Query Session Data vs. Clickthrough Data as Query Suggestion Resources

Makoto P. Kato\textsuperscript{1}, Tetsuya Sakai\textsuperscript{2}, and Katsumi Tanaka\textsuperscript{1}

\textsuperscript{1} Kyoto University, Kyoto, Japan,  
kato@dl.kuis.kyoto-u.ac.jp, tanaka@dl.kuis.kyoto-u.ac.jp  
\textsuperscript{2} Microsoft Research Asia, Beijing, China,  
tetsuyasakai@acm.org

Abstract. Query suggestion has become one of the most fundamental features of Web search engines. Some query suggestion algorithms utilize query session data, while others utilize clickthrough data. The objective of this study is to examine which of these two resources can provide more effective query suggestions. Our results show that query session data outperforms clickthrough data in terms of clickthrough rate.

Keywords: query suggestion, query log, query session, clickthrough, Web search.

1 Introduction

Query suggestion, which enables the user to revise a query with a single click, has become one of the most fundamental features of Web search engines. Providing effective query suggestions to the user is very important for helping him express his information need precisely so that he can access the required information. To this end, some query suggestion algorithms utilize query session data, while others utilize clickthrough data. However, to date, it is not clear which of these two resources can provide more effective query suggestions. Using a large-scale query suggestion usage log from a commercial search engine and some simple query suggestion selection methods, we show that query session data outperforms clickthrough data in terms of clickthrough rate.

2 Related Work

Query suggestion algorithms that rely on query session data include the following. Boldi \textit{et al.} [3] and Anagnostopoulos \textit{et al.} [1] selected query suggestions from queries that are likely to follow a given query. He \textit{et al.} [6] proposed a query suggestion algorithm based on the resemblance between the user’s query sequence and query sequence history.

Whereas, query suggestion algorithms that rely on clickthrough data include the following. One approach is clustering queries based on their clicked URLs, and selecting query suggestions from the query cluster to which the original query belongs [2, 5]. Another approach is incorporating a random walk in a query and clicked URL bipartite graph [7–9].
3 Query Suggestion Selection Methods

To select query suggestions for a given query, we introduce four types of query ranking methods. Given a query \( q \in Q \), each query \( q' \in Q \) is ranked, and the top \( n \) queries are chosen as query suggestions.

3.1 Session-based Ranking Methods

A basic approach to generating query suggestions from query session data is to find queries that often follow or are followed by a given query within the same session as seen in some work [1, 3, 6]. Thus, we used within-session reformulation probability and its symmetric variant to rank queries for a given query by using query session data.

For queries \( q \) and \( q' \), within-session reformulation probability \( P_{\text{session}}(q' | q) \) is defined as follows:

\[
P_{\text{session}}(q' | q) = \frac{f(q, q')}{f(q)},
\]

where \( f(q) \) denotes the number of occurrences of a query \( q \) in the query session data, and \( f(q, q') \) denotes the number of co-occurrences of \( q \) and \( q' \) within the same session, where \( q' \) occurs after \( q \). The probability \( P_{\text{session}}(q' | q) \) can be interpreted as how likely a query \( q' \) follows \( q \) within the same session.

Similarly, symmetric within-session reformulation probability \( P_{\text{sym}}^{\text{session}}(q, q') \) is defined as follows:

\[
P_{\text{sym}}^{\text{session}}(q, q') = \frac{f(q, q') + f(q', q)}{f(q) + f(q')},
\]

which represents how likely a query \( q' \) follows or is followed by \( q \) within the same session.

3.2 Click-based Ranking Methods

A straightforward approach to generating query suggestions from clickthrough data is based on the similarity of URLs clicked in response to a query. Some studies used the clicked-URL similarity [2, 5], while others incorporate a random walk in a query and URL bipartite graph [7–9]. Thus, we used click-based similarity and click-based transition probability for selecting query suggestions from clickthrough data.

For queries \( q \) and \( q' \), click-based similarity \( \text{Sim}_{\text{click}}(q, q') \) is defined as follows:

\[
\text{Sim}_{\text{click}}(q, q') = \frac{\sum_{u \in U} w(q, u)w(q', u)}{\sqrt{\sum_{u \in U} w(q, u)^2} \sqrt{\sum_{u \in U} w(q', u)^2}},
\]

where \( w(q, u) \) represents how many times a URL \( u \) has been clicked in response to a query \( q \), and \( U \) is the whole set of URLs.

