Time-aware Structured Query Suggestion

Taiki Miyanishi*
Graduate School of System Informatics
Kobe University, Japan
miyanishi@ai.cs.kobe-u.ac.jp

Tetsuya Sakai
Microsoft Research Asia, China
tetsuyasakai@acm.org

ABSTRACT

Most commercial search engines have a query suggestion feature, which is designed to capture various possible search intents behind the user's original query. However, even though different search intents behind a given query may have been popular at different time periods in the past, existing query suggestion methods neither utilize nor present such information. In this study, we propose Time-aware Structured Query Suggestion (TaSQS) which clusters query suggestions along a timeline so that the user can narrow down his search from a temporal point of view. Moreover, when a suggested query is clicked, TaSQS presents web pages from query-URL bipartite graphs after ranking them according to the click counts within a particular time period. Our experiments using data from a commercial search engine log show that the time-aware clustering and the time-aware document ranking features of TaSQS are both effective.

Categories and Subject Descriptors

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

Keywords

time-aware information retrieval, query suggestion

1. INTRODUCTION

Query suggestion is an important feature of web search engines [3, 5, 7, 8]. Together with query autocompletion that presents related queries within the search box in real time [1, 9], a list of query suggestions shown within the search engine result page often helps the user reformulate his query easily and effectively. However, even though different search intents behind a given query may have been popular at different time periods in the past, existing query suggestion methods neither utilize nor present such information. As Li and Croft [6] point out, topically relevant but obsolete documents may not satisfy the user if he is seeking recent information. More importantly, even relevant and recent documents may not satisfy the user if he is interested in a specific time period in the past. Some search engines let the user specify a time range together with his search query. However, in order to enjoy such a feature, the user needs to figure out a correct time range, and to explicitly provide it to the search engine.

Table 1: Example of query suggestions in response to a query "romney 2012"

Flat list	TaSQS
romney momentum	**Oct 01 2012 - Oct 11 2012**
romney surge	romney losing
romney losing	why romney is losing
why romney	romney chance to beat obama
obama romney women poll	lenore romney
romney on the economy	economists pick romney
romney and massachusetts	**Oct 12 2012 - Oct 18 2012**
romney obama 2012	romney west virginia
romney how web	romney on education
romney and binder	google romney
romney and women	**Oct 19 2012 - Oct 28 2012**
women for romney	romney women
romney on economy	romney slogan
romney economy	opinion obama romney
romney debate performance	romney women binder
	women for romney
	Oct 29 2012 - Oct 30 2012
	romney momentum
	Oct 31 2012
	romney indiana

In this paper, we introduce Time-aware Structured Query Suggestion (TaSQS), an algorithm for presenting query suggestions along a timeline and helping the user access relevant web pages. TaSQS leverages query-URL bipartite graphs to determine time ranges with different durations for different query suggestions, and frees the user from the burden of entering a specific time constraint with a query. We call this feature *time-aware query clustering*. Table 1 contrasts a standard "flat-list" query suggestions and the query suggestions that TaSQS presents, based on a commercial search engine log from October 1-31, 2012. If the user selects "romney losing" from the flat-list query suggestions, a typical search engine would just return a web search result for this query. In contrast, if the same query is selected on the TaSQS interface, TaSQS interprets this as the user's interest in this query *for the specific time period of October 1-11, 2012*, and presents corresponding web pages from query-URL bipartite graphs after ranking them according to the click counts within that particular time period. We call this feature *time-aware document ranking*.

The objective of the present study is to validate the time-aware query clustering and time-aware document ranking

*This work was done while the author was at Microsoft Research Asia.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

ACM 978-1-4503-2034-4/13/07 ...$15.00.

features of TaSQS. To this end, we evaluate the retrieval effectiveness of the ranked web pages provided by TaSQS, under the assumption that the user is able to select a query suggestion appropriate for his information need on the TaSQS interface like the one shown in Table 1. Our results show that the time-aware query clustering and the time-aware document ranking are both effective.

