2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Predicting Next Search Actions with Search Engine Query Logs

Kevin Hsin-Yih Lin Chieh-Jen Wang Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{hylin, cjwang}@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

***Abstract*—Capturing users' future search actions has many potential applications such as query recommendation, web page re-ranking, advertisement arrangement, and so on. This paper predicts users' future queries and URL clicks based on their current access behaviors and global users' query logs. We explore various features from queries and clicked URLs in the users' current search sessions, select similar intents from query logs, and use them for prediction. Because of an intent shift problem in search sessions, this paper discusses which actions have more effects on the prediction, what representations are more suitable to represent users' intents, how the intent similarity is measured, and how the retrieved similar intents affect the prediction. MSN Search Query Log excerpt (RFP 2006 dataset) is taken as an experimental corpus. Three methods and the back-off models are presented.**

Keywords-action prediction; intent mining; query logs anallysis

I. INTRODUCTION

Understanding what users are doing in the current search sessions, and predict what they will do in the future sessions have many potential applications such as query recommendation, web page re-ranking, advertisement arrangement, and so on. In this paper, we predict users' future actions (i.e., queries and clicked URLs) based on their current access behaviors and global users' query logs.

In a search session, a user submits a sequence of queries intertwined with URL clicks. After each query submission or URL click, we predict the queries and URLs that the user will submit or click during the remainder of the session. Correct predictions can facilitate users' search processes, order the resulting web pages, or arrange suitable advertisements.

This problem is defined formally as follows. Let s = (a_1, a_2, a_3, …, a_n) be a search session of a user, where each action a_i ($1 \le i \le \mathrm{n}$) is either a query submitted by the user or a URL clicked by the user. Session s lasts from the user's web browser's initial connection to the search engine to the time of a timeout between the web browser and the search engine. The actions a_1, a_2, …, a_n are ordered by the time of their occurrences, with a_1 having the earliest occurrence time. Session s can be divided into different pairs (H_1, F_1), (H_2, F_2), …, ($H_{\mathrm{n}-1}$, $F_{\mathrm{n}-1}$) where H_j and F_j are two action sequences such that $H_j = a_1, a_2, …, a_j$ and $F_j = a_{j+1}, a_{j+2}, …, a_n$. We can view H_j as a history of the actions a_1, a_2, …, a_j that a user has performed so far during s, and F_j as the future actions a_{j+1}, a_{j+2}, …, a_n that the user will perform during the remainder of s. The goal is to predict F_j given that H_j is known.

To predict the future actions in F_j based on the current actions in H_j, we postulate the search intents embedded in the actions are coherent and extract the actions of the similar intent in query logs for prediction. The intent shift is one of the major challenging issues in predicting next search actions. The actions in H_j may contain more than one intent. Similarly, a user may change her search intent during F_j. An extreme case is the intent of action a_{j+1} may be different from the intent in H_j. That is, there is an intent shift between a_j and a_{j+1}.

This paper investigates which actions have more effects on the prediction. In other words, we would like to know if using all of the information in the list of queries and URLs that a user has already submitted or clicked in a session is more accurate in predicting a user's future actions than using only the user's most recent submitted queries and clicked URLs. Some other research issues include what representations are more suitable to represent users' intents, how the intent similarity is measured, and how the retrieved similar intents affect the prediction.

The rest of this paper is organized as follows. The related work is presented and compared in Section II. The experimental corpus used in this study is described in Section III. Three prediction methods and their combinations are proposed in Section IV. Experimental results are shown and discussed in Section V. Lastly, Section VI concludes the remarks.

II. RELATED WORK

Several research topics are closely related to our research such as query suggestion, URL recommendation, and context-aware ranking.

The goal of query suggestion is to recommend a set of queries which are related to a user's current search intent. Fonseca *et al.* [1] compute the similarity between a user's current query and a set of candidate queries, and propose the candidate queries which have the highest similarity scores. In 2006, Zhang and Nasraoui [2] convert each session in a query log dataset into a graph of query nodes, and compute the relatedness of the queries based on the queries' path distance from each other. Similarly, Boldi *et al.* [3][4] create a large directed graph out of the queries in a query log dataset. Query suggestions are made by performing PageRank on the graph and recommending the queries with the highest PageRank values. In 2010, Cheng *et al.* [5]

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.15

suggest queries related to the webpage that a user is currently browsing.

The goal of URL recommendation is to suggest a set of URLs which are related to a user's current search intent. Wang *et al.* [6] present a method to predict the future clicked URLs of a user after the user has input a query. Their prediction method aims at generating the URLs that a user will click during the remainder of a user search session.

In the study of context-aware ranking, the documents in the search results are ranked by taking a user's past search actions into consideration. In 2005, Shen *et al.* [7] perform context-aware document ranking by promoting URLs that are more similar to a user's past queries and clicked URLs. Agichtein *et al.* [8] combine users' past click actions with traditional information retrieval model BM25 to rank the results returned by a search engine. A recent study by Xiang *et al.* [9] uses learning-to-rank algorithms to re-rank search engine results.

