Search-based Query Suggestion

Jiang-Ming Yang†, Rui Cai†, Feng Jing‡, Shuo Wang†, Lei Zhang† and Wei-Ying Ma†
† Microsoft Research Asia
{jmyang, ruicai, shuowang, leizhang, wyma}@microsoft.com
‡ NeoGrid Inc
scenery_jf@hotmail.com

ABSTRACT

In this paper, we proposed a unified strategy to combine query log and search results for query suggestion. In this way, we leverage both the users' search intentions for popular queries and the power of search engines for unpopular queries. The suggested queries are also ranked according to their relevance and qualities; and each suggestion is described with a rich snippet including a photo and related description.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*query formulation, search process*

General Terms

Algorithms, Design, Experimentation

Keywords

Query suggestion, query representation, log session mining

1. INTRODUCTION

In Web search, users tend to use terms as queries but not natural language. And they seldom adopt advanced options such as Boolean operators but would like to refine the queries themselves. Therefore, the query suggestion is very necessary to help users formulate queries. Actually, query suggestion has been considered as a must-have feature for search engines, as well as an active topic in academic research. Existing query suggestion approaches could be classified into two categories based on the data they used. One is log-based [2, 3] and the other is search result-based [1]. Query log-based approach and search result-based approach have their own merits and weaknesses, which make them suitable for different kind of queries. This observation motivates us to investigate a new method by integrating both of them in a real system.

2. SEARCH-BASED QUERY SUGGESTION

To provide suggestions for both popular and unpopular queries, we introduce a unified feature representation for queries. Both the corresponding search results and query log sessions are leveraged as the context information. Then, to rank the suggestions, both the similarities and query frequencies are calculated, which are similar to dynamic ranks and static ranks (PageRank) of Web pages respectively. The process of query suggestion could be seen in Fig. 1.

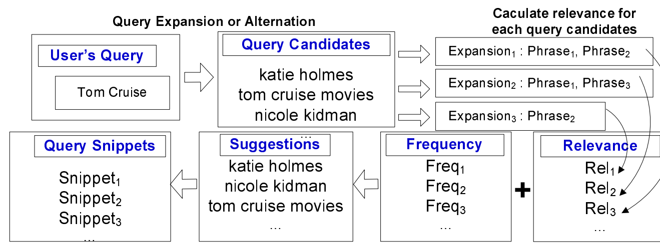

Figure 1: The flowchart of search-based query suggestion which is similar to web search.

2.1 Query Representation

In the following, we will introduce two kinds of context information: search results and query log sessions.

2.1.1 Search Result Context

Instead of using whole Web pages, only the title and snippet of the Web pages from top search results are used to generate keywords. A query q_i could be represented as:

$$\mathcal{F}^{sr}_{q_i} = (r^{sr}_{i,1}, r^{sr}_{i,2}, \cdots, r^{sr}_{i,M}) \tag{1}$$

where $r^{sr}_{i,j}$ represents the relevance between q_i and the j^{th} phrase and M is the number of all possible phrases in a given language such as English:

$$r^{sr}_{i,j} = \frac{D_{i,j}}{\sum_j D_{i,j}} \cdot log \frac{|Q|}{1 + |\{q_i : p_j \in SR_i\}|} \tag{2}$$

Considering that some non-popular queries may have only few results, normalized frequency of the words are used in the left part, where p_j is a word, $D_{i,j}$ is the number of occurrences of the considered word p_j in the set of titles and snippets of current q_i. Similar to the tf-idf weighting, we use the multiplication of phrase frequency and inverse query frequency to weight phrases. Q is the query set, SR_i is the search result content corresponding to q_i.

2.1.2 Query Log Session Context

The other context is query log session. The queries cooccurring with the user submitted query in a certain number of query log sessions could be used as key phrases. Queries are related if they appear in a substantial number of user query log sessions (consecutive queries). Query log session-based algorithms leverage the knowledge of query usage history from numerous users. Therefore, useful queries, which

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.
ACM 978-1-59593-991-3/08/10.

Figure 2: Suggestion snippet in our system.

do not contain the original query strings, could be suggested. We also represent q_i as:

$$\mathcal{F}_{q_i}^{log} = (r_{i,1}^{log}, r_{i,2}^{log}, \cdots, r_{i,M}^{log}) \tag{3}$$

where M is the number of all possible phrases. Similarly its weighting function can be expressed as follows:

$$r_{i,j}^{log} = \frac{N_{i,j}}{\sum_j N_{i,j}} \cdot log \frac{|Q|}{1 + |\{q_i : p_j \in log_i\}|} \tag{4}$$

where p_j is a phrase, $N_{i,j}$ is the number of occurrences of p_j in the sessions of current query q_i, Q is the query set, and log_i is the log session content corresponding to query q_i.