2. RELATED WORK

While flat-list query suggestion is the current de facto standard in commercial web search engines, several researchers have proposed to organize and present query suggestions. For example, Sadikov et al. [8] proposed an algorithm to cluster query suggestions based on clickthrough and session data; Guo et al. [3] proposed a method to provide a label for each query suggestion cluster based on social annotation data; Kato et al. [5] proposed a query suggestion algorithm that presents labeled query suggestion clusters so that the user can make comparisons across multiple entities (e.g. company names). Clustering queries is probably useful for the user to see the relationship across the suggested queries; labeling each cluster is probably useful for the user to see what each cluster represents and how it is related to the original query. However, none of these structured query suggestion methods provide a temporal point of view.

Shokouhi and Radinsky [9] proposed a time-sensitive query autocompletion method based on the idea of Bar-Yossef and Kraus [1], which ranks candidate queries according to forecasted frequencies rather than past popularity. However, their method is not intended for the problem we are tackling, namely, to organize suggested queries along a timeline and to let the user focus on a particular time range without specifying an explicit time constraint.

3. PROPOSED METHOD

Time-aware Structured Query Suggestion (TaSQS) is an algorithm for presenting query suggestions along a timeline and helping the user access relevant web pages. It consists of four main steps: generating query suggestions, time-aware query clustering, time-aware query selection and time-aware document ranking. We describe these steps below.

3.1 Generating Query Suggestions

In this step, we can plug in any existing algorithm for generating suggested queries for an input query. In the present study, we use a popular query suggestion algorithm proposed by Mei et al [7]. First, we construct a query-URL bipartite graph for each day, where queries and URLs are nodes and the weight of each edge corresponds to the click count $w(q,u,t)$, the number of times URL $u \in \mathcal{U}$ is clicked by users with query $q \in \mathcal{Q}$ within a time period t (one day in our case). For every query pair, we compute the hitting time of a random walk, defined as the expected number of steps it takes from a query to another on the bipartite graph, which indicates the "closeness" between the given query and a query suggestion candidate. We take the top L queries based on hitting time and use them as query suggestions: we let $L = 1000$ for all of the methods examined in this study.

To obtain the relevance score of a query suggestion, we shift and normalize the hitting time into the $[0,1]$ range and then subtract it from 1. The resultant relevance score for the initial query q, a query suggestion q' for a time period t is denoted by $\widetilde{r}(q,q',t)$. Furthermore, to smooth the relevance scores, we use the moving average using ϕ preceding days for each t as follows: $r(q,q',t) = \frac{1}{\phi}\sum_{i=0}^{\phi-1}\widetilde{r}(q,q',t-i)$. In this paper, we let $\phi = 3$ based on a pilot experiment.

Algorithm 1 TA-Clustering(M)

Input: A set of relevance score vectors $\mathcal{I} = \{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\}$
Output: A cluster set $\mathcal{C}$ that contains no more than M clusters

```
C ← ∅
for x_{t_i} ∈ I do
    C ← C ∪ {x_{t_i}}
end for
while (|C| > M) and
        (max_{C_i≠C_j∈C} TA-Similarity(C_i, C_j) > 0) do
    ⟨C_i, C_j⟩ ← argmax_{C_i≠C_j∈C} TA-Similarity(C_i, C_j)
    C_i ← C_i ∪ C_j; C ← C − C_j
end while
return C
```

3.2 Time-aware Query Clustering

To accommodate the users' diverse needs from a temporal point of view, we perform query clustering so that each cluster represents a group of query suggestions that were popular at a particular time period in the past. To this end, we modify a popular *Hierarchical Agglomerative Clustering* (HAC) algorithm that iteratively finds a pair of clusters that are the most similar and merges them.

We denote a vector of relevance scores as $x_t = [r_1^t, r_2^t, \ldots, r_m^t]$, where m is the number of generated query suggestions and each element is the aforementioned relevance score $r_j^t = r(q,q_j,t)$. The input to our clustering algorithm is a set of relevance score vectors $\mathcal{I} = \{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\}$, where n denotes the number of time periods (days in this study) considered by TaSQS. For example, $t_1, t_2, \ldots, t_n$ means October 1-31 when TaSQS clusters the vectors in October. To incorporate time-awareness into the similarity calculation of the HAC algorithm, we use the notion of *temporal adjacency*: for example, a cluster that spans May 10-12 and one that spans May 13-21 are temporally adjacent. Thus, we define the time-aware similarity between two clusters C_i and C_j as follows: TA-Similarity$(C_i,C_j) = \frac{1}{|C_i||C_j|}\sum_{x_k\in C_i}\sum_{x_l\in C_j} x_k \cdot x_l$ if C_i is temporally adjacent to C_j, TA-Similarity$(C_i,C_j) = 0$ otherwise. Thus, only temporally adjacent clusters can be merged. Algorithm 1 shows our time-aware query clustering algorithm, from which we obtain no more than M clusters. In this study, we let $M = 5$, as showing more than five clusters on a web search interface may not be realistic.