Another closely-related research topic is personalized search, which can be seen as context-aware ranking tailored to a specific user. This research topic requires query logs which are annotated with user identification information. Dou *et al.* [10] report that doing personalized search by proposing the documents that a user clicks the most often in the past with respect to a query performs very well. To deal with the problem of the sparseness of individual user data, Qiu and Cho [11] construct a topic-level abstraction of query logs. They then exploit a user's preferred topics in personalized search. Teevan *et al.* [12] go beyond the boundary of search engine query logs by incorporating a user's offline desktop search information in their construction of a user preference profile.

Cao *et al.* [13] study context-aware ranking and employ variable length Hidden Markov Model to suggest queries and URLs. Although Cao *et al.* perform query suggestion and URL recommendation, our research goal differs from theirs in that they focus on predicting the next immediate queries and URLs, and they treat query suggestion and URL recommendation as separate tasks. In contrast, our goal is to generate a unified sequence of queries and clicked URLs representing the complete action sequence of a user.

III. A Query Log Dataset

The corpus we use is the MSN Live Search Query Log excerpt (RFP 2006 dataset) [14], which consists of 7,470,915 search sessions dating from May 1st, 2006 to May 31st, 2006. Each session is a sequence of submitted queries and clicked URLs by a user. The exact query strings and clicked URLs in a session are visible. All query submissions and URL clicks are time-stamped. The sessions are anonymous. We normalize query strings to lower cases, and merge consecutive spaces into a single space.

First, we separate the MSN Live Search Query Log excerpt into training and testing datasets. Sessions belonging to the time period from May 1st, 2006 to May 24th, 2006 form the training set denoted by T_r. In total, 5,961,827 sessions are included. The set of unique actions in T_r is denoted by A_{Tr}.

Next, we divide the sessions within May 25th, 2006 and May 31st, 2006 into six groups according to the number of queries in the sessions. The sessions in the six groups contain one, two, three, four, five, and at least six queries, respectively. Table I shows the distribution of the sessions by the number of queries. Intuitively, the distribution is highly skewed to the smaller query counts. If we sample uniformly over the original set, we would obtain very few long sessions with respect to the number of queries, making the performance analysis of long sessions unrepresentative.

Finally, we randomly select a sizeable number of sessions from each group. For the group with a single query, we remove the sessions containing no clicked URLs before sampling, because these sessions are not long enough to make at least one prediction possible. Our final testing dataset contains 1,200 sessions from the period of May 25th to May 31st. This testing dataset is denoted by T_e.

A testing session s consisting of n actions will have n-1 pairs of historical portion H and future portion F, i.e., (H_1, F_1), (H_2, F_2), …, (H_{n-1}, F_{n-1}). There is a prediction for each (H_j, F_j). In T_e, total number of predictions is 7,192.

TABLE I. Distribution of sessions w.r.t query counts.

Query Count	1	2	3	4	5	≥ 6
Percentage	60.4	18.5	8.56	4.54	2.63	5.37

IV. Prediction Methods

In this section, we present three methods to predict future actions. How to use collective intelligence in search query logs is specified.

A. WTAL

The basic idea of the first method is: the actions in a search session are coherent, and the actions in the query logs co-occurring frequently with the actions in the historical portion of a given search session may be probable to appear in its future portion.

The first method called weighted tally (abbreviated WTAL) computes the relevance between a candidate action $a' \in A_{Tr}$ and the recent search action a in the historical portion H of a search session s. The relevance score *WTALSC* of a' and a is measured by their co-occurrence in search query logs T_r as follows.

$$WTALSC(a', a, H) = \sum_{X \in T_r} \sum_{a_i, a_j \in X : a_i = a, a_j = a', j > i} \frac{1}{j-i} \qquad (1)$$

where X is a session in training set T_r, a is the i-th action of X (i.e., a_i=a), and a' is the j-th action of X (i.e., a_j=a'). The value 1/(j–i) (where j>i) denoting a relevance degree of a' and a is higher when the candidate action a' is closer to a in X, i.e., (j–i) is smaller. *WTALSC* sums the relevance scores of a and a' in all the sessions containing a and a'. The *WTALSC* scores of all $a' \in A_{Tr}$ are computed, and the candidate actions are sorted in the descending order of their *WTALSC* scores.

B. *SRPF*

Different from the surface matching of actions in WTAL, the second method considers more action semantics from different sources. The second method called *S*ession *R*etrieval with *P*rediction by *F*requency (abbreviated SRPF) is shown in Figure 1. It includes three major steps: *Indexing*, *Searching*, and *Predicting*. The training sessions in terms of various action semantics are indexed by Indri toolkit[1]. The context in the historical portion *H* is employed to retrieve the relevant sessions in the query logs. They are used to predict the future actions in the portion *F*.