2.1.3 Query Representation

We can combine the two parts of phrases for a query q_i together as:

$$\begin{aligned} \mathcal{F}_{q_i} &= \alpha \times \mathcal{F}_{q_i}^{sr} + \beta \times \mathcal{F}_{q_i}^{log} \qquad (5) \\ &= \alpha \times (r_{i,1}^{sr}, r_{i,2}^{sr}, \cdots, r_{i,M}^{sr}) + \beta \times (r_{i,1}^{log}, r_{i,2}^{log}, \cdots, r_{i,M}^{log}) \end{aligned}$$

We use a constant 0.5 for both α and β in the experiment.

2.2 Query Search

For a new submitted query q^{sub}, we can also represent this query through the method presented in Section 2.1 and the relevance score of a suggested query q^{sug} for the user submitted q^{sub} is:

$$score(q^{sug}, q^{sub}) = \delta \times R_{q^{sug}, q^{sub}} + \psi \times \log Q_{q^{sug}} \tag{6}$$

The left part of the score R represents the similarity between query q^{sug} and q^{sub}, and the right part Q represents the quality (popularity) of query q^{sug}. Similar to a search engine, both dynamic rank and static rank are considered and linearly combined as in Web search.

2.3 Suggestion Snippet

Users may not know the relationship between suggestions and original query. Describing the relationship between the suggested query and the input query may help users to further formulate their queries, as shown in Fig. 2.

Based on the proximity strategy for queries in search engine, we submit two query terms together to a search engine. We can also represent this joint query q^{joint} (joint two queries with a blank) similarly. Then the relevance of snippets be expressed as follows:

$$\mathcal{R}(S, q^{joint}) = \sum_i r_i \quad \text{(for all the phrase in } S\text{).} \tag{7}$$

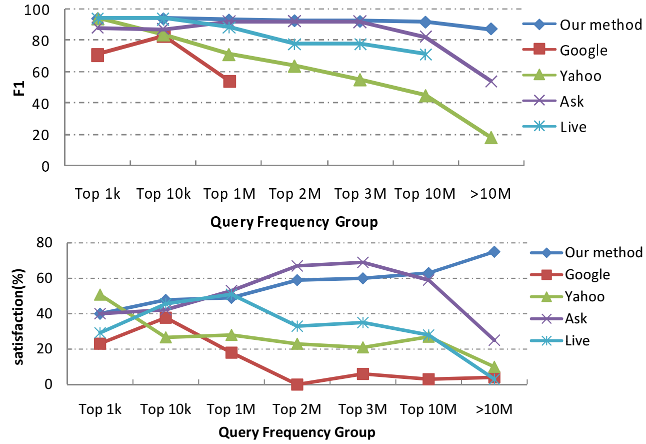

Figure 3: Comparison of the F-measures and satisfactions for different frequency groups and systems.

3. EVALUATIONS

In the experiment, we used 3 month query log data of a commercial search engine in 2007. We divided the top queries into 7 groups based on their frequencies, and randomly sampled 10 queries from each group for testing. Four commercial search engines were selected for the comparison.

Seven subjects were invited to rate the suggestions with 5-point scores ("Precisely related", "Approximately related", "Somehow related, but unclear or useless", "Approximately unrelated", and "Clearly unrelated"). The higher a score is, the more relevant the suggestion is. We randomly selected suggestions of 20 queries for each person. And suggestions of each query were labeled by at least two persons. We transformed the 5-point scores to precision by $(Score-1)/4 \times 100\%$ and recall by $N_i/N_T \times 100\%$, where N_T represents the number of queries in a group and N_i the number of queries with at least one suggestion. The F-measures of different groups are shown in the top part of Fig. 3. Our method consistently has the best performance.

High relevance doesn't necessarily mean better user experience. For example, for "Tom Cruise", suggestions such as "Tom Cruise Picture", "Tom Cruise Photos", and so on are relevant but duplicate. Therefore, diversity and other properties that can not directly reflected by relevance are also important for query suggestion. Therefore, we also conduct a blind test. We built a labeling tool with five columns each corresponds to a suggestion method and fit them into the columns randomly. For each method, only the top 8 suggestions were kept. Users were asked to rank the five results using score $1 \sim 5$. The satisfaction ratio was then calculated as $(Score - 1)/4 \times 100\%$. The result is presented in the bottom part of Fig. 3. The performance of our method is quite good for both top queries and long tail queries.

4. REFERENCES

[1] M. Sahami and T. D. Heilman. A Web-based kernel function for measuring the similarity of short text snippets. In *WWW 2006*, pages 377–386.

[2] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *WWW 2006*, pages 1039–1040.

[3] Q. Zhao, S. C. H. Hoi, and et. al. Time-dependent semantic similarity measure of queries using historical click-through data. In *WWW 2006*, pages 543–552.