Context-Aware Search Personalization with Concept Preference

Di Jiang, Kenneth Wai-Ting Leung, Wilfred Ng
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{dijiang, kwtleung, wilfred}@cse.ust.hk

ABSTRACT

As the size of the web is growing rapidly, a well-recognized challenge for developing web search engines is to optimize the search result towards each user's preference. In this paper, we propose and develop a new personalization framework that captures the user's preference in the form of concepts obtained by mining web search contexts. The search context consists of both the user's clickthroughs and query reformulations that satisfy some specific information need, which is able to provide more information than each individual query in a search session. We also propose a method that discovers search contexts by one-pass of raw search query log. Using the information of the search context, we develop eight strategies that derive conceptual preference judgment. A learning-to-rank approach is employed to combine the derived preference judgments and then a *Context-Aware User Profile* (CAUP) is created. We further employ CAUP to adapt a personalized ranking function. Experimental results demonstrate that our approach captures accurate and comprehensive user's preference and, in terms of Top-N results quality, outperforms those existing concept-based personalization approaches without using search contexts.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Search Personalization, Clickthrough, Query Reformulation

1. INTRODUCTION

With the exponential growth of information available on the Web, search engines have become an indispensable tool of people's daily activities [7]. However, as search queries are typically short and ambiguous, search engines have limited clues to infer a user's true search intents and thus can only return roughly the same result for the same query. This *one size fits all* strategy makes search engines perform only suboptimally for many users [5]. To alleviate this problem, search personalization has been studied in order to adapt search results to individual's explicit or implicit feedback. Since users are largely reluctant to provided explicit feedback due to the extra efforts involved, implicit feedback is commonly used as the major resource for search personalization [4, 8].

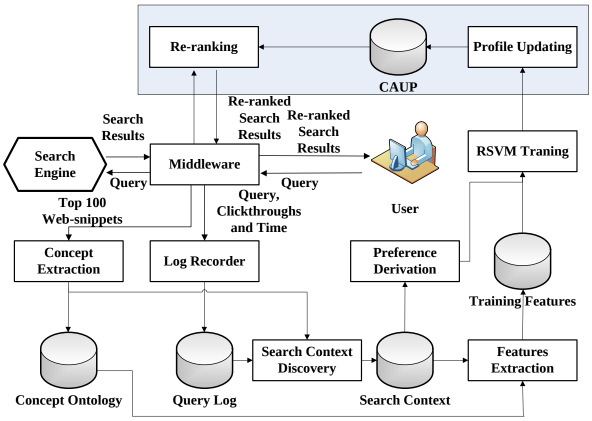


Figure 1: Main Processes in Personalization Framework

The most widely used implicit feedback is the user's clickthrough information [4]. Several personalization approaches have been proposed to personalize search results based on each individual query's clickthrough information [4, 8]. However, as the user usually submits a sequence of queries to satisfy the same information need [11], the search context that is composed of previous queries, previous clickthroughs as well as the semantic relations between queries become a potential resource to infer the user's preference with higher accuracy and broader information coverage.

We propose a framework that utilizes search context to personalize search results. Figure 1 shows the general process of the proposed framework, which consists of two fundamental activities: (1) Profile Updating and (2) Re-ranking.

- **Profile Updating**: When the search results are returned from the backend search engine, the concepts (i.e. important terms and phrases) and their rela-

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.
Copyright 2011 ACM 978-1-4503-0717-8/11/10 ...$10.00.

tions are mined from the search results and stored in *Concept Ontology*. Queries, submitting times and clickthrough behaviors are also recorded in *Query Log*. Then search contexts are extracted from the stored information and eight strategies are employed to derive conceptual preference. The conceptual preference is used in RSVM [4] training to update CAUP.

- **Re-ranking**: When a user submits a query, the search results are obtained from the backend search engines (Google) and then re-ranked according to CAUP.

In order to realize the proposed personalization approach, we need to overcome some challenging issues. First, we need to develop a method that automatically discovers search context from raw search query log. Although many existing methods assume that the context is already known, obtaining search context for personalization usage is non-trivial. As providing irrelevant results for the user is frustrating, a high coherency is required to avoid involving irrelevant information and meanwhile the integrity should also be kept to avoid breaking a context into several separated ones. Thus, we propose a method that utilizes a range of techniques concerning temporal cutoff, query relevance and *Search Engine Result Pages* (SERPs) similarity in order to discover search context by one-pass of the query log.

Second, search context contains different types of information. Based on the diverse nature of the observed evidence in search context, we need to design different strategies to infer the user's preference. Additionally, preference derived from different sources should be seamlessly combined to update CAUP for the user. To meet these requirements, our framework uses three strategies to derive preference judgment from the user's clickthrough behaviors and five strategies to derive preference judgment from the user's query reformulation behaviors. Then a learning-to-rank method is used to combine the derived preference judgment and update CAUP, which is further used to personalize search results.

The main contributions of this paper can be summarized as follows:

- Our search personalization is based on search context related to a specific information need rather than individual queries. We propose a new method that discovers search contexts from query log with high coherency and integrity.

- Eight preference derivation strategies are proposed to infer the user's conceptual preference through mining search context. We study in detail the characteristics and effectiveness of each proposed preference derivation strategy.

- A learning-to-rank approach is proposed to seamlessly combine preference judgment from different preference derivation strategies and utilize them to update CAUP, based on which the search results are re-ranked.

- We implement a working prototype to verify the proposed ideas. It consists of a middleware for capturing user interaction information, performing personalization, and interfacing with backend search engine. Empirical results show that our approach successfully captures the user's real search preference and outperform methods without considering contextual information.

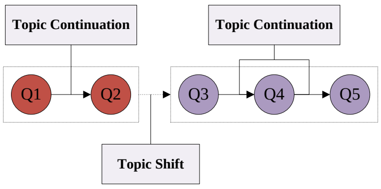


Figure 2: Relations Between Queries

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the method of concept extraction. Section 4 details our search context discovery technique. Section 5 explains the strategies of deriving conceptual preferences from search context. Personalized ranking function is described in Section 6 and experimental results are given in Section 7. Finally, we conclude the paper in Section 8.

2. RELATED WORK

The proposed framework utilizes interaction evidence in search context to support search personalization. In this section, we review two related fundamental topics: *session segmentation* and *search personalization*.

