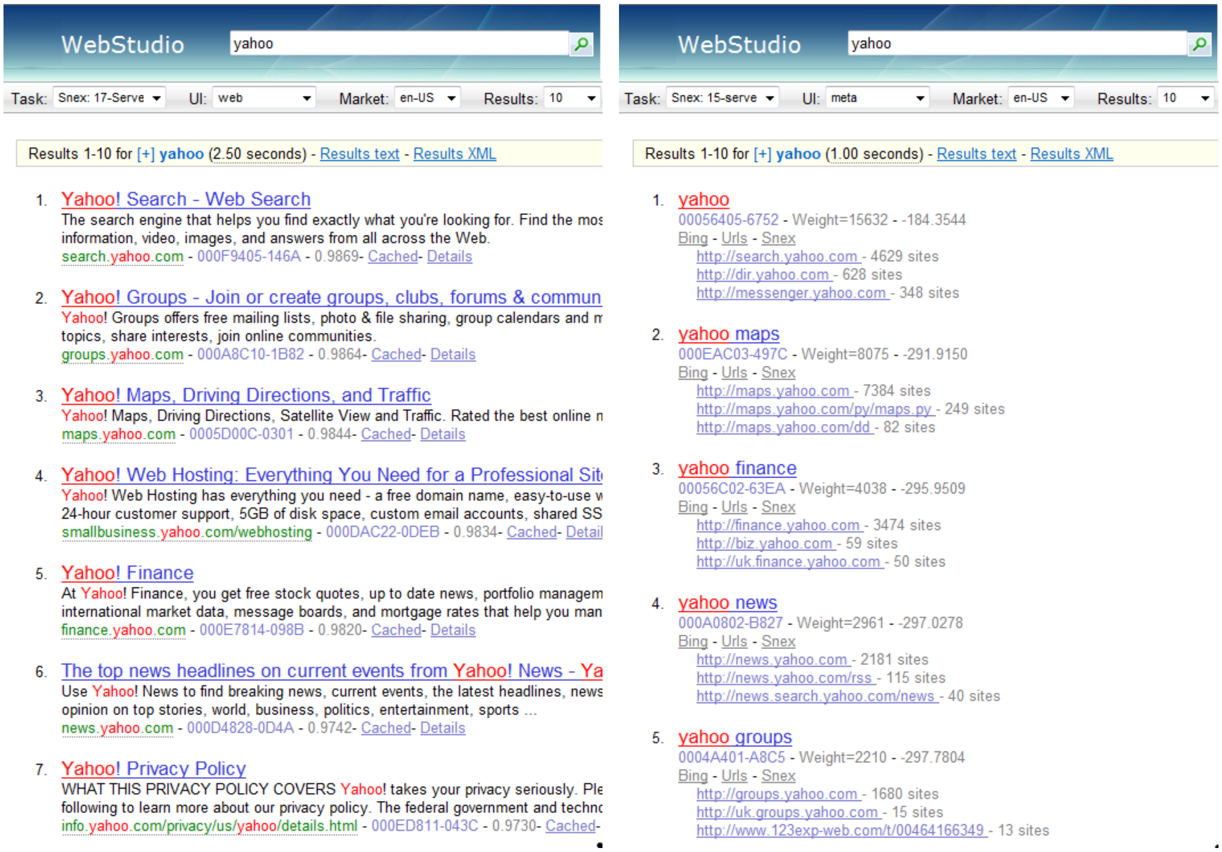
extracted from source pages and merged into the pages they link to. In query phase, each query request from the client will be sent to all machines via an aggregator. Each machine will process the query and retrieve top results separately. The aggregator then collects and aggregates all results and returns them to the client. For the ClueWeb09 dataset, it takes averagely about 5 seconds to get top 1,000 results for a query using WebStudio.

WebStudio is a flexible indexing and ranking platform. It is designed to support different types of source data and index structures. Besides web pages, we index all unique anchor texts of the ClueWeb09 dataset using the WebStudio. We treat each anchor text as a document, and build statistics of anchor text (such as the number of links with this anchor text) as attributes of the document. Given a query, we can search and rank all related anchor texts which include one or more query terms. Other similar data, such as query logs, can be processed in the same manner. Figure 2 shows two snapshots for search results of the query "yahoo". The left shows general web page results, and the right shows anchor text results.

