Personalized Access to Information by Query Reformulation Based on the State of the Current Task and User Profile

Ounas ASFARI, Bich-liên Doan, Yolaine Bourda
SUPELEC/Department of Computer Science
University of Paris-Sud 11
Gif Sur Yvette Cedex, France
Ounas.Asfari@supelec.fr, Bich-Lien.Doan@supelec.fr
Yolaine.Bourda@supelec.fr

Jean-Paul Sansonnet
LIMSI-CNRS
University of Paris-Sud 11
Orsay cedex France
jps@limsi.fr

Abstract—Access to relevant information adapted to the needs and context of the user is a main challenge in the Internet world. Recent studies have dynamically enhanced the user query with the related user's preferences, which are stored in user's profile for providing personalized results; however, not all user preferences are relevant in all situations. As far as we know, these studies don't take into account the state of the current activity during the information retrieval process. This paper presents a method to reformulate user queries depending on his profile together with the state of the task that he is undertaking when the information retrieval process takes place. Here suppose, that the queries we consider are related to some current task at hand. The UML state diagram is chosen for modeling the current task in order to detect the transitions at time intervals with the task state changes. In this paper, a new concept of SRQ (State Reformulated Queries) is introduced and used to reformulate queries. The result obtained by SRQ is more relevant than that obtained by the initial query for the current task at hand.

Keywords—personalization; query reformulation; Information Retrieval; user profile; task modelling.

I. INTRODUCTION

The Internet offers almost unlimited access to information of all kinds. As the volume of the heterogeneous resources on the web increases and the data become more varied, massive response results are issued to user queries. Thus, large amounts of information are generated in which it is often difficult to distinguish relevant information from secondary information or even noise. Present search engines generally handle search queries without considering the changing contexts in which users submit queries. So, it becomes more difficult to obtain proper results.

To overcome the previous problem, studies taking into account the user context are currently undertaken. The user context can be assimilated to all factors that can describe his intentions and perceptions of his surroundings. These factors may cover various aspects: psychological, social, cultural, professional etc. [4]. The so-called Contextual Information Retrieval combines a set of technologies, query knowledge and the user context, in order to deliver the most appropriate answers to his information needs [4].

On the other hand, Web personalization, which is defined as the process of customizing a Web site content and structure to the specific and individual needs of each user without requiring them to ask for it explicitly, is another solution to overcome the previous problem. Studies on personalized search have focused on requiring users to explicitly enter their contextual information including interest topics, bookmarks, etc., and using this contextual information to expand users’ queries or re-rank search results [1]. But forcing users to submit their contextual interests would be a task that users usually prefer not to do. Moreover, it is very difficult for users to define their own contextual interests accurately. Much attention has been paid to learn about user interests automatically by modeling user profiles or user representations [5] [3]. The techniques of query reformulation, which could be considered as one of the personalization techniques, is a solution presented in this domain. One disadvantage of automatic personalization techniques is that they are generally applied out of context. So, not all of the user interests are relevant all of the time, usually only a subset is active for a given situation, and the rest cannot be considered as relevant preferences. Furthermore, these studies on personalized search have focused on requiring the user without requiring them to ask for it explicitly, is another solution to overcome the previous problem. Studies on personalized search have focused on requiring users to explicitly enter their contextual information including interest topics, bookmarks, etc., and using this contextual information to expand users’ queries or re-rank search results [1]. But forcing users to submit their contextual interests would be a task that users usually prefer not to do. Moreover, it is very difficult for users to define their own contextual interests accurately. Much attention has been paid to learn about user interests automatically by modeling user profiles or user representations [5] [3]. The techniques of query reformulation, which could be considered as one of the personalization techniques, is a solution presented in this domain. One disadvantage of automatic personalization techniques is that they are generally applied out of context. So, not all of the user interests are relevant all of the time, usually only a subset is active for a given situation, and the rest cannot be considered as relevant preferences. Furthermore, these studies on personalization and user contexts, however, neglect the task state changes during the information retrieval process. The suggested method in this paper takes into account the user interests and the task states performed by the user during the information retrieval process.

The aim of this paper is to present a method to reformulate user queries depending on the user profile and the state of the task which he is undertaking when the information retrieval process occurs. Moreover, we will consider queries that are related to the task at hand, indeed that are part of it. So, if we consider two formally equivalent queries, their results can be different depending on the task state. Our approach, as a first study, is based on two hypotheses that restrict the domain: Firstly, we consider user queries that are related to a current task at hand, and secondly, this task is modeled by UML state diagrams.

For example, if a user has to organize a workshop, there are many states for this task such as the choice of the workshop topics and the choice of the program committee members, etc. Submitting the same query, different retrieval results of the search engine should be relevant to each state.

The rest of the paper is organized as follows: Section 2 introduces the state of the art, Section 3 introduces the concept of SRQ (State Reformulated Queries), Section 4...
presents the system architecture and then we present the methodology in Section 5 and an example in section 6. Finally, Section 7 gives the conclusion and future work.

II. STATE OF THE ART

In this research, we consider the field of personalized access to information by query reformulation. For this reason, different techniques in personalized information retrieval by query reformulation will be discussed.

A. Query Reformulation Systems based on user profile for information retrieval

A profile is a user model, which specifies the user domain of interest and the preferences that distinguish this user from the others. All queries issued by the same user are evaluated with respect to his specific profile. The same query issued by different users may have different results as it is evaluated using different profiles. Two main approaches based on the user profile to reformulate a query have been proposed: query enrichment and query rewriting.

