THUIR at NTCIR-9 INTENT Task

Yufei Xue, Fei Chen, Tong Zhu, Chao Wang, Zhichao Li, Yiqun Liu, Min Zhang, Yijiang Jin, Shaoping Ma
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
z-m@tsinghua.edu.cn

ABSTRACT

This is the first year IR group of Tsinghua University (THUIR) participates in NTCIR. We register the INTENT task and focus on the Chinese topics of subtopic mining and document ranking subtask. In our experiments, we try to mine subtopics from different resources, namely query recommendation, Wikipedia and the query-URL bipartite graph which is constructed by clickthrough data. We also develop some methods to re-rank the subtopics and remove reduplicate ones with query log and search result snippets in search engines. In the document ranking task, methods applied to diversify English documents are used to validate their effectiveness on Chinese pages, such as HITS, Novelty-Result Selection and Documents Duplication Elimination. Based on the new metric, called D#-nDCG, we propose a Document-Diversification algorithm to select the documents retrieved for subtopics mined in the subtopic mining task, and user browse logs are also leveraged to re-rank these selected results.

Keywords

intents, diversity, ambiguity, subtopics, document ranking

Team Name

THU/THUIR

Subtasks/Languages

Chinese Subtopic Mining and Document Ranking

External Resources Used

Google, Bing, Baidu, Sogou, Youdao, Soso, Chinese Wikipedia, Sogou query log

1. INTRODUCTION

IR Group of Tsinghua University (THUIR) participates in INTENT task of NTCIR-9 this year. It is our first experience of NTCIR. We worked on subtopic mining and document ranking subtask, and submitted 5 runs for the Chinese part of each subtask.

In subtopic mining subtask, we have developed two different methods for mining subtopics. The first method is based on search engines' query recommendations and the Wikipedia data. The other is based on the query log of search engine. Furthermore, we make lots of efforts to re-rank subtopics and remove reduplicate ones.

In document ranking subtask, we extend our methods in TREC. We used many diversification algorithms in TREC 2009 and 2010. Some of these algorithms, including HITS based re-ranking, Novelty-Result Selection, Documents Duplication Elimination, were proved to be effective in diversifying documents in English corpus. We are interested in whether they still work in a Chinese corpus and that is the basic motivation for our three runs using the algorithms above. On the other hand, a new metric called D#-measure is proposed to evaluate the diversity of search results. So we submit two other runs, which are created by D#-nDCG-based Selection algorithm (called D#-select). One of the two runs is further diversified by a method based on the user browse logs, which will be described in details later.

2. MINING SUBTOPICS FROM MULTIPLE RESOURCES

2.1 Extracting Subtopics From Search Engines And Wikipedia

Nowadays, commercial search engines usually provide related queries to users in search engine result page(SERP). The recommended queries are related to the user-submitted query in literal or in semantics. They should be specializations, generalizations or parallel concepts of user query. Among the recommendations, specialization is the most common type. The specialized related queries could be considered as subtopics. So we can mine subtopics from the recommended queries of search engines.

For a given topic, we crawl all related queries from 6 Chinese commercial search engines. There are at most 60 related queries from different search engines. Obviously, lots of them are reduplicated because different search engines may recommend same queries. It is reasonable to assume that the more a query is recommended by different search engines, the more reliable this related query is. Based on this assumption, we use the search engines to vote for all related queries. The search engines and the weights of their votes are shown in Table 1.

As mentioned before, a related query may be specification, generalization or parallel concept of the original query. In fact, most of the recommended queries belong to the first class, especially when the input query is short and ambiguous. In NTCIR-9 INTENT task, most given topics have such characteristics. So it is not necessary to take great efforts to classify the related queries into the different types. We only filter out the queries which are substrings of topics, since they are very likely to be generalized topics instead of

Table 1: The search engines and their weights

Search Engine	Weight
Google	1
Baidu	1
Bing	1
Sogou	1
Soso	0.5
Youdao	0.5

subtopics.