Click-based transition probability from \( q \) to \( q' \), \( P_{\text{click}}(q' | q) \), is defined as follows:

\[
P_{\text{click}}(q' | q) = \sum_{u \in U} \frac{w(q, u)}{\sum_{u \in U} w(q, u)} \frac{w(q', u)}{\sum_{r \in Q} w(r, u)},
\]
Table 1. Data statistics

<table>
<thead>
<tr>
<th></th>
<th># of queries</th>
<th># of clicks</th>
<th># of unique queries</th>
<th># of unique URLs</th>
<th>Total impression count</th>
<th>Total clicked count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Query session data.</td>
<td>22,212,088</td>
<td>20,686,083</td>
<td>11,216,354</td>
<td>11,317,011</td>
<td>3,572,451,604</td>
<td>18,935,221</td>
</tr>
<tr>
<td>(b) Clickthrough data.</td>
<td>774,096</td>
<td></td>
<td>7,161,456</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Query suggestion log.</td>
<td></td>
<td></td>
<td>110,805,265</td>
<td>876,537,724</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

which represents the probability of a transition from $q$ to $q'$ via a URL in a bipartite graph, where queries and URLs are nodes, and each edge is weighted by $w(q, u)$.

4 Experiments

4.1 Data

Query session and clickthrough data were collected from October 1st to October 10th, 2009 through Microsoft Internet Explorer. Query session data are triplets comprising a session id, a query and a timestamp, while clickthrough data are query-URL pairs. The statistics of the two data are shown in Table 1(a) and (b), respectively.

To evaluate the quality of query suggestions generated from the two data resources, we utilized a query suggestion log between May 2-8, 2010 from Microsoft’s Bing search engine. The query suggestion log contains records whose fields are query, query suggestion, impression count (how many times the suggestion was shown), and clicked count (how many times the suggestion was clicked.) The statistics are shown in Table 1(c).

4.2 Evaluation Method

Query suggestion clickthrough rate (QSCTR) is a metric used to evaluate the quality of query suggestions. We define QSCTR of a $<\text{query}, \text{suggestion}>$ pair as its clicked count divided by its impression count. QSCTR can be interpreted as the probability that a user clicks on a query suggestion given in response to a query. The average QSCTR over our query suggestion log is 0.00632.

A natural evaluation method for the four query suggestion ranking methods would be to let the session-based methods rank all queries from the query session data and let the click-based methods rank all queries from the clickthrough data (Table 1(a) and (b)). However, as we need the click and impression counts for each query suggestion for evaluation, we only rank the $<\text{query}, \text{suggestion}>$ pairs contained in the query suggestion log (Table 1(c)).

4.3 Results and Discussions

Average QSCTRs for top $k$ $<\text{query}, \text{suggestion}>$ pairs are shown in Figure 1. We can see that session-based methods $P_{\text{session}}$ and $P_{\text{sym session}}$ outperform click-based methods.

---

3 http://www.bing.com/
Fig. 1. Average QSCTRs for top $k$ query-suggestion pairs.

Sim$_{\text{click}}$ and $P_{\text{click}}$. Moreover, <query, suggestion> pairs ranked high by the session-based methods achieved high QSCTR, while there is little correlation between ranks by the click-based methods and QSCTR. The symmetric session-based method appears to be the overall winner. This result suggests that considering not only forward reformulations but also backward reformulations is effective for improving QSCTR. Note also that this strategy can suggest more queries than the forward-only method.

To drill down into the overall result, we utilized a query reformulation classification schema proposed by Boldi et al. [4]. Let $X$, $Y$, and $Z$ denote nonempty sets of query terms. Thus, we define specialization as a transition from a query represented by $X$ to that represented by $X \cup Y$; generalization as a transition from $X \cup Y$ to $X$ (or $Y$); parallel movement as a transition from $X \cup Y$ to $X \cup Z$; and error correction as a reformulation where the Levenshtein distance between a query and reformulated query is less than $\theta$ (2 in this study). All other reformulations are classified as new.

Figure 2 shows the QSCTR results per query reformulation type (generalization is omitted due to small sample size.) Figure 2(d) shows that session-based methods are not effective for the new type: it underperforms click-based methods for large $k$. In addition, Figure 2(b) shows that QSCTRs of session-based methods are relatively low for the parallel movement type. For the other two query reformulation types, the general picture is similar to the overall results shown in Figure 1.

5 Conclusion

Using a large-scale query suggestion usage log, we compared some query suggestion selection methods based on query session and clickthrough data. Our experimental results strongly suggest that query session data is superior to clickthrough data as a query suggestion resource.

Acknowledgments

This work was supported in part by the following projects and institutions: a Kyoto University GCOE Program entitled “Informatics Education and Research for Knowledge-Circulating Society,” Grants-in-Aid for Scientific Research (No. 18049041) and JSPS Fellows (No. 10J04687) from MEXT of Japan.
Fig. 2. Average QSCTRs for top $k$ query-suggestion pairs per query reformulation type.

References