3.3 Time-aware Query Selection

After clustering, we need to select query suggestions from the clusters for presentation. These presented query suggestions are used for accessing the corresponding web pages based on the query-URL bipartite graphs, and the time periods associated with the suggestions are used for ranking these pages. Thus we need to select a set of representative and diverse query suggestions from the clusters to accommodate diverse search needs. For this, we rank query suggestions in descending order of the following time-aware score:

$$\text{TA-Score}(q,q') = \lambda R(q,q',C) - (1-\lambda)R(q,q',\bar{C}), \quad (1)$$

where q' is a query suggestion, $\bar{C} = \{x_{t_i} \in \mathcal{I} : x_{t_i} \notin C\}$, $R(q,q',C) = \frac{1}{|C|}\sum_{\{t:x_t\in C\}} r(q,q',t)$, λ is a parameter ($0 <$

Table 2: Examples of search topics used in our experiments. The relevance assessors did not have access to the specialized query and target time fields highlighted in gray.

Initial query	Specialized query	Description	Target time
avengers	scarlett johansson avengers	Find Scarlett Johansson talking about playing Black Widow in "The Avengers"	May 4, 2012
euro 2012	euro 2012 spain vs portugal	Find information about the Euro 2012 semifinal Spain vs. Portugal	Jun. 27, 2012
olympics	missy franklin 2012 olympics	Find Missy Franklin's comments on the Summer Olympics in 2012	Aug. 19, 2012

$\lambda < 1$) that determines the balance between the average relevance score for query suggestion q' over cluster C and that for q' over the complement of C. Thus, the TA-Score for a given query suggestion and a cluster is high when its average relevance score over that cluster is high while its average relevance score over the complement of that cluster is low. We set λ to 0.4 based on a preliminary experiment. In addition, we take the top N query suggestions according to TA-Score for presentation. In this study, we let $N = 15$ (See Table 1).

3.4 Time-aware Document Ranking

The final step of TaSQS is to present a ranked list of web pages in response to the user's click on a suggested query, by leveraging the query-URL bipartite graphs. As a specific time period is tied to each suggested query in the TaSQS interface, we assume that the user is interested in that suggested query *within the context of that particular time period*. More specifically, we assume that the user is interested in documents that were frequently clicked within that specific time period.

We have two methods to rank the web pages u available from the query-URL bipartite graphs. The first is to rank them by the overall *popularity* of the cluster C' to which the selected query suggestion q belongs:

$$\mathrm{POP}(u, q, C') = \sum_{\{t:x_t \in C'\}} w(q, u, t). \tag{2}$$

Recall that $w(q, u, t)$ denotes a click count within the time period t. Based on the above method, "constantly popular" web pages such as Wikipedia are always ranked high. However, what we really want to accomplish is to favor web pages that were particularly popular within the specific time period. To this end, our second method ranks web pages by *relative popularity*:

$$\mathrm{rPOP}(u, q, C') = \frac{\sum_{\{t:x_t \in C'\}} w(q, u, t)}{w(q, u)}, \tag{3}$$

where $w(q, u) = \sum_t w(q, u, t)$.

4. EVALUATION

This section reports on a laboratory experiment for evaluating the time-aware query clustering and the time-aware document ranking features of TaSQS in terms of the retrieval effectiveness of the ranked lists generated by TaSQS.

4.1 Experimental Setup

Click data. We evaluate TaSQS using Microsoft Bing's query log from May 1 to October 31, 2012, where each record consists of a query and a clicked URL. In the preprocessing step, we removed redundant white spaces, punctuations and queries that were longer than 100 characters from the data. URLs more than 300 characters long were also removed. In addition, we filtered out queries and URLs related to adult contents, as well as queries with fewer than five clicks per day. As a result, we obtained 5,348,081 unique queries, 54,445,091 unique URLs, and 75,488,967 query-URL pairs. Moreover, we partitioned the query-URL pairs by day to form query-URL-date triples for each day. The final data set consists of 648,047,630 triples, which has an average of 3,580,374 triples per day.