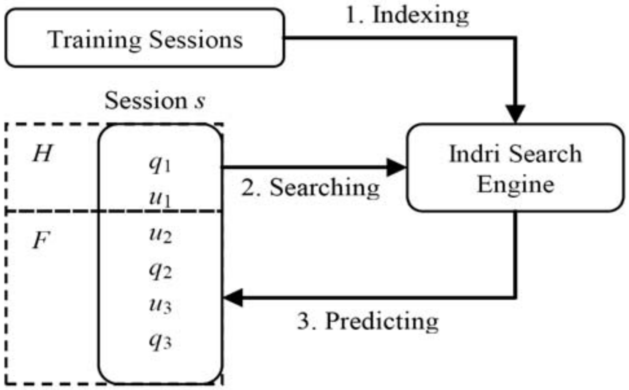

Figure 1. SRPF system architecture.

1) *SRPF Indexing Step*: In this subsection, we describe the indexing step in Figure 1. Various types of information are used to describe an action, and thus a session. The first step of the SRPF method is to convert each session in the training corpus into a pseudo text document for indexing by Indri toolkit. We extract the text strings of the nine fields listed in Table II from each training session.

TABLE II. SESSION INFORMATION.

Field	*Group*	*Information*
f_1	G_1	Query terms
f_2	G_2	Clicked URLs
f_3	G_2	Clicked URLs' domain names
f_4	G_3	Clicked URLs' ODP category paths
f_5	G_3	Clicked URLs' ODP category unigrams
f_6	G_3	Clicked URLs' ODP category path descriptions
f_7	G_4	Clicked URLs' webpage titles listed in the ODP database
f_8	G_4	Clicked URLs' webpage content descriptions listed in the ODP database
f_9	N/A	Clicked URLs' webpage contents

In Table II ODP refers to the Open Directory Project[2], which is an online database of URLs manually annotated with information. Each URL in the ODP database is assigned at least one category path from the ODP's category hierarchy. For example, one of the ODP category paths of Microsoft Corporation (http://www.microsoft.com/) is Computers/Companies/Microsoft_Corporation. ODP category unigrams refer to the individual levels in the ODP category path. For example, the unigrams of Computers/ Companies/Microsoft_Corporation are Computers, Companies, and Microsoft_Corporation. The ODP database further contains a textual description for each category path. In addition, the ODP database provides a webpage's title and content description.

For the clicked URLs' webpage contents, we download the contents of the clicked URLs in the training corpus. Since sessions in the query logs are from the May of 2006, some webpages no longer exist. In T_r, there are 4,033,272 unique URLs. We are able to retrieve webpages for 1,945,490 URLs, 48.24% of a complete set.

After all of the sessions in T_r are converted into pseudo text documents, we index these documents using the Indri toolkit to obtain a database of training sessions ready for retrieval.

2) *SRPF Searching Step*: In this subsection, we describe the searching step in Figure 1. The goal of this step is to retrieve the training sessions related to current user's search context *H*. The intent of the historical portion is represented in the similar way as specified in previous section. SRPF extracts from *H* the text strings in the fields f_1 to f_8 listed in Table II and regards them a pseudo text query for retrieving the relevant sessions in *Tr*.

To test the influence of these eight fields on the final prediction performance, we explore different field combinations when formulating an Indri query in our experiments. We group the fields to reduce the number of combinations. The fields' groups are shown in Table II. Because the ODP coverage of the URLs in the entire corpus is low (5.45%), to prevent having many empty Indri queries, we do not use G_3 and G_4 alone. In the end, twelve combinations of the four groups are tested and Table III lists details.

TABLE III. COMBINATIONS OF FIELDS TESTED

Field Group Combination	*Member Field Groups*
Cobination 1	G1
Cobination 2	G2
Cobination 3	G1, G2
Cobination 4	G1, G3
Cobination 5	G2, G3
Cobination 6	G1, G2, G3
Cobination 7	G1, G4
Cobination 8	G2, G4
Cobination 9	G1, G2, G4
Cobination 10	G1, G3, G4
Cobination 11	G2, G3, G4
Cobination 12	G1, G2, G3, G4

Indri provides various retrieval models listed in Table IV IR_1 and IR_2 are BM25 and TF-IDF, respectively. IR_3 is based on a combination of language model and inference network. Indri also supports the indexing of structured documents. That is, when converting a training session into a text document, we can track which of the eight fields f_1 to

[1] http://www.lemurproject.org/indri.php/

[2] http://www.dmoz.org/

f_8 a text token is extracted from. This information is used with the structured queries of IR_4 to allow only the matches of text tokens in an Indri query and a training session document text of the same field. In the experiments, we set the maximum number of retrieval sessions to 1,000.

TABLE IV. IR MODELS FOR RETRIEVING TRAINING SESSIONS.