2.1 Session Segmentation

In the field of session segmentation, the relations between queries are categorized as *Topic Continuation* and *Topic Shift*, which are illustrated in Figure 2. Assume that query Q1 and Q2 are semantically related, thus they are grouped in the same session and the relation between them is *Topic Continuation*. Similarly, Q3, Q4 and Q5 are also from the same session and the relations between them are *Topic Continuation*. On the contrary, Q2 and Q3 have no semantic relation and the relation between them is *Topic Shift*, which generates a session boundary.

In previous studies, a fixed temporal cutoff is widely used as the indicator of *Topic Shift*. Radlinski *et al.* [11] used a half-an-hour cutoff on the log of a library search engine and discovered that this helps achieve a good precision in finding search sessions. However, Jones *et al.* [6] reported that only using timeout cutoff is not good enough for logs from general search engines. Gayo-Avello [1] provided a good survey about several session segmentation methods, which primarily based on temporal cutoff and lexical similarity between queries. More recently, in order to have a better understanding of the relation between search queries, Huang *et al.* [3] built a rich taxonomy of query refinement strategies and developed a high precision rule-based classifier to detect related queries. *Topic Continuation* is further characterized as different types of query reformulation.

Our work on search context discovery is different from previous studies in two ways. First, as high accuracy is required for personalization, keeping the coherency of search context is prioritized by our search context discovery method and meanwhile it also keeps the context integrity to avoid breaking a search context into several separated ones. Second, the search context discovered by our approach is not simply a group of semantically related queries. Through one-pass of the query log, our method also captures the query reformulation type between consecutive queries, which contains valuable information for inferring the user's preference.

2.2 Search Personalization

Search personalization aims to return the most relevant search results according to the user's interests and preferences. In order to achieve this goal, clickthrough is widely used in previous works to infer the user's preferences. Based on the research of search engine users' browsing behaviors, Joachims [5] first developed a framework that utilized clickthrough data to infer users' preferences on documents and then learned to adapt the ranking function. Radlinski *et al.* [11] extended Joachims' method to infer the user's document preference based on the clickthrough raised by different queries. As Radlinski's method can deduce new types of preference judgment, the learned ranking function outperformed the methods that only used information of individual query. Shen *et al.* [12] proposed a method which incorporates previous queries and clickthroughs to build language models. Luxenburger *et al.* [10] proposed a personalization framework that match users' information need with their searching history. Xiang *et al.* [15] proposed four strategies to re-rank search results by the relation between queries, and their method showed performance improvement compared with those without context information. Leung *et al.* [8] developed several profiling methods to capture the user's positive and negative preferences by concepts. By analyzing each search query individually, the profiling methods in [8] captures the user's preference with fine granularity and demonstrates better performance than those based on document preference. However, with the importance of context and the advantage of concept-based preference representation, few works has been done to study concept-based search personalization in a context-aware approach.

The differences between our work and the previous counterparts are: First, our framework utilizes search context rather than individual queries to derive preference and thus obtains more comprehensive information coverage, which improves the accuracy of generated preference judgment. Second, the user's preference is represented by concepts in current search results as well as the related concepts from the concept ontology. In this way, our framework captures the user's preference on more concepts and thus improves personalization accuracy. Third, we propose a learning-to-rank method that seamlessly combines different conceptual preference judgment and use the combined preferences to update the *Context-Aware User Profile.*

3. CONCEPT EXTRACTION

Informally, if a term/phrase c appears frequently in the web-snippets [1] arising from a query q, then c represents an important concept related to q, as it co-exists in close proximity with the query in the top documents. Our concept extraction method first extracts all the terms and phrases from the web-snippets arising from q and denote a term or a phrase as c_i. After obtaining a set of term/phrase, we use the following formula to clean the concept set and delete those concepts that are not related to the query or the user's interests.

$$\theta < support(c_i) = \frac{sf(c_i)}{n} \cdot |c_i|$$

where $sf(c_i)$ is the snippet frequency of the term/phrase c_i (i.e. the number of web-snippets containing c_i), n is the number of web-snippet returned and $|c_i|$ is the number of term in the term/phrase c_i. θ is the frequency threshold we use to distinguish concepts from other terms. θ is set to 0.03 in the experiments. We choose a relative small threshold in order to obtain good information coverage.

ID	Reformulation Type	Example
1	Repeat	Apple -> Apple
2	Add Whitespace/Punctuation	Apple Pie -> Apple, Pie
3	Remove Whitespace/Punctuation	Apple Pie -> ApplePie
4	Add URL	apple -> www.apple.com
5	Strip URL	www.apple.com -> apple
6	Word Reorder	Apple Pie -> Pie Apple
7	Expand Acronym	UN -> United Nations
8	Form Acronym	United Nations -> UN
9	Expand Abbreviation	Soft App -> Software Application
10	Form Abbreviation	Software Application -> Soft App
11	Word Substitution	Apple -> Fruit
12	Stemming	Running -> Run
13	Singular/Plural Conversion	Woman -> Women
14	Substring	Music Record -> Music Rec
15	Superstring	Music Rec -> Music Record
16	Add Words	Apple -> Apple Pie
17	Remove Words	Apple Pie -> Apple
18	Spelling Correction	Appple -> Apple
19	Multiple Reformulation	Running hound -> Dog

Figure 3: Query Reformulation Taxonomy

4. SEARCH CONTEXT DISCOVERY

Search context consists of queries sharing the same topic, clickthrough raised by these queries and various reformulation types between queries. In this section, we present our search context discovery method, which captures this information through one-pass of raw search query log.

4.1 Temporal Cutoff

Temporal cutoff is widely used as the indicator of topic shift. Temporal cutoffs ranging from 5 minutes to 30 minutes are frequently used to identify session boundaries. In order to evaluate the effectiveness of these cutoffs, we utilize different temporal cutoffs within 90 minutes to evaluate their performance of search context discovery. We observe that a 30-minute cutoff achieves fairly good performance and the results are presented in Section 7.

