THUIR at NTCIR-9 INTENT Task

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

THUIR

Subtopic Mining

Extracting Subtopics from Web Resrouces

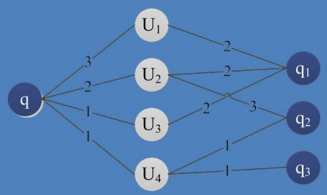
In our experiment, we try to extract subtopics from the query recommendations of comercial search engines and Wikipedia items. We decide a subtopic's rank by its appearance in different places.

Resource		Weight
Query Recommendation of	Google	1
	Bing	1
	Baidu	1
	Sogou	1
	Youdao	0.5
	Soso	0.5
Wikipedia Item from	Disambiguition Page	0.9
	Other	0.4

- Most recommended queries in SERPs are subtopics of the original query.
- If a recommended query contains no more terms than the original query, it should be ignored.

Mining Subtopics from Clickthrough Data

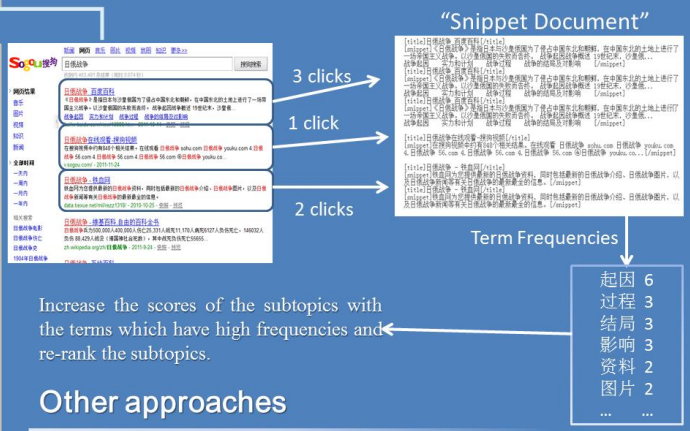
This graph is an example of a bipartite graph which is constructed by clickthrough data. In the graph, q and q_is are queries, and U_js are URLs. The edges between them stand for users' clicks. Define *Score(q,q_i)*:



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

The meaning of *Score(q, q_i)* is the probability that user clicks the same URL when searching q and q_i. This score is able to reflect the relevance of two queries. Then we rank the queries by the score.
Since we only want subtopics, the q_i in our final result should not be a substring of q, and there must be a common term between them.

Re-ranking Based on Clicked Titles and Snippets



Increase the scores of the subtopics with the terms which have high frequencies and re-rank the subtopics.

Other approaches

- We find synonymous subtopics by analyzing the clickthrough data and remove reduplicate subtopics from our runs.
- Specificially, we try to recognize four kinds of common intents of topics: Online Music, Online Video, Online Novel and Encyclopedia. If we find that a topic has one of the above intents, we will improve the rank of related subtopics.

Experimental Results

Run name	SYSDESC field	D#-nDCG@10
THU-S-C-1	Hints from Search Engines with user needs re-rank, removing reduplicate ones with Qurey-Url graph model	0.5921
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	0.5993
THU-S-C-3	Hints from Search Engines with user needs re-rank	0.5967
THU-S-C-4	Topics generated based on the log, using query-url model. Appended with anchor text according to retrieved documents.	0.3347
THU-S-C-5	Topics generated based on large logs, using query-url model. Appended with anchor text according to retrieved documents.	0.3672
THU-S-C-comp	Hints from Search Engines	0.5972

Document Ranking

Documents Retrieval Models

Probabilistic model is leveraged for document ranking, which is based on BM25 and combined with our previous proposed word pair model.

$$R(Q,D) = W_{BM25} + \alpha_1 \cdot W_{wp}$$

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

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

Result re-ranking with HITS

Top m documents sorted by either Authority or Hub Value in the search result are placed up to the front. Its new rank is determined as follows:

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

D#-nDCG-based Selection algorithm

Define the probability of intent i for query q as $p(i|q)$, and the gain of document d under intent i as $g_i(d)$:

$$p(i|q) = \frac{w_i}{\sum_i w_i} \qquad g_i(d) = \begin{cases} 5, & r_d \epsilon [1,5] \\ 4, & r_d \epsilon [6,20] \\ 3, & r_d \epsilon [21,50] \\ 2, & r_d \epsilon [51,100] \\ 1, & r_d \epsilon [101,1000] \end{cases}$$

Then the algorithm can be described as follows:

```
Given q, I, D, S
if |I|>3 then
  for every d in D do
    GG(d) = Σ_i Pr(i|q) g_i(d)
    C_i(d) = p(i|q) · Σ_{k=1}^{i} g_i(d)
  end for
  while |S|<1000 do
    for every d in D do
      I − rec(d) = Σ_i g_i(d) · (1 − α)^{c_i(r−1)}
      D#value(d) = γI − rec(d) + (1 − γ)GG(d)
      Add max D#Value(d) to S, then delete it in D
    end for
  end while
  Return S
else
  Return D
end if
```

The novelty of document d_i is defined as follows:

$$f(d_i, S) = \frac{|S|}{\sum_{d_j \epsilon S} \alpha_j \cdot \frac{1}{cos < \omega_i, \omega_j >}}$$

$$\alpha_j = \frac{1}{original\ rank\ of\ d_j}$$

D#-nDCG-based Selection + user browse logs

The browse graph is built based on the filtered Sogou toolbar logs of 2008, and then PageRank is calculated on this graph. At last, the search result is re-ranked by the PageRank value.

Documents Duplication Elimination

For the search result of a query, cosine similarities between every two documents are calculated. They form an upper triangular matrix. Then document j satisfying a_{ij} $(where\ i < j) > 0.4$ is eliminated.

Novelty-Result Selection algorithm

The main idea of this algorithm is: when deciding the candidate document at position k, we select the document which could introduce the most novel information despite of all the results before position k. In this way, we hope to make the top documents in the result cover as many diverse information needs as possible.

Experimental Results

	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.