IR Model	***Explanation***
IR_1	BM25
IR_2	TF-IDF
IR_3	Indri retrieval model
IR_4	Indri retrieval model with structured query

3) *SRPF Predicting Step*: This subsection describes the predicting step Figure 1. After retrieving training sessions related to H, the SRPF method tallies the frequencies of the actions in the retrieved training sessions. Actions with top frequencies are proposed as a user's future queries or clicked URLs.

When tallying the frequencies of actions in the retrieved training sessions, we have two different weighting schemes. The first weighting scheme is to have an equal weight for every occurrence of an action. The ranking score of an action a in the retrieval result of H is computed as:

$$FREQ(a, H) = \sum_{X \in RETR(H)} \sum_{a_i \in X : a_i = a} 1 \qquad (2)$$

where $RETR(H)$ is the set of training sessions in the retrieval result of H, X is a session (i.e., an action sequence), and a is the i-th action in X. The list of actions in the retrieval result of H is sorted in the descending order of their *FREQ* scores and considered as candidates.

Different from the binary weighting in the first scheme, the weight of an action in the second scheme is the relevance score of the retrieved training session containing the action. The ranking score of an action a with respect to H is computed as:

$$WFREQ(a, H) = \sum_{X \in RETR(H)} \sum_{a_i \in X : a_i = a} REL(X, H) \qquad (3)$$

where $REL(X, H)$ is the relevance score of session X with respect to the retrieval result of H.

When tallying the frequencies of queries and clicked URLs, we also incorporate a technique for eliminating the portion of a retrieved training session which is similar to H from tallying because we postulate that users will not do similar actions in F. To be exact, for a retrieved training session $s = (a_1, a_2, a_3, \ldots, a_n)$, let j be the largest number such that $a_j \in s$ and $a_j \in H$. Then we keep only the subsequence $s' = (a_{j+1}, a_{j+2}, a_{j+3}, \ldots, a_n)$. In the experiments, we use the notation *ELIM* to indicate that the elimination is used. Conversely, *NoELIM* means that elimination is not used.

In summary, the combination of information fields, IR models, ranking score functions, and elimination functions give us 192 variants of SRPF.

C. ACTF

In this subsection, we present the action flow graph method, abbreviated ACTF. This method is an extension of Boldi *et al.*'s [3] query flow graph method for query suggestion.

In their work, Boldi *et al.* use a query log corpus to construct a directed graph of queries, called a query flow graph. In this graph, each node is a query that appears in the query logs. There is a directed edge from a query q_i to another query q_j if and only if q_j appears immediately after q_i time-wise in a search session. Boldi *et al.* use a modified version of the PageRank algorithm on their query flow graph to increase the weights of the query nodes near the nodes representing the queries that a user has already submitted. The queries with top PageRank scores are proposed to users as suggested queries.

We extend Boldi *et al.*'s query flow graph algorithm to ACTF as follows. Instead of creating a graph out of queries, we construct a graph out of both queries and clicked URLs. This graph is called an *action flow graph*. In the graph, each node is either a query or a clicked URL that appears in T_r. There is an edge from node a_i to node a_j if a_j immediately follows a_i in a search session. The weight of an edge from a_i to a_j is $f(a_i, a_j)/f(a_i)$, where $f(a_i)$ is total occurrences of a_i in T_r, and $f(a_i, a_j)$ is the number of occurrences of a_j immediately follows a_i in a session in T_r.

The complete action flow graph constructed from the training corpus contains 9,417,766 nodes and 16,992,035 edges. To reduce the computation time of the PageRank algorithm, we prune the action flow graph by keeping only the nodes $a \in H$ and the descendant nodes a' of a such that the edge connecting a' to its parent has a weight of at least 0.05. We generate a list of predicted future actions from the pruned graph and sort them in the descending order of their PageRank values.

V. EXPERIMENTS AND DISCUSSIONS

In this section, we present the experimental setup, evaluation metrics, performance, and discussions.

A. Experimental Setup

The goal of the experiments is to determine how well our methods perform in predicting the future actions of a user. More formally, each method has to perform the task of predicting F given H is known. In the experiments, each prediction method predicts an action sequence for each of the 7,192 prediction tasks. The quality of a prediction sequence is determined based on how closely the prediction sequence resembles a user's real future action sequence F.

We use four evaluation metrics including R-Precision, LCSF, ExactMatch and First1. Let L be an action sequence generated by a prediction method. R-Precision measures the fraction of the first $|F|$ actions in L that is correct. LCSF determines the longest common subsequence between F and the first $|F|$ actions in L, and measures the performance by the ratio of the subsequence to F. ExactMatch measures the length of the longest identical consecutive portions of F and L starting from the first action in F and L, and expresses the length as a fraction of |*F*|. First1 measures whether the first action of *F* is the same as the first action of *L*. If yes, then the First1 score is 1, othereise it is 0.

We use two approaches to average the performance scores over the 7,192 prediction tasks. The first approach is to divide the total performance score by 7,192. This averaging approach is denoted by AVG in the experimental results. However, this average is not representative of the real-world performance, because the testing sessions are first grouped according to query count before sampling (see Section III). In the second approach, the prediction tasks are grouped by query counts in testing sessions. The average performance within each group is then computed. Then we compute the weighted average of the groups' scores using the weights listed in Table I. This second average score, denoted by WAVG, is representative of the real-world performance.