4.2 Query Relevance

We utilize query reformulation taxonomy to evaluate the relevance between query strings. The reasons of using query reformulation taxonomy are twofold: First, the taxonomy-based approach demonstrates the state-of-the-art precision of detecting *Topic Continuation*, i.e., the approach has a high precision in detecting query relatedness. Second, reformulation taxonomy captures the reformulation type between consecutive queries and we will show later that various reformulation types contain valuable information that helps infer the user's preference.

The taxonomy is presented in Figure 3. Reformulation types except the last one belong to *Single Reformulation*, which is used to detect a certain type of change between two queries. The *Single Reformulation* we use is an enriched version of that in [3] and readers may refer to it for more detailed information. As Huang *et al.* [3] reported that only using *Single Reformulation* tends to damage the

[1] A "web-snippet" denotes the title, summary and URL of a Web page returned by search engines.

Table 1: Weight of Term Similarity Features

Features	Weight
Same	≃1.99
Singular/Plural Conversion	≃1.99
Stemming	≃1.99
Spelling Correction	<0.01
Superstring	<0.01
Substring	<0.01
AddURL	<0.01
StripURL	<0.01

integrity of search context. To resolve this problem, we use *Multiple Reformulation* to handle the scenario that a user reformulates his/her queries by combining more than one reformulation types. For example, *horses race* → *horse* is a reformulation that combines *SingularAndPlural* and *RemoveWords*. Although these two queries are semantically related, reformulation types in *Single Reformulation* are too strict to capture them. As exhaustively enumerating each combination of reformulation types is infeasible, we propose to evaluate the relevance between two queries by the similarity of each pair of their terms. We first tokenize each query into terms by whitespace and then use six term similarity features to train an SVM model. The weight of each feature is listed in Table 1, from which we observe that the features *Same*, *Singular/Plural Conversion* and *Stemming* play the dominant roles.

4.3 SERPs Similarity

The limitation of taxonomy-based method is that it cannot capture reformulations beyond the predefined taxonomy. As search results returned from search engine usually contain abundant information about the query, *Search Engine Result Pages* (SERPs) are potentially helpful to detect the similarity between queries. For each query q, we extract concepts from the top 100 snippets returned by the search engine and then the query is represented as a concept vector $\vec{C_q}$. For simplicity, we utilize cosine similarity to evaluate the similarity between two SERPs. It is defined as follows:

$$SERPs(q_1, q_2) = cos(\vec{C_{q_1}}, \vec{C_{q_2}}) = \frac{\vec{C_{q_1}} \cdot \vec{C_{q_2}}}{\|\vec{C_{q_1}}\| \|\vec{C_{q_2}}\|}$$

Based on the three metrics proposed in Section 4.1, 4.2 and 4.3, we present our search context discovery method in Algorithm 1. Technically, separating queries from different users is straightforward, since each query is associated with user identifiers in search query log. Thus, we focus on how to discover search context for queries submitted by the same user.

Algorithm 1 Search Context Discovery

Input: Temporal Cutoff T; Query Reformulation Taxonomy Tax; SERPs Similarity Threshold θ_{SERPs};
Output: Search contexts segmented by *Topic Shift*;

```
for all $q_i$ in log do
    if Time($q_{i+1}$)-Time($q_i$)>T then
        Relation($q_i, q_{i+1}$) ←Topic Shift
    else
        if (Reformulation($q_i$,$q_{i+1}$)∈ $Tax$)=FALSE then
            if $SERPs(q_i, q_{i+1}) < \theta_{SERPs}$ then
                Relation($q_i, q_{i+1}$) ←Topic Shift
            else
                Relation($q_i, q_{i+1}$) ←Unknown Reformulation
            end if
        else
            Relation($q_i, q_{i+1}$) ←Reformulation($q_i$,$q_{i+1}$)
        end if
    end if
end for
```

5. CONCEPT PREFERENCE DERIVATION

In this section, we discuss the strategies of deriving the user's conceptual preference from the search context. We propose three strategies to derive preference from clickthrough behavior and five strategies to derive preference from query reformulation behavior.

5.1 Preference Derivation by Clickthrough

Joachims *et al.* [5] made an in-depth study about a user's browsing behaviors in the process of web searching and proposed three assumptions:

- Users scan the results in order from top to bottom.
- Users at least read the top two result snippets and the first one is much more likely to be clicked on.
- Users read one snippet below any they click on.

Based the aforementioned assumptions, we define a user's examination range of the search results as follows:

- If a snippet ranked i is the last snippet that a user clicked on the result, the examination range of the result is $1 \rightarrow (i+1)$.
- If no snippet is clicked on the result, the examination range of the result is $1 \rightarrow 2$.

The following three strategies focus on snippets within the examination range and ignore those beyond it.

1. Click $>_q$ Skip Above: *Assume $(s_1, s_2, ...)$ is the ranking of snippets arising from query q, the corresponding concept set for each snippet is denoted as $(C_1, C_2, ...)$. If s_i is clicked while s_j is not clicked and $i > j$, derive a concept preference judgement $c_a >_q c_b$ for concept c_a from C_i and concept c_b from C_j.*

The intuition behind this strategy is that a user scans the search results from top to bottom. If he/she skipped a snippet s_i before clicking on snippet s_j, he/she must have scanned s_i and decided not to click on it. Based on this observation, we infer that concepts in s_j are more likely to be preferred than concepts in s_j and pairwise preference judgment is generated for each concept pair between s_i and s_j.

2. Click $>_q$ No-Click Next: *Assume $(s_1, s_2, ...)$ is the ranking of snippets arising from query q, the corresponding concept set for each snippet is denoted as $(C_1, C_2, ...)$. If s_i is clicked while s_j is not clicked and $j = i + 1$, derive a concept preference judgment $c_a >_q c_b$ for concept c_a from C_i and concept c_b from C_j.*

The intuition behind this strategy is that a user would not scan much below a clicked snippet s_i, and he/she usually view the immediately following snippet s_j. Therefore concepts in s_i are more likely to be preferred than concepts in s_j and pairwise preference judgment is generated for each concept pair between s_i and s_j. Note that the preference generated by this strategy confirms the original ranking.