Besides search engines, we also use the corpus of Wikipedia. In Wikipedia, a term can be associated with more than one Wikipedia topic. There are a lot of disambiguation pages to resolve this kind of conflicts. Different meanings of an ambiguous term are listed on the disambiguation page. We check each NTCIR-9 INTENT topic in Wikipedia. If it has a disambiguation page, the topics on the page would be regarded as candidate subtopics. We also review all the terms in Wikipedia and find out the terms which contain a topic of INTENT task as a substring (for example, the INTENT topic "巧克力" is a substring of a Wikipedia term "白巧克力"). These terms are considered as candidate subtopics, too. We combine the candidates using the vote mechanism discussed above. Since the candidates from Wikipedia are not as reliable as the ones from search engines, the weights of the Wikipedia terms are lower. Specifically, the weight of disambiguation term is assigned as 0.9 and the weight of the terms which contains topics is assigned as 0.4.

2.2 Mining Subtopics From Clickthrough Data

Search engine's query log contains a great deal of information on different users' intents of search queries. We can infer from a user's click log what his intent is behind the query. From large amount of query logs, we are able to find out different intents or subtopics of a query.

Query-URL bipartite graph is usually used for presenting users' clicks on search results[1]. In the bipartite graph, the vertices are queries and URLs. If a user submitted a query and clicked a URL in search result list, there should be an edge which connected the query and the URL. In subtopic mining task, we try to mine and rank subtopics by analyzing the query-URL bipartite graph which is generated from either SogouQ or larger-scale Sogou query log. Figure 1 is an example of a Query-URL bipartite graph. In order to express our method more clearly, the given topic q and candidate subtopics q_1, q_2⋯are placed in different sides of clicked URLs. The weights of the edge stand for numbers of the clicks in query log.

The bipartite graph in Figure 1 shows a given query q, all the queries $\{q_i\}$ which have common clicked URLs with q and the common URLs. Each q_i is regarded as a candidate subtopic. Define $Score(q, q_i)$:

$$Score(q, q_i) = \sum_j \frac{W(q, U_j)}{\sum_k W(q, U_k)} \times \frac{W(q_i, U_j)}{\sum_k W(q_i, U_k)}$$

For example, in Figure 1,

$$\sum_k W(q, U_k) = 3 + 2 + 1 + 1 = 7,$$

$$\sum_k W(q_2, U_k) = 3 + 1 = 4.$$

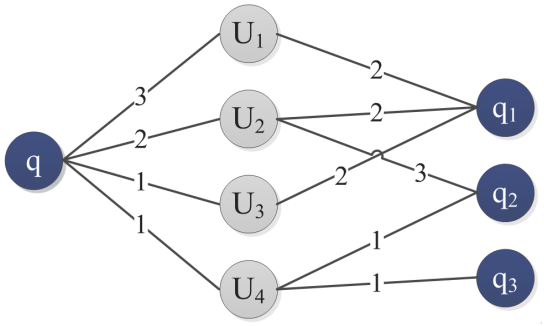

Figure 1: An example of Query-URL bipartite graph.

So,

$$Score(q, q_2) = \frac{2}{7} \times \frac{3}{4} + \frac{1}{7} \times \frac{1}{4} = \frac{1}{4}.$$

The meaning of $Score(q, q_i)$ is the probability that user clicks the same URL when searching different query q and q_i. This score is able to reflect the relevance of two queries. With this score, we rank the candidate subtopics and get a ranked list. As mentioned in Section 2.1, the related queries may also be a generalized topic or a parallel concept of the given topic. To solve this problem, we filter out the candidate subtopic q_i of topic q if:

- q is a substring of q_i, or
- q and q_i have no common word.

Because of the limitation of query log's quantity in experiment, the number of subtopics we get by this method may be not sufficient for submission. In order to generate more subtopics, we extract related anchor texts in SogouT and append the anchors after the subtopic list.

2.3 Removing Reduplicate Subtopics

When a ranked list of subtopics is generated, there are some similar items which are literally different, but describe a same subtopic. It is necessary to find out the reduplicate subtopics and remove the redundant items from the subtopic list so that we can show more different subtopics in top 10 or top 20 results.

Our reduplicate subtopic removing algorithm is also based on query-URL bipartite graph. For each given topic, we construct a query-URL bipartite graph G_{q-U} which contains all subtopics. If subtopic query q_1 and q_2 have a common clicked URL U1 in query log, there should be edge (q_1, U_1) and (q_2, U_1) in G_{q-U}. In other words, there is a path $< q_1, U_1, q_2 >$ between q_1 and q_2. We call this kind of path which connects two query q_1 and q_2 by a URL a q-U-q path. Obviously, a pair of queries may have several different q-U-q paths.