For generating query suggestions, we constructed the query-URL bipartite graphs for each day and obtained 183 bipartite graphs. Moreover, from each bipartite graph, we extracted a subgraph that consists of queries that contain a term from the initial query. This was used for generating query suggestions for all of the methods examined in this paper.

Search topics. To evaluate the effect of our proposed time-based ranking method, we constructed 47 search topics which consists of four parts: initial query, specialized query, description, and target time. Table 2 shows a few examples of the search topics used in our experiments. For example, the first entry in the table represents a search scenario where the user is looking for information from around May 4, 2012 on Scarlett Johansson talking about playing Black Widow in the movie "The Avengers" and starts out with the initial query "avengers." What we want to evaluate in this study is the retrieval effectiveness of the ranked list of web pages provided by TaSQS, assuming that the user actually clicks the specialized query in the query suggestion interface.

We constructed the search topics as follows. First, to construct realistic search topics that align well with our click data from 2012, we selected queries that resulted in a click on a Wikipedia page whose title contained "2012." For example, the initial query "avengers" was selected because it had clicks on the Wikipedia page "The Avengers (2012)." Then, we compared the monthly frequencies of each initial query between May and October 2012 to determine the "peak month" for it. For example, the peak month for "avengers" was May. We then manually selected a specialized query (i.e. one that subsumes the initial query) from the "torso" queries within the peak month, and defined its target time as the day when the specialized query was most frequently used within the peak month. In our example, "scarlett johansson avengers" was selected as the specialized query, and its target time is defined as May 4. Finally, we manually formulated the description fields by examining some web pages that were frequently clicked during the target time.

Relevance assessments. To conduct retrieval effectiveness evaluation, we hired nineteen assessors with computer science skills as relevance assessors. We pooled top five web pages from all of the methods described in Section 4.2 for each of our search topics, and obtained a total of 716 documents to be judged. Each of the pooled documents were judged by two different assessors on a two-point scale (relevant or nonrelevant); the inter-assessor agreement was moderate (Cohen's kappa: 0.501). Finally, we consolidated the

Table 3: IR performances by different methods. The best performing run is indicated in bold and statistically significant differences are marked using the symbols in the top-right corner of each method name.

Method	nDCG@1	ERR@1	RR@1	nDCG@3	ERR@3	RR@3	nDCG@5	ERR@5	RR@5
POP$^{\clubsuit}$	0.624	0.117	0.809	0.640	0.201	0.887	0.645	0.235	0.887
GOOGLE$^{\diamondsuit}$	0.652	0.121	0.809	0.726	0.223	0.879	0.723	0.257	0.890
EqualSplit + rPOP$^{\heartsuit}$	0.709	0.133	0.809	0.681	0.217	0.89	0.700	0.255	0.894
TA-Clustering + POP$^{\spadesuit}$	0.660	0.124	0.830	0.637	0.202	0.901	0.662	0.239	0.901
TA-Clustering + rPOP	**0.780**$^{\clubsuit}_{\diamondsuit}$	**0.146**$^{\clubsuit}_{\diamondsuit}$	**0.936**$^{\clubsuit\heartsuit}_{\diamondsuit}$	**0.740**$^{\clubsuit}_{\spadesuit}$	**0.234**$^{\clubsuit}_{\spadesuit}$	**0.954**$^{\diamondsuit}_{\heartsuit}$	**0.744**$^{\clubsuit}_{\spadesuit}$	**0.271**$^{\clubsuit}_{\spadesuit}$	**0.954**$^{\heartsuit}$

two sets of assessments to form graded relevance assessments as follows: 2=highly relevant (judged relevant by both assessors); 1=relevant (judged relevant by just one assessor); and 0=nonrelevant (judged nonrelevant by both assessors).

Evaluation measures. To evaluate retrieval effectiveness, we use *normalized discounted cumulative gain* (nDCG) [4], *expected reciprocal rank* (ERR) [2] and *reciprocal rank* (RR). nDCG and ERR consider graded relevance while RR does not. Thus RR treats both highly relevant and relevant documents as "just relevant." We discuss statistical significance of results using a two-tailed paired t-test with $p < 0.05$ throughout this paper.