3. Click $>_{q'}$ No-Click Earlier: *Assume q' is an earlier query of q within the search context. $(s'_1, s'_2, ...)$ is the ranking of snippets arising from q' and the corresponding concept set for each snippet are denoted as $(C'_1, C'_2, ...)$. Similarly, $(s_1, s_2, ...)$ is the ranking of snippets arising from q and the corresponding concept set for each snippet are denoted as $(C_1, C_2, ...)$. If s_i is clicked while s'_j is skipped by the user, derive a concept preference judgment $c_a >_{q'} c'_b$ for concept c_a from C_i and concept c'_b from C'_j.*

The intuition behind this strategy is that concepts in clicked snippets are more likely to be preferred than concepts in skipped snippets arising from earlier query. This strategy can generate preference judgment over concepts arising from different queries. By applying this strategy, we are able to compare concepts from different queries and thus capture more information to infer the user's preference.

5.2 Preference Derivation by Query Reformulation

Query reformulation reflects the semantic relation between two consecutive queries. We first evaluate the distribution of query reformulation types from a 10K search query log. We invite human judges to manually label the reformulation type between queries. The result is shown in Table 2.

Table 2: Distribution of Reformulation Types

Type	Percentage	Design Strategy
Repeat	42.0%	√
SpellingCorrection	8.9%	√
AddWords	6.6%	√
RemoveWords	2.5%	√
StripURL	2.2%	√
Multiple Reformulation	18.1%	×
Unknown Reformulation	15.6%	×
Others	4.1%	×

We design reformulation-based strategies for the first five types in Table 2. These query reformulation types cover 62.2% of the overall reformulations. As *Multiple Reformulation* is the hybrid of different reformulation types, we design no strategy for it. *Unknown Reformulation* is the reformulation types that cannot be captured by the reformulation taxonomy. From our empirical evaluation, we observe that most of *Unknown Reformulation* need external information such as SERPs to infer the relevance. Thus, no query reformulation-based preference derivation strategy is designed for *Unknown Reformulation.* As query reformulation types in *Others* only take up a small proportion, no strategy is designed for it either. However, in practice the effort of capturing *Multiple Reformulation*, *Unknown Reformulation* and *Others* by the context discovery algorithm would not be wasted. Capturing the three reformulation types is still helpful to glue semantically related queries in the same search context, from which we can derive preference judgment by *Click $>_{q'}$ No-Click Earlier*, otherwise related queries would be separated into different search contexts.

Table 3: Click/Skip Pattern of Reformulation Types

Type	S-S	C-S	S-C	C-C(S)	C-C(D)
Repeat	51.7%	13.0%	14.6%	7.6%	13.1%
SpellingCorrection	45.0%	5.1%	47.2%	0.7%	2.0%
AddWords	28.8%	14.4%	32.4%	2.3%	22.1%
RemoveWords	41.4%	9.5%	28.4%	1.7%	19.0%
URLStripping	40.4%	5.8%	42.3%	5.8%	5.7%

In order to better understand the nature of the five query reformulation types, we also conduct an empirical study on the user's Click/Skip behavior before and after query reformulation. A query is labeled as Click(C) if it results in clickthrough, otherwise, it is labeled as Skip(S). Five Click/Skip patterns and their proportions for each reformulation type are listed in Table 3. In the table, the C-C pattern is further categorized as C-C(S/D) if the clicked URL is the same/different for the two queries. We notice that for each reformulation type, S-S and C-S patterns take a large proportion of the observed behavior patterns, indicating that query reformulation usually fails to bring satisfactory results. As there is no guarantee that a reformulated query would bring better result, we consider a reformulated query closer to the user's information need only if it results in some clickthrough.

4. Repeat Strategy: *Assume q' is an earlier query of q within the search context and q is the repeated query of q'. $(s'_1, s'_2, ...)$ is the ranking of snippets arising from q' and $(s_1, s_2, ...)$ is the ranking of snippets arising from q. The combined snippet ranking is denoted as $(s^c_1, s^c_2, ...)$ and s^c_i is labeled as clicked if s_i or s'_i is clicked. Derive preference using **Click $>_q$ Skip Above** and **Click $>_q$ No-Click Next** from the combined ranking list instead of original ones.*

After Repeat reformulation, the second query brings the same result as the previous one. For S-C pattern and C-C(D) pattern, applying the three clickthrough-based strategies would generate conflicting preference judgment as well as redundant preference judgment. For C-C(S) pattern, applying the three clickthrough-based strategies would generate redundant preference judgment. In order to alleviate this problem, we propose *Repeat Strategy* to avoid the undesirable effects arising from "judgment conflict" and "judgment inflation".

5. SpellingCorrection Strategy: *Assume q' is an earlier query of q within a search context. If q results in clickthrough while q' results in no clickthrough, only derive preference judgment for q by **Click $>_q$ Skip Above** and **Click $>_q$ No-Click Next**.*

S-C pattern takes a large proportion for SpellingCorrection. We infer that the user usually corrects the spelling mistake and does the search again. Such behavior pattern results from clicking the query suggested by the search engine. As there may be no real "preference judgment" made by the user, we constrain that for S-C pattern of SpellingCorrection, preference judgment is only derived from the search result of the second query.

6. AddWords Strategy: *Assume q' is an earlier query of q within a search context. Terms belong to q/q' are denoted by the set S. If q results in clickthrough, concepts containing terms in S is preferred over those skipped concepts for query q.*

Through AddWords reformulation, the second query q

contains some terms that do not exist in the first query q'. We denote these terms as q/q', which form a set S. If the second query results in clickthrough, we infer that these terms successfully represent the user's information need and thus concepts containing these terms should be preferred to concepts in skipped snippets. From the Click/Skip pattern, we observe that S-C and C-C(D) take a relative large proportion. For these two patterns, the clickthrough indicates that the added terms bring results that are closer to the user's information need.

7. RemoveWords Strategy: *Assume q' is an earlier query of q within a search context. Terms belong to q'/q are denoted by the set S. If q results in clickthrough, clicked concepts is preferred over those containing terms in S for query q.*

Through RemoveWords reformulation, some terms in query q' are removed from q. We denote these terms as q'/q, which form a set S. If the second query results in clickthrough, we infer that the user is looking for more general information than that obtained from q'. Query terms in earlier queries that make the search result too specific would not be preferred by the user and clicked concepts are preferred to concepts containing these obsolete terms. From the Click/Skip pattern, we also observe that S-C and C-C(D) take a large proportion for RemoveWords. For these two patterns, the clickthrough indicates that removing these terms successfully brings relevant results to the user.