2.2. Ranking function

As introduced in Section 1, our primary goal is to develop search result diversification methods, but not to tune an optimal general search function. We simply use a ranking function named MSRA2000, which was proposed and used in TREC 2004 [1]. It combines augmented BM25 scores of four different fields including title, body, URL, and anchor. The augmented BM25 function considers the proximity between query term occurrences.

We do not use PageRank in MSRA2000 for TREC 2009. Instead we use an aggregated weight of incoming links which is calculated based on the number of incoming links and sources sites.



Task: Snex: 17-Serve UI: web Market: en-US Results: 10

Results 1-10 for [+] yahoo (2.50 seconds) - Results text - Results XML

1. Yahoo! Search - Web Search
   The search engine that helps you find exactly what you're looking for. Find the mos
   information, video, images, and answers from all across the Web.
   search.yahoo.com - 000F9405-146A - 0.9869- Cached- Details
2. Yahoo! Groups - Join or create groups, clubs, forums & commun
   Yahoo! Groups offers free mailing lists, photo & file sharing, group calendars and m
   topics, share interests, join online communities.
   groups.yahoo.com - 000A8C10-1B82 - 0.9864- Cached- Details
3. Yahoo! Maps, Driving Directions, and Traffic
   Yahoo! Maps, Driving Directions, Satellite View and Traffic. Rated the best online n
   maps.yahoo.com - 0005D00C-0301 - 0.9844- Cached- Details
4. Yahoo! Web Hosting: Everything You Need for a Professional Sit
   Yahoo! Web Hosting has everything you need - a free domain name, easy-to-use w
   24-hour customer support, 5GB of disk space, custom email accounts, shared SS
   smallbusiness.yahoo.com/webhosting - 000DAC22-0DEB - 0.9834- Cached- Detail
5. Yahoo! Finance
   At Yahoo! Finance, you get free stock quotes, up to date news, portfolio managem
   international market data, message boards, and mortgage rates that help you man
   finance.yahoo.com - 000E7814-098B - 0.9820- Cached- Details
6. The top news headlines on current events from Yahoo! News - Ya
   Use Yahoo! News to find breaking news, current events, the latest headlines, news
   opinion on top stories, world, business, politics, entertainment, sports ...
   news.yahoo.com - 000D4828-0D4A - 0.9742- Cached- Details
7. Yahoo! Privacy Policy
   WHAT THIS PRIVACY POLICY COVERS Yahoo! takes your privacy seriously. Ple
   following to learn more about our privacy policy. The federal government and technc
   info.yahoo.com/privacy/us/yahoo/details.html - 000ED811-043C - 0.9730- Cached-

(a) Web pages

Task: Snex: 15-serve UI: meta Market: en-US Results: 10

Results 1-10 for [+] yahoo (1.00 seconds) - Results text - Results XML

1. yahoo
   00056405-6752 - Weight=15632 - -184.3544
   Bing - Urls - Snex
   http://search.yahoo.com - 4629 sites
   http://dir.yahoo.com - 628 sites
   http://messenger.yahoo.com - 348 sites
2. yahoo maps
   000EAC03-497C - Weight=8075 - -291.9150
   Bing - Urls - Snex
   http://maps.yahoo.com - 7384 sites
   http://maps.yahoo.com/py/maps.py - 249 sites
   http://maps.yahoo.com/dd - 82 sites
3. yahoo finance
   00056C02-63EA - Weight=4038 - -295.9509
   Bing - Urls - Snex
   http://finance.yahoo.com - 3474 sites
   http://biz.yahoo.com - 59 sites
   http://uk.finance.yahoo.com - 50 sites
4. yahoo news
   000A0802-B827 - Weight=2961 - -297.0278
   Bing - Urls - Snex
   http://news.yahoo.com - 2181 sites
   http://news.yahoo.com/rss - 115 sites
   http://news.search.yahoo.com/news - 40 sites
5. yahoo groups
   0004A401-A8C5 - Weight=2210 - -297.7804
   Bing - Urls - Snex
   http://groups.yahoo.com - 1680 sites
   http://uk.groups.yahoo.com - 15 sites
   http://www.123exp-web.com/t/00464166349 - 13 sites

(b) Anchor text

Figure 2. Web page and anchor text search results for query “yahoo” retrieved from the WebStudio system. Note that Figure (b) shows top anchor texts related to query “yahoo”.