As a comparison to our methods, we use Wang *et al.*'s Simu0Default method in [6] on our corpus. Although Wang *et al.* also present other methods in [6], we do not use them because their other methods rely on knowing the correct answer before making a prediction, which is not possible in the online, real-world setting that we simulate in our experiments.

B. SRPF Critical Features

By varying the features of SRPF, we obtain 192 variants (See Section IV.B). In this subsection, we examine SRPF's most critical features. Figure 2 shows the best SRPF variants for each of the eight measures. Variant 1 uses f_1, IR_1, *WFREQ* and *ELIM*. Variant 2 uses f_1, f_2, f_3, IR_1, *WFREQ* and *NoELIM*. Variant 3 has the same features as Variant 1. Variants 4, 6 and 8 have the same features as Variant 2. Variant 5 uses f_1, f_2, f_3, IR_1, *WFREQ* and *ELIM*. Variant 7 has the same features as Variant 5.

To study which features are more critical in the prediction, we perform paired *t*-tests on each of the eight variants in Figure 2 against all other 191 variants. This analysis procedure is best explained through an example. Take Variant 1 as an example. Paired *t*-tests are performed on the performance measure that Variant 1 excels in WAVG R-Precision. After performing the paired *t*-tests on Variant 1 against all other 191 variants, we consider the group of variants whose WAVG R-Precision scores are not statistically significantly different (p-value ≥ 0.01) from Variant 1 as the leading performance group for the WAVG R-Precision measure. Common features in this group are identified and regarded as the most important features contributing to the WAVG R-Precision score. The same procedure is repeated for the other seven variants on their corresponding top-performing measures. Note that for the statistical significance tests on WAVG scores, we use the approximation weighted paired *t*-test method described in Donner and Donald's work [15].

Results of the aforementioned paired *t*-test procedure are shown in Table V. We see that f_1 (i.e., the use of query terms when converting *H* into an Indri query) is present for all eight performance measures.

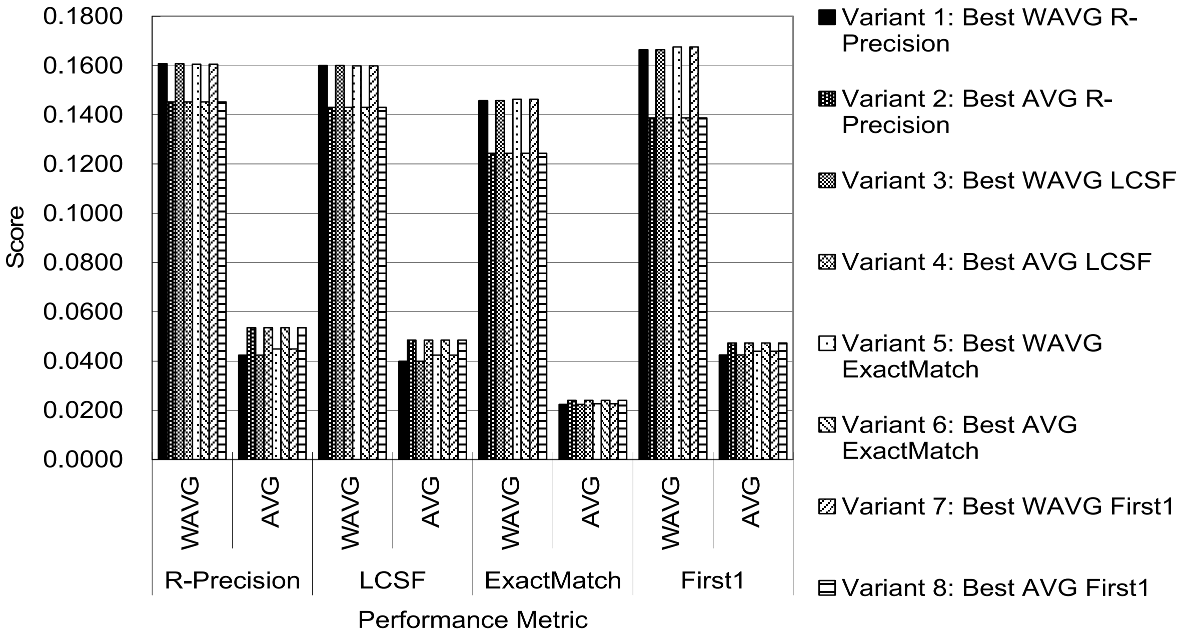

Figure 2. The best eight SRPF variants.

TABLE V. SRPF CRITICAL FEATURES.