8. StripURL Strategy: *Assume q' is an earlier query of q within a search context. If q is obtained by stripping URL from q', snippets whose URLs sharing terms with q are preferred by the user.*

StripURL implies that the user has navigational search intent, i.e., the user is looking for a specific Web page rather than some informational contents. In our data set, through StripURL reformulation, 80.64% of the clicked URLs share terms with the reformulated query. In contrast to other query reformulation-based strategies, we capture the user's preference with respect to URLs rather than concepts. Thus, we design this strategy to promote results whose URLs share terms with the reformulated query.

6. PERSONALIZED RANKING FUNCTION

Based on the preference judgment obtained from the previous section, we evaluate each concept's attractiveness and generate a ranking function that utilizes weighted concepts to personalize search results.

Suppose we have input preference judgment over concepts c_i and c_j for a given query q in the following form: $c_i >_q c_j$. We write the above preference as a constraint as follows:

$$\vec{w} \cdot \Phi(q, c_i) > \vec{w} \cdot \Phi(q, c_j),$$

where Φ is a function that maps a query to a concept vector. RSVM [4] is employed to find $\vec{w}$ by which the maximal pairwise preference judgment can be satisfied.

The mapping Φ determines which kind of function RSVM to learn. We first build a *concept space*, which contains concepts in the preference judgment as well concepts have semantic relation with them in the concept ontology. Then, for each c_i arising from q, we create a feature vector $\phi(q, c_i) = (f_1, f_2, \ldots, f_n)$ based on all concepts in the *concept space.* The value of each feature is given by:

$$f_k = \begin{cases} 1 & if \quad k = i \\ sim(c_i, c_k) & if \quad sim(c_i, c_k) > 0 \\ 0 & otherwise. \end{cases}$$

The benefit of building *concept space* from concept ontology is to capture the semantic relation between concepts. For example, a user submitted the query "sun" and clicked on a snippet containing concept "star". From the concept ontology, we observe that "star", "planet" and "solar system" have relative strong relation, thus we infer that these concepts would be favored by the user and results interpreting "sun" as a company or a newspaper would not be relevant to the user's information need.

The semantic relation between two concepts, $sim(c_i, c_k)$, is measured by *Pointwise Mutual Information* (PMI). Let $p(c_i)$ and $p(c_j)$ denote the probability of c_i and c_j. Let $p(c_i, c_j)$ be the joint probability of c_i and c_j. The formula used to calculate PMI is given by:

$$PMI(c_i, c_j) = \log\left(\frac{p(c_i, c_j)}{p(c_i)p(c_j)}\right),$$

$$p(c_i) = N_{c_i}/N; p(c_j) = N_{c_j}/N; p(c_i, c_j) = N_{c_i,c_j}/N,$$

where N_{c_i} is the number of snippets containing c_i, N_{c_j} and is the number of snippets containing c_j and N_{c_i,c_j} is the number of snippets containing both c_i and c_j. N is the total number of snippets. The value of N_{c_i}, N_{c_j}, N_{c_i,c_j} and N are obtained from statistics stored in the concept ontology. PMI may yield negative values; in this case we replace it by zero. In this work, $sim(q, c_i, c_k)$ is replaced by normalized PMI which is denoted by $PMI_N(c_i, c_j)$. We adapt the formula of normalized PMI used in [2] as follows:

$$PMI_N(c_i, c_j) = PMI(c_i, c_j)/(-\log P(c_i, c_j))$$

Suppose the original ranking of search results arising from q is denoted by $R(s_1, s_2, ..., s_n)$, where s_i denotes the snippet ranked at position i. Let C_i be the set of concepts extracted from snippet s_i. The preference score for s_i is obtained by summing up the attractiveness of each concept in C_i weighted by $\vec{w}$.

$$P(s_i) = \sum_{\forall c \in C_i} \vec{w} \cdot \phi(q, c)$$

Based on the preference score of each snippet, we obtain a new ranking and present it to the user. Note that the aforementioned re-ranking method will not be apply to StripURL, which relies on URLs rather than concepts. For StripURL, we simply apply *StripURL Strategy.*

7. EXPERIMENTS

In this Section, we present a comprehensive evaluation of the proposed personalization framework. In Section 7.1, we detail the experimental setup. In Section 7.2, we compare the performance of our search context discovery algorithm with several baseline methods. In Sections 7.3 and 7.4, we evaluate the clickthrough-based strategies and query reformulation-based strategies. In Section 7.5, we evaluate the effectiveness of combining preference derivation strategies and provide an example of the obtained CAUPs.

7.1 Experimental Setup

For the evaluation of search context discovery, we prepare the experimental data from AOL search query log. Each record contains the attributes of user ID, query, timestamp and the URLs that the user clicked. The 8,852 search records are manually segmented into 4,598 search contexts, which are used as the ground truth when evaluating our context discovery algorithm. Table 4 shows the statistics of our experimental data. We compare our context discovery method with several baseline methods such as fixed temporal cutoff and the taxonomy-based method proposed in [3].

Table 4: Statistics of the Experimental Data

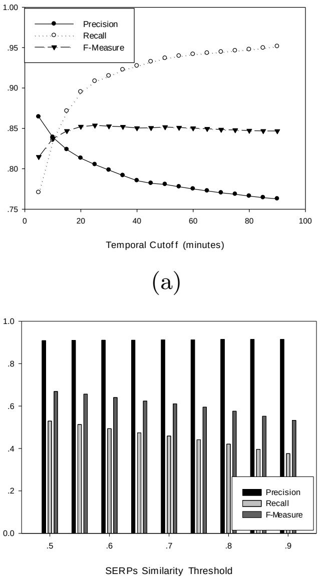
Number of users	215
Number of queries	8,852
Average number of queries per user	41.2
Average number of clicks per user	31.3
Number of search contexts	4,254

For the evaluation of personalization strategies, we compare our approach with two methods proposed in [8], which shows state-of-art of performance to capture the user's conceptual preference. In the evaluation of traditional information retrieval systems [14], expert judges are employed to judge the relevance of a set of documents (e.g., TREC) based on a description of the information need. However, the same evaluation method cannot be applied to personalize Web search, because the same query issued by two different users may have different goals behind it. Thus, instead of having expert judges to evaluate the results with optimized information goals, we invite the users to examine the results and judge what they would consider as relevant for precision computation. A similar evaluation approach has also been used in [13]. We collect the experimental data from the prototype shown in Figure 1. The users were asked to perform relevance judgment on the top 100 results for each query by filling in a score for each search result to reflect the relevance of the search result to the query. We define three level of relevancy (Good, Fair and Poor) on documents. Documents rated as "Good" are considered relevant while those rated as "Poor" are considered irrelevant to the user's information need. Documents rated as "Fair" are treated as unlabeled. In our experiments, documents rated as "Good" are used to compute the Top-N precisions for different methods.