3. subtopic Mining

Usually different types of sources contain somehow different information on user intentions. We assume that better understanding of user intentions can be achieved by combining the subtopics from these complementary sources. In this paper, we mine subtopics from three different data sources, including anchor texts, search result clusters, and web site of search results. Other data sources, such as query logs and dictionaries, are also good data sources that show different dimensions of diversity. We will investigate these data sources in future work. Given a source, we associate a weight with each mined subtopic on how important the subtopic is. In the following sections, we will briefly introduce subtopics we have mined.

3.1. Anchor texts

Anchor texts created by web designers provide descriptions of target documents. They are usually short and descriptive, and share similar characteristics with web queries. We use the WebStudio system to index all unique anchor texts together with some statistics about their popularity, and use the following ranking function to retrieve top 10 anchor texts related to a given query:

$$f(q,c) = \text{NumOfSites} + log(\text{NumOfLinks} - \text{NumOfSites} + 1) + 10 * \text{QueryLengh}/\text{AnchorLength} \quad (1)$$

In the above ranking function, NumOfLinks means the number of links that contain the anchor text c, and NumOfSites means the number of unique source sites. AnchorLength is the number of terms contained in the anchor text. An anchor text will be ranked higher if it is more popular and it contains less words. Note that here we assume a default AND operator between query terms. This means that all query terms must appear in retrieved anchor texts.

We simply assume that each anchor text can stand for an unique subtopic contained in the query. Based on (1), we use the following formula to calculate importance of a subtopic. We assume that an anchor text has an average weight, namely 0.5, if it has about 50 unique sources sites.

$$w_c = \frac{1}{1 + e^{-(f(q,c)-50)/50}}$$

For each subtopic, we treat it as a new query and get its top 100 results using the WebStudio.

3.2. Search results clusters

Search result clustering is one of the ways to solve the problem of query ambiguity. There has been much work on this area. By using clustering algorithm, documents with similar content and key phrases are grouped together. We use the search result clustering algorithm introduced in [2] to group top 200 original search results into 10 clusters (i.e., subtopics) based on key phrases in documents. We treat each cluster as a subtopic, and assume that a subtopic is more important if: (1) the cluster is ranked higher than other clusters; (2) the cluster contains a document which is ranked higher in original ranking list. We use the following equation to calculate importance of a subtopic:

$$w_c = 0.5 \times \left(\frac{10 - \text{ClusterRank} + 1}{10} + \frac{1.0}{\text{HighestRankInCluster}} \right)$$

Here ClusterRank is rank of the cluster among all clusters, and HighestRankInCluster is the original rank of the document which has the highest rank within the cluster. Note that we also include the “Other” cluster which includes documents that are not grouped into any other clusters.

We do not need to retrieve search results for subtopics because we can directly get documents in each subtopic. In each cluster, the order of documents in the original query is reserved.

3.3. Sites of search results

As web pages from the same web site are sometimes providing similar information, it is natural to rank results from multiple different web sites, instead of the same site, in top results. It is quite easy to accomplish this in our framework: just treating each web site as a subtopic.

We count the number of results for each site which appears in the top 200 results, and use the following equation to calculate its importance:

$$w_c = \frac{1}{1 + e^{-(\text{NumOfResults}-2)}}$$

Here NumOfResults is the number of results in the site. We assume that the more results a site includes, the more important the site is. We assume that a site gets a medium score 0.5 when there are two results from this site.

We also do not need to retrieve search results for site-based subtopics because we have already retrieved the documents in an initial search. Ranking order of documents in original search are kept in subtopics.

4. Search Result Diversification

In this section, we propose a search result diversification algorithm which uses explicit subtopics.

4.1. Algorithm

To better illustrate our algorithm, we first define the following symbols:

- R: candidate search results. It may be the set of all documents, or the set of document to be re-ranked (coming from the original query or subtopics);
- $s(q, d)$: the relevance score of document d for query q;
- $r(q, d)$: the importance (0 to 1) of document d for query q according to original relevance;

- $r(q_c, d)$: the importance of document d for subtopic q_c;
- $\mathbb{C}$: the set of all diversity categories;
- $C \in \mathbb{C}$: one specific diversity category (e.g., anchor text);