Performance Measure	*Leading Performance Group Size*	*Common Features*
WAVG R-Precision	28	f_1
AVG R-Precision	7	f_1,f_2,f_3, *WFREQ*, *NoELIM*
WAVG LCSF	25	f_1, *ELIM*
AVG LCSF	6	f_1,f_2,f_3, *WFREQ*, *NoELIM*
WAVG ExactMatch	25	f_1, *ELIM*
AVG ExactMatch	24	f_1
WAVG First1	24	f_1, *ELIM*
AVG First1	15	f_1, *WFREQ*

C. Performance Comparison across Methods

In this subsection, we compare the performance of the proposed methods to identify the best one. Figure 3 shows the performance of our three major methods and Wang *et al.*'s method. In Figure 3, SRPF Variant 1 from Figure 2 is used to represent SRPF. Although the other seven SRPF variants are not shown in Figure 3, their performance scores are also ranked between ACTF and Wang *et al.*'s methods as SRPF Variant 1 in Figure 3.

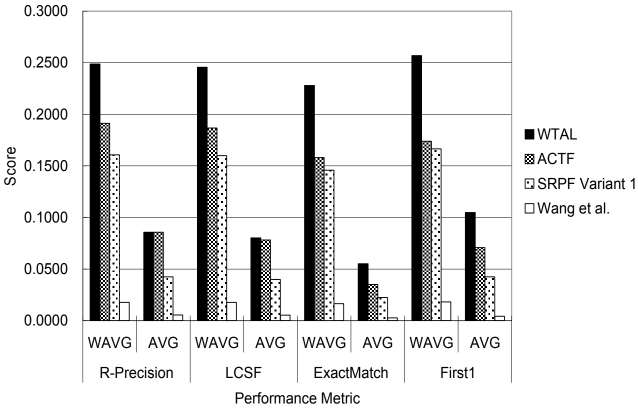

Figure 3. Performance of different methods.

In Figure 3, WTAL has the best performance for all measures except AVG R-Precision. Statistical analysis indicates that, except for AVG LCSF, WTAL's performance in its dominating measures is statistically significantly higher than that of other methods with p-value < 0.01. For AVG LCSF, the performance difference between WTAL and ACTF is not statistically significant, but the difference is statistically significant between WTAL and all other methods. As for AVG R-Precision, ACTF has the best performance. However, the difference between ACTF's AVG R-Precision and WTAL's is not statistically significant. ACTF's AVG R-Precision is statistically significantly higher (p-value < 0.01) than the other two methods'. In summary, WTAL performs statistically significantly better than all other methods in six of the eight measures. Hence, we consider WTAL to have the best overall performance.

The performance of Wang *et al.*'s method is much lower than the other methods. There are several possible reasons for the low performance. First, Wang *et al.*'s method is designed to predict only clicked URLs, but in our evaluation, we require a method to predict both queries and clicked URLs. Hence, the performance of Wang *et al.*'s method may be hindered by its lack of query predictions. To explore this factor, we perform a separate evaluation of Wang *et al.*'s method where we remove every query from *F*. Results show that the change in performance is very small (i.e., maximum absolute performance difference is 0.0004 for all eight performance measures). Thus, the inability to predict queries is not a major factor contributing to low performance. Another possible reason is that Wang *et al.*'s method uses only the queries which appear at least five times in the training corpus, so a substantial amount of useful information in the training corpus may be filtered out. A third possible reason may be the simulation of snippet generation in our implementation of Wang *et al.*'s method. In Wang *et al.*'s work, a pseudo text document for each query consists of the query's associated clicked URLs' snippets generated by a commercial search engine. In our work, we simulate snippet generation by extracting the words that are within ten words from a query term from the clicked URLs' HTML files we download. Using this snippet generation method, we are only able to generate snippets for 23.79% of the clicked URLs. Wang *et al.*'s original snippet generation method may have a better coverage.

D. Effectiveness of Information in H

We observe WTAL has the best overall performance is due to the fact that WTAL uses only very recent information in *H*. That is, WTAL uses only the information associated with a user's most recent query in *H* to make a prediction of future actions. This realization leads us to speculate the importance of the older information in *H* in the prediction of future actions. To investigate further, we change SRPF Variant 1 to use only the information in *H* up to the most recent query, the most recent 2 queries, the most recent 3 queries, and the most recent 4 queries only. This modification effectively gives us four sub-variants of SRPF Variant 1. We examine the performance of these sub-variants to determine the effectiveness of the historical information in *H* with respect to how many recent queries are considered in prediction.

Figure 4 shows the performance of SRPF Variant 1 using different amounts of information in *H*. In the figure's legend, the notation *Size* = *n* means that this particular variant uses only the information from the current action up to the *n* most recent queries in *H*. The notation *Size* = *All* means that all information in *H* is used (i.e., the original SRPF Variant 1). We call these as the *history window sizes*.

In Figure 4, history window size 1 has the best performance in all eight measures. This observation entails that for SRPF Variant 1, using more and older information in *H* does not improve performance. Intent shift may be one of the possible reasons.