7.2 Evaluation of Search Context Discovery

We evaluate the performance of our search context discovery approach in terms of coherency (Precision), integrity (Recall) and overall performance (F-measure). Precision is defined as the fraction of true topic continuation in all the detected topic continuations. Recall is the fraction of true topic continuation that has been detected by the method. F-measure is a metric involving both precision and recall and it is defined as $(2 \times Precision \times Recall)/(Precision + Recall)$.

We first evaluate the effectiveness of fixed temporal cutoffs and the result is shown in Figure 4(a). The result demonstrates that a cutoff between 25 minutes to 30 minutes is a fairly good cutoff for balancing precision and recall because it achieves the highest F-measure. Thus, we choose 30 minutes cutoff to be the default value for context discovery methods involving temporal cutoff. The evaluation result of SERPs similarity threshold is presented in Figure 4(b). We observe that SERPs similarity thresholds between 0.5 and 0.9 demonstrate very high precision and low recall. In order to strike a balance between precision and recall, we choose threshold 0.75 to be the default value for context discovery methods involving SERPs similarity.



(a)

(b)

Figure 4: Temporal Cutoff and SERPs Threshold

The performance comparison of different context discovery methods is presented in Table 5. We observe that the temporal cutoff method tends to group both relevant and irrelevant queries together, and thus results in a high recall (92.73%), but low precision (78.48%). The low precision results from the risky assumption that queries within a period should be submitted to satisfy the same information need. The Single Reformulation method is solely based on the taxonomy proposed in [3] and demonstrates a high precision (93.13%). The high precision shows that, by utilizing strict reformulation rules, Single Reformulation can effectively detect query relatedness. However, a major drawback of Single Reformulation is that reformulation rules are too strict, and thus it cannot effectively discover similar queries (e.g., semantically rephrased queries) that are not covered by the reformulation rules, resulting low recall (66.33%). Low recall indicates that Single Reformulation tends to split a search context into several separated ones and therefore context integrity is damaged.

New Reformulation is the method based on the reformulation taxonomy in Figure 3. It yields fairly good precision (86.20%) and recall (84.40%). We observe that, by incorporating multiple reformulation, it boosts the recall by 18.07% while sacrifices 11.00% precision. The *Multiple Reformulation* type contributes a lot to the recall improvement. The method only using SERPs similarity is denoted by SERPs, which yields a precision of 91.23% and a recall of 44.14%. The result is consistent with the argument in [9], which reported that methods based on document overlap suffer from data sparseness problem, meaning that the chance for two queries to have common documents is very low. The method incorporating new reformulation taxonomy and SERPs sim-

Table 5: Comparison of Search Context Discovery Methods

Search Context Discovery Methods	Precision	Recall	F-measure	Capture Query Reformulation
Temporal Cutoff	78.48%	92.73%	0.8501	×
Single Reformulation	93.13%	66.33%	0.7748	√
New Reformulation	86.15%	84.38%	0.8525	√
SERPs	91.23%	44.15%	0.5950	×
SERPs and New Reformulation	86.15%	84.75%	0.8544	√
Temporal Cutoff and New Reformulation	96.89%	78.01%	0.8643	√
Temporal Cutoff and SERPs	99.41%	40.28%	0.5733	×
Temporal Cutoff, New Reformulation and SERPs	96.89%	78.36%	0.8664	√

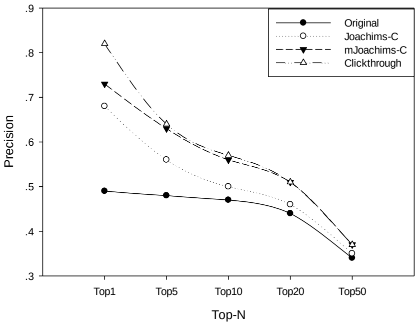


Figure 5: Effectiveness of Clickthrough-based Strategies

ilarity boosts the precision to 99.40%. However, it still suffers from the data sparseness problem, and thus yielding low recall (40.30%).

The method combining New Reformulation and temporal cutoff achieves good performance, yielding a precision of 96.89% and a recall of 78.01%. Compared with New reformulation, this approach improves the precision by 10.74% while sacrifices the recall by 6.37%. The result indicates that temporal cutoff is a good complementary factor for improving precision. The method based SERPs similarity and temporal cutoff shown the highest precision (99.41%), however, due to the data sparseness inherent from SERPs similarity, the method also has the lowest recall (40.28%).

Finally, by integrally using temporal cutoff, New Reformulation and SERPs similarity, the context discovery method proposed in Algorithm 1 yields a high precision (96.89%) while keeps a good recall (78.36%). It also achieves the highest F-measure. We use this method in our prototype and experiments.

7.3 Personalization with Clickthrough Strategies

Following the definitions in [8], the method embodying *Click* $>_q$ *Skip Above* is denoted as *Joachims-C* and the method embodying both *Click* $>_q$ *Skip Above* and *Click* $>_q$ *No-Click Next* is denoted as *mJoachims-C*. As the two methods have been proved useful in [8], we compare them with a new method taking into all the three clickthrough-based strategies. The new method is denoted by *Clickthrough*.

Figure 5 shows the Top-N precisions of the three methods. The Top-N precisions of all the three personalization methods have an obvious increase compared to the original ranking, meaning that all the obtained user profiles can correctly capture the users' preferences. Comparing to the original ranking, *Joachims-C* and *mJoachims-C* have a significant improvement in Top1 (19% and 24%), Top5 (18% and 25%) and Top10 precisions (3% and 9%). However, the improvement for Top20 (2% and 5%) and Top50 (1% and 3%) are not so obvious. We also observe that the extra preferences inferred by *mJoachims-C* help improve the personalization effectiveness. For example, if only the first search result is clicked, *Joachims-C* does not generate any preference judgment, while *mJoachims-C* can still deduce some preference judgments with *Click* $>_q$ *No-Click Next* to personalize the search results.