We assume that a good ranking should cover as many relevant subtopics as possible. As it is a NP problem to find such a ranking, we employ a greedy algorithm to iteratively select the best document from the remaining documents. Given R as the set of document to be ranked, and suppose S is the set of documents which have already been selected, we use the following equation to select next best document from remaining documents:

$$d_{|S|+1} = \arg \max_{d \in R \setminus S} \left[\alpha r(q, d) + \Re_{C \in \mathbb{C}} v(d, S, C) \right] \tag{2}$$

here α is a parameter which controls the importance of original relevance and the diversity of ranking. $\Re$ is a operator which is used to combine multiple dimensions of subtopics. It can be $\sum, \prod, \max$, or $\min$. $v(d, S, C)$ is the importance score of document d in terms of subtopic definition C given a set of document S already selected, and

$$v(d, S, C) = \sum_{c \in C} w_c \cdot \phi(c, S) \cdot r(q_c, d)$$

w_c is the weight of subtopic c in subtopic definition C for query. $\phi(c, S)$ is the current importance of subtopic c after documents set S have been selected. We assume that for a subtopic, if some documents have already been selected for it in previous steps, its importance should be reduced. By using this method, we prompt search result diversity based on mined explicit subtopics. Suppose that documents are independent in terms of importance to one subtopic, we use the following function, which is similar to that in [3], to calculate $\phi(c, S)$:

$$\phi(c, S) = \begin{cases} 1 & \text{if } S = \{\}; \\ \prod_{d_s \in S} \left[1 - r\left(q_c, d_s\right)\right] & \text{else.} \end{cases} \tag{3}$$

4.2. Parameters

We have the following parameter settings in our search result diversification algorithm.

- α. Similar to the MMR algorithm [4], parameter α in Equation 2 decides a tradeoff between original document relevance and diversity on subtopics. Documents will be ranked totally based on subtopic diversity when α equals to 0, while it will be equivalent to original ranking when α is very large.
- $r(q, d)$ and $r(q_c, d)$. These two functions decide how important the documents are for the original queries and corresponding subtopics. We use the same setting for these two functions in a run. Suppose we already have a rank list for query q, the following functions can be used to decide the importance of document d for query q:
  - OriScore: $r(q, d) = s(q, d)$. This just uses the original relevance score $s(q, d)$. Please note that $s(q, d)$ should be normalized to [0, 1]; The problem is that the distribution of original relevance score is decided by original ranking function, and different ranking functions may generate different distributions of ranking scores.
  - Sigmoid. Suppose $s(q, d)$ has been normalized to [0,1]. By using the sigmoid function, we can get a new distribution of document importance.

    $$r(q, d) = \frac{1}{1 + e^{-(s(q,d) - 0.5) * m}}$$

  - Rank. When $s(q, d)$ is unknown, we can directly use the rank of document: $r(q, d) = \frac{1.0}{rank(q,d)}$. $r(q, d)$ will decrease rapidly with the increasing of document rank.
  - RankSqrt. $r(q, d) = \frac{1}{\sqrt{rank(q,d)}}$. It is similar to the previous one, but is less sensitive to the rank of document.
  - LinearRank. $r(q, d) = \frac{N - rank(q, d_i) + 1}{N}$. Here N is the total number of results.
- Results to be diversified. For some subtopics, such as subtopics mined from anchor texts, we can get additional search results for subtopics. These results may not appear in results of the original query. It may help discover new relevant documents for the original query if these results are used, but it may also increase the danger of more irrelevant results being ranked higher.

- Combination of diversity dimensions $\Re$. We can use different functions, such as $\sum, \prod, \max$, or $\min$, to combine subtopics from different diversity dimensions. For example, a document needs to be novel enough in each diversity dimension if function $\min$ is used, while it just needs to be novel in one diversity dimension if function $\max$ is used.

5. Experimental Results

We design several experiments to investigate whether our proposed algorithm can improve diversity of search results. We try several combinations of mined subtopics which are described in the following list (note that MSRA is abbreviation of Microsoft Research Asia).

- MSRABASE - Baseline, a general ranking function without results diversification
- MSRAC - Search result diversification based on search result (C)lustering.
- MSRAS - Search result diversification based on (S)ites of search results.
- MSRAAF - Search result diversification based on anchor texts. New documents from subtopics are not included.
- MSRAAT - Search result diversification based on (A)nchor texts. New documents from subtopics are included.
- MSRACS - Search result diversification based on search result (C)lustering and (S)ites of search results.
- MSRAACSF - Search result diversification based on (A)nchor texts, (C)lusters, and (S)ites of search results. New documents from subtopics are not included.
- MSRAACST - Search result diversification based on (A)nchor texts, (C)lusters, and (S)ites of search results. New documents from subtopics are included.