4.2 Baselines

TaSQS first conducts time-aware query clustering, and if a resultant query suggestion is clicked by the user, it ranks the search results based on the popularity or the relative popularity (Eqs. 2 and 3). We denote these two versions as TA-Clustering + POP and TA-Clustering + rPOP, respectively.

To evaluate TaSQS, we also prepared several baseline methods. Our first baseline, POP, does not involve query clustering, but ranks retrieved documents based on the popularity (i.e. click count) statistics from May 1 to October 31. This can be compared with TA-Clustering + POP which uses the popularity information *within the time period to which the clicked query belongs*. Our second baseline, GOOGLE, also does not involve query clustering, but simply uses the Google Custom Search API[1] with May 1-Oct 31 as the time constraint. Finally, our third baseline, EqualSplit + rPOP, divides each month equally instead of applying time-aware clustering. For example, October is divided into 1-6, 7-12, 13-18, 19-24, and 25-31 as five clusters. By comparing this with TA-Clustering + rPOP, we can quantify the benefit of flexibly determining time periods based on time-aware query clustering.

For each method, we assume that the user has clicked the specialized query of each topic, and evaluate the corresponding ranked list of web pages. For TA-Clustering + POP, TA-Clustering + nPOP and EqualSplit + nPOP, we assume that the user is able to select the correct "peak month" (i.e. the month that covers the target time of the topic) and a cluster that covers the target time among five clusters using the query suggestion interface: the responsibility of the methods being evaluated is to present query suggestions along the timeline within that month (e.g. October in Table 1) and to present ranked web pages when one of the suggestions in that cluster is clicked.

4.3 Experimental Results

Table 3 shows the nDCG, ERR and RR performances of the five methods at document cutoffs 1, 3 and 5, with statistical significance test results. It can be observed that TA-Clustering + rPOP significantly outperforms other methods. More specifically, it significantly outperforms POP and GOOGLE in terms of many evaluation measures, which suggests that the combination of time-aware query clustering and document ranking based on the relative popularity is effective. Moreover, it significantly outperforms EqualSplit + rPOP in terms of RR at cutoffs 1, 3 and 5, which suggests that our time-aware query clustering approach which determines variable-length time periods is effective. Furthermore, as it significantly outperforms TA-Clustering + POP in terms of nDCG and ERR at cutoffs 3 and 5, it appears that the relative popularity is more effective than the raw popularity: documents clicked frequently within a particular period in time are more useful than those that are constantly popular (See Eqs. 2 and 3).

[1]https://developers.google.com/custom-search/

5. CONCLUSION

In this study, we evaluated the retrieval effectiveness of ranked web pages produced by Time-aware Structured Query Suggestion (TaSQS), which first clusters query suggestions from a temporal point of view and then presents web pages from query-URL bipartite graphs after ranking them according to their popularity within a specific time period. Through retrieval effectiveness evaluation, we showed the effectiveness of time-aware query clustering (i.e. obtaining flexible time periods for different query suggestions) and that of time-aware document ranking (i.e. presenting web pages based on their popularity within a specific time period). In addition, TaSQS outperformed simple baselines including Google search with a time constraint. We plan to evaluate TaSQS in the context of real user search tasks in future work.

6. REFERENCES

[1] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *WWW*, pages 107–116, 2011.

[2] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM*, pages 621–630, 2009.

[3] J. Guo, X. Cheng, G. Xu, and H. Shen. A structured approach to query recommendation with social annotation data. In *CIKM*, pages 619–628, 2010.

[4] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *TOIS*, 20(4):422–446, 2002.

[5] M. P. Kato, T. Sakai, and K. Tanaka. Structured query suggestion for specialization and parallel movement: effect on search behaviors. In *WWW*, pages 389–398, 2012.

[6] X. Li and W. Croft. Time-based language models. In *CIKM*, pages 469–475, 2003.

[7] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *CIKM*, pages 469–478, 2008.

[8] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. Clustering query refinements by user intent. In *WWW*, pages 841–850, 2010.

[9] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *SIGIR*, pages 601–610, 2012.