By introducing contextual clickthrough information, the *Clickthrough* method demonstrates significant improvement comparing to *Joachims-C* and *mJoachims-C* in term of Top1 precision(increased by 14% and 9%). We observe that if a user does not click on any search result for a particular query, *Joachims-C* and *mJoachims-C* then obtain no preference judgment, and thus no personalization can be done for the query. On the other hand, the *Clickthrough* method can still deduce preference judgment by utilizing *Click* $>_{q'}$ *No-Click Earlier*. There are two reasons for the performance improvement of incorporating *Click* $>_{q'}$ *No-Click Earlier*: (1) This strategy is robust to queries that result in no clickthrough; (2) It enables concepts raised from different queries comparable and therefore results in better information coverage. Utilization of the two information sources helps yield higher precision comparing to *Joachims-C* and *mJoachims-C*.

7.4 Personalization with Reformulation Strategies

We now study the effectiveness of each query reformulation-based strategy. In case the evaluation is biased by other factors, for each reformulation type, we sample 50 search contexts only containing the corresponding reformulation type.

We evaluate each strategy by comparing with the original ranking from search engine (denoted as Original) and the personalized ranking by integrally applied the three clickthrough-based strategies (denoted as Clickthrough). The method of applying all three clickthrough-based strategies as well as the reformulation strategy under evaluation is denoted by (Clickthrough+ *corresponding* Strategy). The experiment results are shown in Figure 6. A special case is the *StripURL Strategy*, which is applied without any clickthrough-based strategies.

We observe that *Repeat Strategy* is effective in boosting the overall precisions. By applying *Repeat Strategy* reformulation, the precision is improved by 8.4% comparing to that without applying it for Top5 precision, 7.8% for Top10 precision, 5.5% for Top20 precision, and 4.5% for Top50 precision. The result validates that, by removing judgment in-

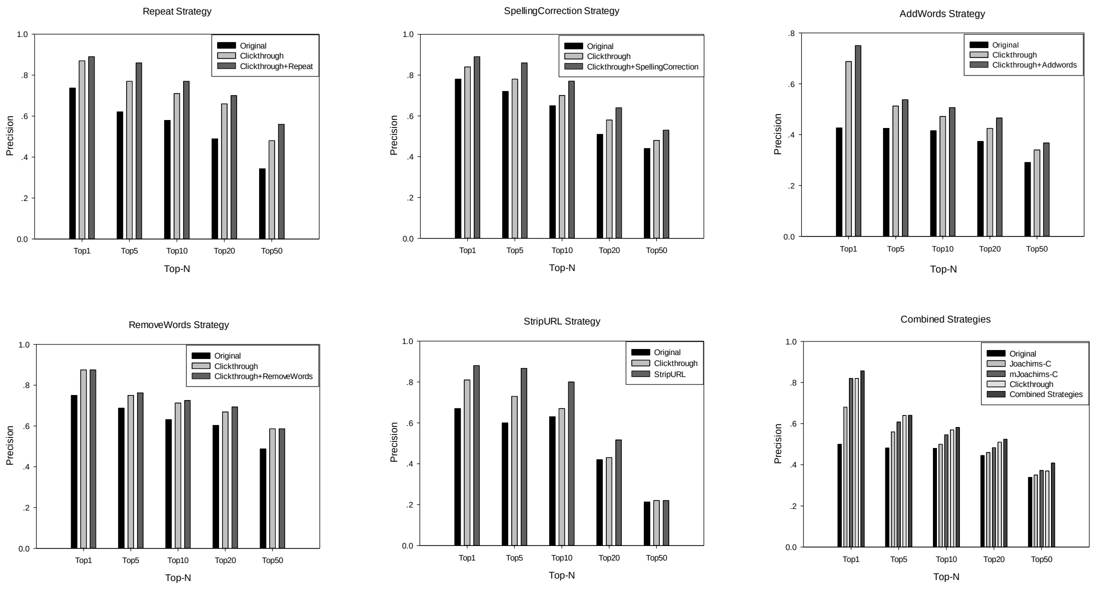


Figure 6: Effectiveness of Query Reformulation-based Strategies and Combined Strategy

flation and judgment conflict, this strategy avoids distorting the user's real search preference and thus boosts the overall personalization effectiveness, especially for Top1, Top5 and Top10 precision.

By combining *SpellingCorrection Strategy*, the Top1 and Top5 precisions have been improved by roughly 5% compared with that without applying this strategy. We also observe that the *Clickthrough* method is plagued by *Click* $>_{q'}$ *No-Click Earlier* for two reasons. First, for some queries with spelling mistakes, the backend search engine can successfully correct them and brings relevant results, in this case, the top two results of the first query may be relevant to the user's information need, even though no clickthrough is raised by the user. Second, if the spelling mistake cannot be corrected by the backend search engine, *Click* $>_{q'}$ *No-Click Earlier* generates preference judgment with some information raised from an irrelevant query and these judgment are not helpful to improve the result quality of current query. By applying this strategy, we avoid making such corrupted preference judgment and thus the personalization performance is improved accordingly.

AddWords Strategy is very helpful to boost the Top-N precision. By applying this strategy, the precision is improved by 6.25% comparing to that without it for Top1 precision, 2.5% for Top5 precision, 3.4% for Top10 precision, 4.1% for Top20 precision, and 2.8% for Top50 precision. It can significantly improve the Top1 precision, showing that when a user reformulates his/her query by adding some terms, these terms bear valuable information for inferring his/her preference. Thus, if a user clicks on the search results of a reformulated query, it is very likely that the reformulated query with the added terms successfully represents the user's actual information need and thus brings satisfactory search results.

RemoveWords Strategy is not as effective as the aforementioned strategies. The performance improvement obtained by applying this strategy is not obvious. For Top1 precision and Top5 precision, the improvement obtained by applying this strategy is less than 2%. The reason is that this strategy's effectiveness highly relies on the amount of concepts which are raised by the reformulated query and also contain the discarded terms. We observe that very few concepts raised from the new query contains the obsolete terms, and thus only a few additional preference judgments can be deduced for the personalization. The sparseness of preference judgment limits the effectiveness of this strategy.