As we do not have judgment data to tune ranking functions and select parameters, we just simply select the following settings for our official run submissions:

- $\alpha = 1.3$, which means that the original relevance is slightly more important than subtopic richness;
- $\Re = \sum$. We assume that a document should have a reasonable overall subtopic diversity score across all diversity dimensions.
- Use RankSqrt to calculate document importance to the original query and subtopics, i.e., $r(q, d) = \frac{1}{\sqrt{rank(q,d)}}$.

5.1. Diversity task

As we are allowed to submit up to three runs, we submitted MSRABASE, MSRACS, and MSRAACSF to the diversity task. We evaluate the other runs using the tool provided by TREC after the judgments are released. As there may be some new documents that are not judged in these runs, the results for new runs might be worse than correct results.

Table 1 includes the results of all runs. This table shows that:

- All runs with result diversification outperform the baseline run MSRABASE in terms of α-NDCG. They also outperform the baseline on the top five results in terms of IA-P (intention aware precision).
- The method MSRAACSF performs the best among three submitted runs in terms of α-NDCG@10, the primary measurement used in the diversity task. Note that MSRAACSF is also the best run in the diversity task. After experimenting with different settings of the combination parameter α, we find that MSRAACSF is consistently better than other methods. This means that combining multiple types of subtopics can further improve result diversity over any sole use of one type of them.
- The anchor text based methods, MSRAAF and MSRAAT, perform the worst result among all diversification methods. This may caused by the following reasons. Firstly, this method needs additional search results for subtopics. It heavily depends on whether the baseline ranking function can retrieve good results for these subtopics. We do find some bad results are ranked to top for some sub-queries, and these results may be ranked higher in final ranking. Secondly, our mined anchor texts (subtopics) may not fully matched with those given by judgments. For example, anchor text-based subtopics ”castle defender” and “public defender” are not listed in the judgments.
- For runs which use anchor text based subtopics, the differences are not significant when new documents from subtopics are used (runs with T) or not (runs with F). This may by caused by: (1) anchor texts have been used in the original ranking function; (2) we assign a high weight to original ranking. For new documents, their ranking scores will be assumed to be 0, which may stop them being ranked to top.

Table 1. Diversity task results. Runs with * are submitted to the diversity task. Note that MSRAACSF performs the best α-NDCG@10 results among all offical runs in the diversity task

run	alpha-ndcg@5	alpha-ndcg@10	alpha-ndcg@20	IA-P@5	IA-P@10	IA-P@20
MSRABASE*	0.244	0.286	0.328	0.116	0.105	0.098
MSRAC	0.248	0.293	0.334	0.117	0.109	0.099
MSRAS	0.285	0.31	0.351	**0.13**	**0.117**	0.101
MSRACS*	0.281	0.31	0.357	0.126	0.112	0.106
MSRAAF	0.261	0.295	0.334	0.121	0.103	0.091
MSRAAT	0.262	0.288	0.33	0.124	0.099	0.09
MSRAACSF*	0.281	0.316	0.365	0.127	0.112	**0.108**
MSRAACST	**0.287**	**0.318**	**0.366**	**0.13**	0.113	**0.108**

Table 2. Adhoc task results. Runs with * are submitted to adhoc task

runid	map	gmmap	Rprec	bpref	reciprank	P5	P10	P20
MSRANORM*	0.0832	0.036	0.1324	0.2227	0.5824	0.4	0.37	0.312
MSRAC*	**0.0867**	**0.0372**	**0.1439**	**0.2276**	0.5893	0.428	**0.4**	**0.328**
MSRAS	0.0756	0.0317	0.1274	0.2237	0.5917	0.42	0.356	0.262
MSRACS	0.0784	0.0332	0.1297	0.2243	0.5947	0.44	0.374	0.285
MSRAAF*	0.0829	**0.0363**	**0.136**	0.2185	0.6201	0.404	0.354	0.321
MSRAAT	0.0808	0.0354	0.1354	0.2183	0.6056	0.4	0.348	0.313
MSRAACSF	**0.088**	0.0357	0.1342	**0.2287**	**0.6721**	**0.448**	0.378	0.319
MSRAACST	0.0828	0.0352	0.1346	0.2232	**0.6518**	**0.448**	**0.38**	**0.322**