According to the bipartite graph G_{q-U}, we construct a new graph G. The nodes in G are all the queries in G_{q-U}. In G, there is an edge connecting q_1 and q_2, if and only if there is at least a q-U-q path connecting them in G_{q-U}. The weight of edge (q_1, q_2) indicates the number of q-U-q paths from q_1 to q_2 in G_{q-U}.

It is reasonable to believe that the greater the weight of an edge is, the more similar two queries are. Since we would

like to remove reduplicate subtopics, we need to find out the subtopics with high similarity. We remove the edges with weight<4, and get a new graph G'. So, if two queries are connected in G', they should have at least 5 common clicked URLs. These connected queries should be very similar. Now the graph G' consists of several connected components. According the analysis before, the queries in the same connected component describe very similar topic and can be regarded as reduplicate subtopics. For the queries in a same connected component, we only keep the one with the highest rank, and remove the others from the subtopic list.

2.4 Re-ranking Based on Clicked Titles and Snippets

Another important algorithm we used in our runs is re-ranking the subtopics based on clicked titles and snippets on SERPs.[4] Search engines usually present the search results by the title of webpage with snippet text. These texts are the only channel for users to learn about the webpage before they click the search result link. So the content in clicked title and snippet text can reflect the most important facets of the search query. This section will introduce our subtopic re-ranking algorithm based on analyzing the titles and snippets of clicked search results.

Firstly, we crawl the top 5 SERPs of the given topic, and extract the title and snippet text of each search result. Then we extract all search result clicks of this topic in top 5 SERPs. We gather all the clicked snippets and titles into a "snippet document". If a search result is clicked n times, its snippets and titles will appear n times in the snippet document. To estimate the intents of the topic, we try to find the reprehensive component in the snippet document. We eliminate all stop words in the snippets and calculate the frequencies of the terms in the remaining part. This term list with frequencies represents the meanings and intents of original topic. We rank all the terms by their frequencies in descending order and assign a score to each term. The score of top rank is 1 and the last rank is 0.5. The scores of the other terms are uniformly distributed between 1 and 0.5. Then we look back into our subtopic list with rank scores. For each term in the subtopic, we add the term's score (which we have assigned according to the term frequency rank) to the current score of the subtopic. After we process all the subtopics, the rank scores are all updated. With the new rank scores, we re-rank the subtopics and get a new subtopic list which has considered the users' concern of the topic.

2.5 Find Main Intents of Topics

In our work of analyzing search engine's query log, we have found that there are 4 types of common needs in Chinese search engines: online music, online video, online novel and encyclopedia. Each type may be corresponding to a latent subtopic for some topics. For example, "Britney Spears" is a query with the "online music" intent. So "Britney Spears' Music" or "Britney Spears' Song" should be an important subtopic. For each type of needs, there are some popular websites which cover most of the needs. We can get the websites list from an Internet directory website. So we can get 4 lists of websites which are corresponding to 4 types of needs. For a given topic in NTCIR-9 INTENT task, we extract all the clickthrough data in query log and classify the clicked URL by their websites. If any of the 4 types of websites occupies an important part of the clicked URLs, we call it the main intent of the query and improve the ranking of related subtopic. This method is applied on all given topics, and finally affects about 30% of them. The whole process is automatic and rule-based.

3. DOCUMENT RETRIEVAL AND DIVERSIFICATION

3.1 Retrieval Models and Dataset

3.1.1 *Improved probabilistic model in our retrieval system*

In retrieval step, probabilistic model is leveraged for document ranking, which is based on BM25[5] and combined with our previous proposed word pair model [8]:

$$R(Q,D) = W_{BM25} + \alpha \cdot W_{wp}$$

$$W_{BM25} = \sum_{i=1}^{m} log(\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_i\cdot(1-b+b\cdot|D|/avgdl)})$$

$$W_{wp} = \sum_{i=1}^{m} log(\frac{N-n(q_iq_{i+1})+0.5}{n(q_iq_{i+1})+0.5} \cdot \frac{f(q_iq_{i+1},D)\cdot(k_1+1)}{f(q_iq_{i+1},D)+k_i\cdot(1-b+b\cdot|D|/avgdl)})$$

$BM25(Q,D)$ is the traditional ranking model, W_{wp} is defined as the sum of the BM25 relevancy between the document and each phrase formed by two contiguous words in the original query and α is the combination weights for word pair modal. N is the total number of documents, $n(q)$ is the number of documents contain q. k_1 and b are experimental parameters of BM25 ranking. $|D|$ is the length of document D, $avgdl$ is the average document length, $f(q,D)$ is the term frequency of q in D.