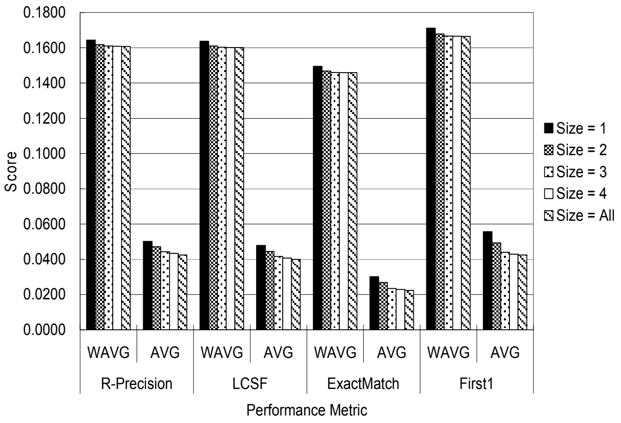

Figure 4. Effects of SRPF history window size.

E. *Performance by Session Query Count*

In Section III, we separate testing sessions prior to sampling into six groups according to session query count. In this subsection, we study our methods' performance for each of these six groups.

Figure 5 shows the performance of our methods with respect to session query count. For each of the six query count groups, its performance is obtained by taking the non-weighted average of the performance of the prediction tasks within the group.

In Figure 5, a trend is that, in almost all cases, the performance of a method decreases as the number of queries increases in a session. Since the WAVG scores give higher weights to the better-performing, smaller query count groups, the WAVG scores are higher than the AVG performance scores. This explains why in Figure 2 to Figure 4 the WAVG scores are always higher than their AVG counterparts.

As to why performance drops as the number of queries per session increases, we conjecture as the number of queries per session increases, the variety of information contained in a session also increases. And a testing session with a wider information variety will have more queries and clicked URLs that are not in the training set, which results in more cases that are impossible to predict using our methods. To examine whether the query log corpus supports this conjecture, we compute the average coverage rate of testing session queries and clicked URLs in T_r with respect to the number of queries in a testing session. We find that the single query testing sessions have the highest coverage of 60.01%. The coverage decreases as query count increases, and drops to 40.30% for the group with at least six queries. This observation supports our conjecture that testing sessions with more queries contain more information that are not present in the training set, and hence make more future queries and clicked URLs impossible to predict.

F. *Back-off Methods*

Figure 3 shows that WTAL has the best performance in general, ACTF comes in second, and SRPF comes in third. This order is the opposite order of the methods' coverage capabilities, which is the ability to generate a non-empty prediction. WTAL has the lowest coverage capability of 34.18%, because it requires the most recent query in H to appear in T_r. ACTF has the second lowest coverage capability of 69.35%, because it requires at least one query or clicked URL in H to appear in T_r. SRPF has the highest coverage capability of 99.49%, because its IR models for training session retrieval allow non-exact matches between the actions in H and the training sessions.

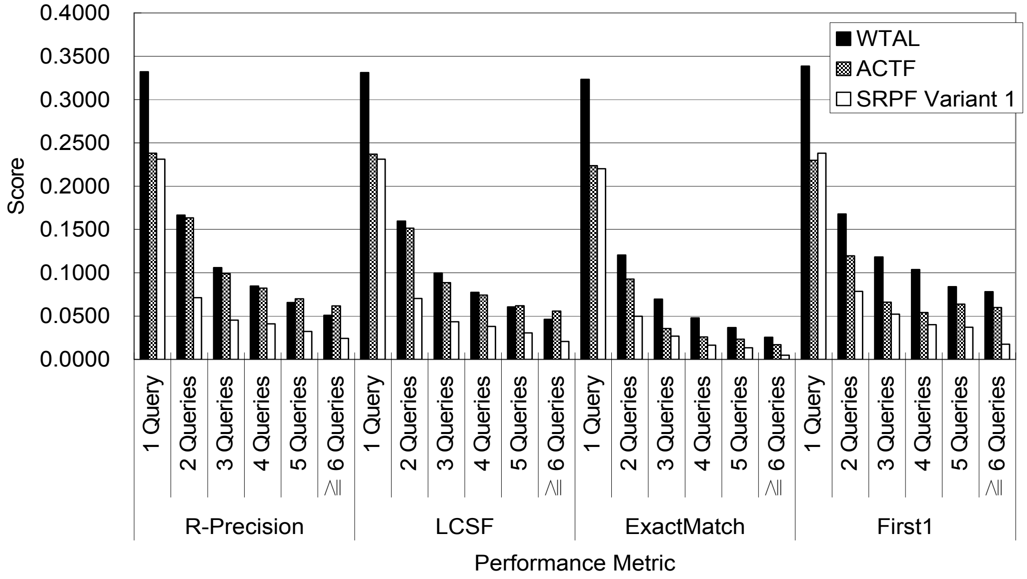

Figure 5. Performance by session query count.