5.2. Adhoc task

We also submitted three runs, including MSRANORM, MSRAAF, and MSRAC, to the adhoc task to investigate whether diversification algorithms can help improve general ranking. Here MSRANORM is also a baseline ranking function which does not consider search result diversity. It is just slightly different from MSRABASE with same different settings of term proximity on anchor field. Note that all other runs submitted to adhoc task are generated based on MSRANORM instead of MSRABASE. Results are shown in Table 2. This table shows:

- MSRAC outperforms the baseline model in terms of map, R_prec, P@10, and P@20, while MSRAACSF and MSRAACST outperform the baseline model in terms of P@5 and recip_rank. This means that MSRAACSF and MSRAACST can help improve top results, while MSRAC can help improve overall ranking.
- Compared with Table 1, we find evaluation results based on two different types of judgments and evaluation metrics are not totally consistent. For example, MSRAS performs the best in diversity task in terms of IA-P@10, while it performs worse than baseline model in adhoc task in terms of P@10. This may caused by the following reasons. Firstly, judgments are made by different annotators. Secondly, judgments may bias towards different types of judgment tools and methods.

5.3. Parameters

Table 3 shows performance of MSRAACSF and MSRAACST when different methods of combining multiple subtopic sources are used. This tables shows that the $\sum$ combination method performs the best, and $\max$ also shows good results in terms of α-NDCG@10 and IA-P@10.

Table 4 shows performance of MSRAACSF and MSRAACST when different types of function $r(q, d)$ and $r(q_c, d)$ are used. This tables shows that our selected function, namely RankSqrt, performs consistently well.

6. Conclusions

In the Web track of TREC 2009, we proposed to mine subtopics from multiple perspectives to diversify search results for a query. We think that mining subtopics across multiple dimensions can help better understand user intents. We proposed an algorithm which combines multiple types of subtopics to diversity search results. Our experimental results show that proposed search result diversification algorithms improve diversity of search results in terms of α-NDCG, and multiple types of subtopics do help predict user intents and further improve result

Table 3. Performance of MSRAACSF with different methods for combining multiple diversity dimensions

runid	alpha-ndcg@5	alpha-ndcg@10	alpha-ndcg@20	IA-P@5	IA-P@10	IA-P@20
MSRAACSFProduct	0.259	0.294	0.338	0.12	0.105	0.099
MSRAACSFMax	0.26	**0.318**	0.355	0.115	**0.122**	0.105
MSRAACSFMin	0.259	0.3	0.337	0.12	0.108	0.099
MSRAACSF	**0.281**	0.316	**0.365**	**0.127**	0.112	**0.108**
MSRAACSTProduct	0.256	0.29	0.333	0.121	0.103	0.096
MSRAACSTMax	0.264	0.32	0.356	0.117	**0.122**	0.105
MSRAACSTMin	0.263	0.296	0.335	0.124	0.106	0.097
MSRAACST	**0.287**	**0.318**	**0.366**	**0.13**	0.113	**0.108**

Table 4. Performance of MSRAACSF and MSRAACST with different types of $r(q, d)$ and $r(q_c, d)$

runid	alpha-ndcg@5	alpha-ndcg@10	alpha-ndcg@20	IA-P@5	IA-P@10	IA-P@20
MSRAACSFRank	0.278	0.31	0.356	**0.127**	0.11	0.1
MSRAACSFLinearRank	0.286	0.311	0.36	0.11	0.093	0.094
MSRAACSFSigmoid	**0.288**	0.315	**0.37**	0.112	0.097	0.1
MSRAACSF	0.281	**0.316**	0.365	**0.127**	**0.112**	**0.108**
MSRAACSTRank	0.275	0.311	0.356	0.127	0.11	0.1
MSRAACSTLinearRank	0.27	0.293	0.346	0.108	0.09	0.093
MSRAACSTSigmoid	0.277	0.303	0.362	0.113	0.096	0.1
MSRAACST	**0.287**	**0.318**	**0.366**	**0.13**	**0.113**	**0.108**

diversity. Furthermore, we found that different types of judgments in the diversity and the adhoc tasks (considering diversity or not) yield inconsistent evaluation results.

References

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

[2] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma, “Learning to cluster web search results,” in *SIGIR ’04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2004, pp. 210–217.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, “Diversifying search results,” in *WSDM ’09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2009, pp. 5–14.

[4] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *In Research and Development in Information Retrieval*, 1998, pp. 335–336.