3.1.2 *Dataset and retrieval strategy*

SogouT dataset contains 3 parts: the text content of each document (called "Content"), the anchor information of each document (called "Anchor") and the click information of each document (called "Click"). To examine the retrieval effect of each part, we build index and training models for every part separately. So given a query, we can generate 3 relevant document lists (document amount of each list is up to 1000). To generate the final retrieval results, we use the linear combination of 3 lists:

$$score(D) = \sum \omega_i \cdot score_{List(i)}(D)$$

If list i does not contain document D, we just assign 0 to the score. Then we can rank all the documents according to the score and choose top 1000 as the final result list. The experimental parameters generating from training set are shown in Table 2.

Table 2: The experimental parameters generating from training set.

part	α_1	k_1	b	ω
Content	0.2	1.2	0.55	0.2
Anchor	0.1	0.9	0.3	0.5
Click	0.1	1.6	0.3	0.3

3.2 Result Diversification

3.2.1 Documents Duplication Elimination

When our team took part in TREC 2009, Documents Duplication Elimination was applied to diversify the search results as an independent method [2], while it was conducted in coordination with HITS in TREC 2010[3]. In this paper, this method is also used independently to check whether it is useful to diversify the Chinese documents. As described in [2], cosine similarities between every two documents are calculated. They form an upper triangular matrix A_{ij}(its element a_{ij} represent the similarity between document i and j, where $i < j$). Then document j satisfying $a_{ij} > \theta$ is eliminated. In our run, θ is set to 0.4.

3.2.2 Novelty-Result Selection algorithm

To make top documents cover as many diverse information needs as possible, Novelty-Result Selection directly diversifies the search results [3]. The main idea is: when deciding the candidate document at position k, we select a document which could introduce the most novel information despite of all the results before position k. There are two assumptions for this method. One is that in the given search results, all the documents are of high relevance to the query and ranked by the probability that they could satisfy user's information needs under the query. The other is that, the search results can cover various information needs the user might have, regardless of the position of each documents. So we do not need to search for documents which satisfy various information needs of the query, but re-rank the documents to better cover diverse information needs in the top results [3]. More details could be found in [3].

3.2.3 D#-nDCG-based Selection algorithm

In [6], the author proposed a new metric called D#-measure to evaluate the diversity of search results. It could solve the undernormalization problem of the IA metrics and also includes a mechanism to significantly boost intent recall [7]. This is because D#-measure gives a global gain for every document in the result list, which is different from the IA metrics' local gain for every intent of a query. In this paper, based on D#-nDCG, we propose a selection algorithm called D#-select to diversify the search results: for a query q, its intent set I and respective weights W could be found in the subtopic mining task. Then documents for every intent are retrieved separately. At last, D#-select is used to diversify the result. To better explain D#-select algorithm, we define:

$$p(i|q) = \frac{w_i}{\sum_i w_i}$$

where w_i stands for the weight of the ith intent. $p(i|q)$ is the possibility of an intent for query q.

$$g_i(d) = \begin{cases} 5, & r_d \in [1,5] \\ 4, & r_d \in [6,20] \\ 3, & r_d \in [21,50] \\ 2, & r_d \in [51,100] \\ 1, & r_d \in [101,1000] \end{cases} \quad (1)$$

r_d is the original rank of document d, and $g_i(d)$ is the gain of d. Then the process of D#-select can be described as follows:

Given q, I, D, S
if |I|>3 **then**
  for every d in D **do**
    $GG(d) = \sum_i Pr(i|q)g_i(d)$
    $C_i(d) = p(i|q) \cdot \sum_{k=1}^{r} g_i(d)$
  end for
  while $|S| < 10000$ **do**
    for every d in D **do**
      $I - rec(d) = \sum_i g_i(d) \cdot (1-\alpha)^{c_i(r-1)}$
      $D\#value(d) = \gamma I - rec(d) + (1-\gamma)GG(d)$
      Add $maxD\#Value(d)$ to S, then delete it in D
    end for
  end while
  Return S
else
  Return D
end if

where I is the intent set of q. D is the search result collection of intent set I, and S is the re-ranked list. $I - rec(d)$ stands for the recall how much the documents in S cover the intents in I.