Table 6: Example Queries and the Clickthroughs

Query	**Clicked URL**
q_1 =sun	oracle.com/sun
q_2 =sun microsystems	wikipedia.org/sun_micro..
q_3 =sun microsystems history	thocp.net/sun_micro..

StripURL Strategy is effective for improving the quality of Top1, Top5 and Top10 results. We observe that by applying this strategy, the personalized result outperforms the original ranking as well as that obtained by applying *Clickthrough* method. In the experiments, we also find that the extent of improvement relies on the popularity of the query. If the query itself is a popular navigational query, such as "www.apple.com", the original result obtained from the backend search engine is already very good and therefore the improvement is limited. However, if the query is not so popular, this strategy can effectively boost results with preferred URLs and thus brings a better personalization performance.

7.5 Personalization with Combined Strategies

In this section, we demonstrate the effectiveness of integrally applying the eight proposed strategies on fifty search contexts. We denote the combined strategies as *Combined Strategies* and compare it with the original ranking, *Joachims-C*, *mJoachims-C* and *Clickthrough*. The result is shown in Figure 6. We observe that the *Combined Strategies* can effectively improve the quality of the top-ranked search results and achieves the best performance in terms of Top-N precision.

Table 7: Example CAUPs for q_3="sun microsystems history"

Concept	Joachims-C	mJoachims-C	Clickthrough	Combined Strategies
company	0.4136	1.1500	1.1704	1.0272
oracle and sun	-0.2898	-0.4053	-0.4931	0.3205
solar system	-0.0318	-0.2293	-0.2754	-0.2220
technology	0.4873	0.4327	0.3713	0.2880
history	-0.1761	-0.5672	-0.4366	0.4118
sport	-0.3423	-0.2293	-0.4478	-0.4644

In order to gain a better insight for the profiles obtained from different approaches, we provide an example of the CAUPs obtained by different preference derivation methods. Table 6 shows the queries and the clickthroughs for a user who is searching for the history of "Sun Microsystems". Table 7 shows the user profile obtained from the four preference derivation methods running on the data in Table 6. We observe that the *Joachims-C*, *mJoachims-C*, and *Clickthrough* methods predicted the wrong preferences on the concepts "oracle and sun' and "history" due to the limited information coverage. The *Combined Strategies* can fix up the wrong preferences the concepts "oracle and sun' and "history". It predicts the correct preferences (i.e., positive preferences on "company", "oracle and sun", "technology", and "history", while negative preferences on "solar system" and "sport") from the input training data as shown in Table 6. Comparing to the existing *Joachims-C* and *mJoachims-C* methods, the extra cost of CAUP profiling is minimal. The extra cost comes from the additional preferences derived from the simple strategies (i.e. the third clickthrough-based strategy and query reformulation-based strategies), and it is minimal comparing to the overall cost spent in the training and ranking, which is roughly the same for all the strategies.

8. CONCLUSIONS

In this paper, we study the problem of using search contexts to facilitate concept-based search personalization. We introduce a new method that discovers search contexts from raw search query log. This method captures comprehensive information through one-pass of the query log and has good performance in keeping context coherency and integrity. Eight strategies are then developed to infer the user's conceptual preference from the search context. Through mining clickthroughs and query reformulations, the proposed strategies is able to capture the user's preference with higher accuracy than existing approaches that only consider individual query. We further adopt a learning-to-rank approach to seamlessly combine preference judgment derived from different strategies and update the *Context-Aware User Profile*, which is used to personalize the search results returned from the backend search engine. Empirical studies show that our proposed personalization framework yields higher Top-N precision compared to those without considering contextual information. Importantly, the extra cost of incorporating CAUP into personalized web searching is minimal, since we only incorporates a few more simple rules that manipulate the concept weight in the re-ranking process.

9. ACKNOWLEDGEMENTS

This work is partially supported by RGC GRF under grant number HKUST 618509.

10. REFERENCES

[1] Daniel Gayo-Avello, *A survey on session detection methods in query logs and a proposal for future evaluation*, Information Sciences **179** (2009).

[2] A. Herdagdelen and et al., *Generalized syntactic and semantic models of query reformulation*, Proc. of the SIGIR Conference, 2010.

[3] J. Huang and E. N. Efthimiadis, *Analyzing and evaluating query reformulation strategies in web search logs*, Proc. of the CIKM Conference, 2009.

[4] T. Joachims, *Optimizing search engines using clickthrough data*, Proc. of the SIGKDD Conference, 2002.

[5] T. Joachims and et al., *Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search*, ACM TOIS **25** (2007).

[6] R. Jones and K. L. Klinkner, *Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs*, Proc. of the CIKM Conference, 2008.

[7] Y. Ke, L. Deng, W. Ng, and D. L. Lee, *Web dynamics and their ramifications for the development of web search engines*, Computer Networks **50** (2006), no. 10.

[8] K. W. T Leung and D. L. Lee, *Deriving concept-based user profiles from search engine logs*, IEEE TKDE **22** (2010).

[9] K. W. T. Leung, W. Ng, and D. L. Lee, *Personalized concept-based clustering of search engine queries*, IEEE TKDE **20** (2008).

[10] J. Luxenburger, S. Elbassuoni, and G. Weikum, *Matching task profiles and user needs in personalized web search*, Proc. of the CIKM Conference, 2008.

[11] F. Radlinski and T. Joachims, *Query chains: learning to rank from implicit feedback*, Proc. of the SIGKDD Conference, 2005.

[12] X. Shen, B. Tan, and C. X. Zhai, *Context-sensitive information retrieval using implicit feedback*, Proc. of the SIGIR Conference, 2005.

[13] Jaime Teevan, Meredith Ringel Morris, and Steve Bush, *Discovering and using groups to improve personalized search*, Proc. of the ACM WSDM Conference, 2009.

[14] E. Voorhees and D. Harman, *Trec experiment and evaluation in information retrieval*, MIT Press, Cambridge, MA, 2005.

[15] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li, *Context-aware ranking in web search*, Proc. of the SIGIR Conference, 2010.