The opposite order relationship of performance and method coverage capabilities gives us an insight on improving performance. In a nutshell, we use high coverage capability, low performance methods to back off the low coverage capability, high performance methods when high performance methods cannot yield a prediction. In this way, we propose three combinations: WTAL-SRPF, ACTF-SRPF, and WTAL-ACTF-SRPF consulted in the order of coverage. For WTAL-SRPF, if WTAL is unable to generate a prediction, then SRPF is used. The same idea applies to ACTF-SRPF. For WTAL-ACTF-SRPF, if WTAL is unable to generate a prediction, then ACTF is used. If ACTF is also unable to generate a prediction, then SRPF is used. We use SRPF Variant 1 in Figure 3 as the SRPF method in all combined methods involving SRPF.

Figure 6 shows the performance of the combined methods. WTAL is shown for comparison. In the figure, WTAL-ACTF-SRPF has the best performance in all measures, and it performs better statistically significantly than WTAL with p-value < 0.01 in every measure. Thus, combining the methods together indeed enhances the overall performance.

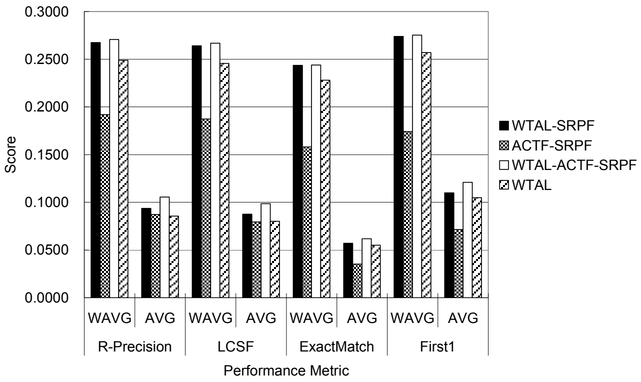

Figure 6. Performance of Back-off methods.

VI. Conclusion and Future Work

In this work we predict user future action sequence based on their current search actions and global search engine query logs. We propose three different methods including WTAL, SRPF and ACTF to deal with this problem. Experimental results show that WTAL has the best performance, but has a low coverage problem. Merging the individual methods together achieves the best performance. Experimental results also show that using more historical search information in a user's search session is not guaranteed to be helpful to improve prediction performance.

It is more challenging to predict future actions of the sessions containing many queries. In the future, we plan to further improve the correctness of the prediction sequence order of the proposed approaches. In addition, we will use the predicted queries and clicked URLs as hints for advertisement recommendation.

VII. Acknowledgments

Research of this paper was supported in part by National Science Council, Taiwan, under the contract NSC98-2221-E-002-175-MY3. We are also grateful to Microsoft Research Asia for the support of MSN Search Query Log excerpt.

VIII. References

[1] B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani, "Using Association Rules to Discover Search Engines Related Queries," in *Proceedings of the First Conference on Latin American Web Congress*, pp. 66–71, 2003.

[2] Z. Zhang and O. Nasraoui, "Mining search engine query logs for query recommendation," in *Proceedings of the 15th international conference on World Wide Web*, pp. 1039–1040, 2006.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The query-flow graph: model and applications," in *Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 609–618, 2008.

[4] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna, "Query suggestions using query-flow graphs," in *Proceedings of the 2009 workshop on Web Search Click Data*, pp. 56–63, 2009.

[5] Z. Cheng, B. Gao, and T. Liu, "Actively predicting diverse search intent from user browsing behaviors," in *Proceedings of the 19th international conference on World wide web*, pp. 221–230, 2010.

[6] X. Wang, B. Tan, A. Shakery, and C. Zhai, "Beyond hyperlinks: organizing information footprints in search logs to support effective browsing," in *Proceeding of the 18th ACM conference on Information and knowledge management*, pp. 1237–1246, 2009.

[7] X. Shen, B. Tan, and C. Zhai, "Context-sensitive information retrieval using implicit feedback," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 43–50, 2005.

[8] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 19–26, 2006.

[9] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li, "Context-aware ranking in web search," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 451–458, 2010.

[10] Z. Dou, R. Song, and J. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proceedings of the 16th international conference on World Wide Web*, pp. 581–590, 2007.

[11] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," in *Proceedings of the 15th international conference on World Wide Web*, pp. 727–736, 2006.

[12] J. Teevan, S. T. Dumais, and E. Horvitz, "Personalizing search via automated analysis of interests and activities," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 449–456, 2005.

[13] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li, "Towards context-aware search by learning a very large variable length hidden markov model from search logs," in *Proceedings of the 18th international conference on World wide web*, pp. 191–200, 2009.

[14] N. Craswell, R. Jones, G. Dupret, and E. Viegas, "Proceedings of the 2009 workshop on Web Search Click Data," Barcelona, Spain, p. 95, 2009.

[15] A. Donner and A. Donald, "Analysis of data arising from a stratified design with the cluster as unit of randomization," *Statistics in Medicine*, vol. 6, no. 1, pp. 43-52, 1987.