3.2.4 D#-nDCG-based selection+user browse logs

In this experiment, user browse logs are leveraged for document diversification. The browse graph is built based on the filtered Sogou toolbar logs of 2008, when the SogouT was crawled, and then PageRank is calculated on this graph. At last, the result list is re-ranked by the PageRank value.

3.2.5 Result re-ranking with HITS

In TREC 2010, HITS was adopted to re-rank the baseline search results in both AdHoc and Diversity task [3]. Top m documents sorted by either Authority or Hub Value are placed up to the front. Its new rank is determined as follows:

$$R_{new} = R_{old} - R_{old} \times (Authority + Hub)$$

where R_{new} stands for the new rank of the document, and R_{old} is the old one. As in [3], m is set to 40 according to the training results in the TREC 2009 and TREC 2010 diversity task, because top 40 is a stable choice for ERR-IA value. It is proved that HITS could stably improve the diversity of the search result [3]. But in TREC, the corpora (Both Collection A and B) are in English. So this year we apply HITS on the SogouT, which is a collection of Chinese documents, to see whether it also could do a good job.

4. SUBMITTED RESULTS

4.1 Subtopic Mining

In subtopic mining task, we submitted five runs. The SYSDESC fields and the approaches of the runs are shown in Table 3. For comparison, we give another run which only use the votes of search engines and Wikipedia.

Table 4 shows the I-rec@10, D-nDCG@10, D#-nDCG@10 values of the runs. From the evaluation results, we can find that the related queries from search engines are very useful for subtopic mining task. The reduplicate subtopic removing method can improve the recall in top ranks, but the D#-nDCG@10 value decreases when reduplicate subtopics are removed. The re-ranking based on clicked snippets and titles shows to be helpful for enhancing D#-nDCG measure.

Table 3: Runs and SYSDESC fields in subtopic mining subtask.

Run name	SYSDESC field	Applied methods in
THU-S-C-1	Hints from Search Engines with user needs re-rank, removing reduplicate ones with Qurey-Url graph model	Section 2.1, 2.3 and 2.5
THU-S-C-2	Hints from Search Engines with user needs re-rank, removing reduplicate ones with Qurey-Url graph model, re-ranking based on snippets and titles of pages	Section 2.1, 2.3, 2.4 and 2.5
THU-S-C-3	Hints from Search Engines with user needs re-rank	Section 2.1 and 2.5
THU-S-C-4	Topics generated based on the log, using query-url model. Appended with anchor text according to retrieved documents.	Section 2.2
THU-S-C-5	Topics generated based on large logs, using query-url model. Appended with anchor text according to retrieved documents.	Section 2.2
THU-S-C-comp	Hints from Search Engines	Section 2.1

Table 4: Evaluation results of subtopic mining runs.

Run name	I-rec@10	D-nDCG@10	D#-nDCG@10
THU-S-C-1	0.4946	0.6896	0.5921
THU-S-C-2	0.4801	0.7186	0.5993
THU-S-C-3	0.4828	0.7107	0.5967
THU-S-C-4	0.2654	0.4040	0.3347
THU-S-C-5	0.2888	0.4455	0.3672
THU-S-C-comp	0.4835	0.7109	0.5972

THU-S-C-4 and THU-S-C-5 are based on Sogou's query log and SogouT corpus. They do not perform so well as the other three submitted runs. The only difference between the two runs is that THU-S-C-4 only use the query log in SogouQ dataset while THU-S-C-5 use more query log. From the comparison of them, we can see that using more click-through data can make the subtopics more reliable.

4.2 Document Ranking

In the document ranking task, we totally submit five runs, all of which are created automatically by programs. The evaluation results and their descriptions are listed in Table 5. To get a compare with the baseline, we also evaluate the original result, and list the values in Table 5. From these values, we can find that HITS does get a great promotion in diversifying documents. Specifically, on the I-rec which evaluates the recall of subtopics, it gets an increase by 33% at most. Even more, on the D-nDCG which stands for the diversification level of the search results, it improves by 54% at most. Because the weight γ in D#-measure is 0.5, increases on I-rec and D-nDCG can significantly promote the final D#-measure equally. At this moment, we can conclude that HITS could diversify documents not only in English but also in Chinese. Furthermore, that is because of the significant promotions on both subtopic recall and result diversification. Instead, Novelty-Result Selection and Documents Duplication Elimination get some decrease based on the HITS. But Documents Duplication Elimination could get a better subtopic recall than HITS. Indeed, eliminating documents that are content-similar in the result list equals to re-ranking documents for a better cover of the subtopics. At last, the D#-select algorithms get the worst results. They directly select the retrieved documents which are not re-ranked by HITS. But they also get a bit improvement on D#-nDCG, which is caused by the promotion of I-rec. This implies that document diversification could benefit from the explicit calculation of subtopic recall, which has been rarely cared by the existing diversification algorithms.

5. CONCLUSION AND DISCUSSION

Understanding users' search intents is a very interesting research topic. In the subtopic mining subtask, we find that:

- For a general query, the related queries from different search engines could show most of the primary intents or subtopics.
- It is feasible to mine subtopics from query log. But for good performance, very large amounts of query log are needed, especially when the query is not so popular.
- Our method of removing reduplicate subtopics is effective for improving the recall of intents at top ranks. But some useful subtopics are missed because of the arbitrary removing. Putting the reduplicate subtopics at lower ranks should be a better strategy.
- The subtopic re-ranking algorithm based on clicked snippets and titles shows very good performance for improving D-nDCG and D#-nDCG values.

The evaluations of our runs in the document ranking task show that both HITS and Documents Duplication Elimination get a stable promotion in diversifying Chinese documents, while the Novelty-Result Selection algorithm gets a bit worse. And the D#-nDCG-based selection algorithms perfom worst in all of our submitted runs, because in this method, not only subtopics need to be mined but also the weight of each subtopics should be evaluated. If any of these is different from the answer, nothing would benefit from the selection algorithm. So the methods that do not explicitly attempt to diversify the result list may get a better result.

6. ACKNOWLEDGMENTS

We would like to thank Junwei Miao and Ting Yao for their effort on improving our methods.

7. REFERENCES

[1] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 407–416, New York, NY, USA, 2000. ACM.

Table 5: Evaluation results of document ranking runs.

	I-rec@10	D-nDCG@10	D#-nDCG@10	Description
THUIR-D-C-1	0.6893	0.4542	0.5717	Documents Duplication Elimination.
THUIR-D-C-2	0.6495	0.3853	0.5174	Novelty-Result Selection algorithm.
THUIR-D-C-3	0.5979	0.2598	0.4288	D#-nDCG-based selection algorithm.
THUIR-D-C-4	0.6001	0.2569	0.4285	D#-nDCG-based selection+user browse logs.
THUIR-D-C-5	0.6861	0.4573	0.5717	Result re-ranking with HITS.
Baseline	0.5157	0.2967	0.4062	the original retrievaled results.

[2] Z. C. Li, F. Chen, Q. L. Xing, J. W. Miao, Y. F. Xue, T. Zhu, B. Zhou, R. W. Cen, Y. Q. Liu, M. Zhang, Y. J. Jin, and S. P. Ma. Thuir at trec 2009 web track: Finding relevant and diverse results for large scale web search. In *in Proceedings of the twelfth Text REtrieval Conference (TREC 2009)*, 2009.

[3] Z. C. Li, Q. Fang, B. Zhou, F. Chen, Q. L. Xing, T. Zhu, Y. Q. Liu, M. Zhang, Y. J. Jin, and S. P. Ma. Thuir at trec 2010 web track: revisiting the use of anchor text and finding novel results for diversification. In *in Proceedings of the twelfth Text REtrieval Conference (TREC 2010)*, 2010.

[4] Y. Liu, J. Miao, M. Zhang, S. Ma, and L. Ru. How do users describe their information need: Query recommendation based on snippet click model. *Expert Systems with Applications*, 38(11):13847 – 13856, 2011.

[5] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[6] T. Sakai, N. Craswell, R. Song, R. S., Z. Dou, and C.-Y. Lin. Simple evaluation metrics for diversified search results. In *In Proceedings of EVIA 2010*, pages 42–50, 2010.

[7] T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '11, 2011.

[8] M. Zhang, C. Lin, Y. Liu, L. Zhao, and S. Ma. Thuir at trec 2003: Novelty, robust and web. In *in Proceedings of the twelfth Text REtrieval Conference (TREC 2003)*, pages 556–567, 2003.