—The query enrichment process consists in integrating elements of the user profile into the user’s query [2]. The user profile is defined as a list of disjunctive predicates, including selections and joins. Given such a profile, the query enrichment process consists in reformulating the initial user query by adding predicates from this profile. The first step of query enrichment consists in selecting the top K profile predicates which will be used to enrich the user query. In order to be selected, each predicate has to be related to the user query and not to conflict with it. The second step of query enrichment consists in integrating the top K profile predicates to the query. Two strategies can be followed to do this: the generation of a single query or the generation of multiple queries.

—The query rewriting process consists in transforming the user query expressed on the virtual schema so that it can be evaluated on data sources. It aims at determining contributive data sources for query execution and to use their definitions to reformulate the query [7].

B. Contextualized Query

A contextualized query returns relevant results by accounting for the meaning of query terms and the user’s preferences. A contextualized query will minimize the distance between the information need, I, and the query, Q [6]. Distance (I to Q) is minimized by minimizing:

- The use of the wrong concepts in the query to represent the information needs. Prior research suggests Ontology’s for doing that.
- The lack of preferences in the query to constrain the concepts requested. This lack can be minimized by using user profiles.
- The lack of precision in the language used in the query terms. Lexicons which comprise the general vocabulary of a language can minimize this lack of precision in the language by identifying terms with minimal ambiguity.

III. SRQ (State Reformulated Queries)

In the following, we introduce a new concept State Reformulated Queries (SRQ) that was based on previous concepts as the technique of query enrichment.

SRQ are introduced by the reformulation of the initial query, depending on the state of the current task and the user profile. These queries can be handled by studying the various states of the current task, thus for each state the relevant results for the same user query will be different. The task states and user profiles are modeled by predicates of typical form <attribute, value>. Some values of the relevant predicates are added to the initial query, and thus a reformulated query is generated. We suppose that a task is modeled by a UML state diagram to detect the task state changes. Each state in the state diagram has many predicates which are composed of the pairs <attribute, value>. We will present how the SRQ are produced depending user profiles and task states.

We suppose that the user queries are related to current task at hand which has several states modeled by UML state diagram. We suppose also that there is at least one relevant attribute as, for each task state Si:

\[
A_S := \sum_{i \geq 1} A_{S_j} ;
\]

As; set of task state attributes at the state Si.

Let P user profile that is stored in the form of attribute-value pairs (au, vu).

\[
Au_j := \sum_{i \geq 0} au_i ;
\]

Au; set of relevant user profile attributes for the user u.

Q= {t1, t2, ..., tk} the initial query which is related to the task at hand.

The state reformulated query in the current task state s and user profile P is: SRQ= <Q, P, S>

For the state s, and user u:

\[
SRQ < Q, P, S_i > := Q + As_i + Au_j
\]

\[S_i, RQ < Q, P, S_i > \neq S_j, RQ < Q, P, S_j >, \text{ for two different states } S_i, S_j.\]

The relevant results Di in the states Si are produced by applying $SRQ < Q, P, S_i >$ on a search engine or information retrieval system. We suppose that the results Di in the state Si are more pertinent than the normal results produced by using the initial query Q in the states $S_i$

IV. SYSTEM ARCHITECTURE

The method for query reformulation based on the state of the current task includes modeling one or more workflow activities used to perform work tasks. We suppose that there is at least one relevant state attribute, in order to be able to generate state reformulated queries for retrieving information. UML state diagrams (which are composed of a finite number of states) describe the states of activities by showing the sequence of activities performed. Figure 1 shows the system architecture.

V. METHODOLOGY

The methodology uses both user profile and ontology to process web queries.
User Profile

A user profile is a collection of personal data associated to a specific user. A user profile can be either static, when the information it contains is never or rarely altered (e.g., demographic information), or dynamic when the user profile’s data change frequently, (e.g., all the visited pages can be considered as user interests to various degrees). Such information is obtained either explicitly, using on-line registration forms and questionnaires resulting in static user profiles, or implicitly, by recording the navigational behavior and the preferences of each user, resulting in dynamic user profiles.

In our model, a dynamic user profile is considered which is stored in a table in the form of attribute-value pairs where each pair represents a profile’s property. The properties are grouped by categories. There are two types of preferences: Global Constraint (apply to all queries for example: langue, address, age, etc.) and local constraints (apply based on the query context for example: category of restaurant preference, hotel, travel preference, music, videos, etc.)

Ontology

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts; it is used to reason about the properties of that domain and can be used to define the domain. Ontology’s can minimize the use of the wrong concepts in the query by helping users understand relationship between concepts. For example, to find a job as a Professor, ontology might suggest relevant related terms, such as teaching and research. In this methodology, the ontology is used to determine the related user profile concepts in order to calculate the relevant terms that can be used in the initial query reformulation.

The two major aspects of the methodology are query expansion and query refinement. During query expansion, the query is expanded with new terms to improve the retrieval performance. In the other systems the expansion is usually performed using only synonyms of the initial query terms but here we use the state terms and user profile terms.

Query refinement is the incremental process of transforming a query into a new query SRQ that more accurately reflects the user’s information need, (e.g., remove irrelevant terms which are produced during the query expansion process).

The proposed methodology consists of four phases: (1) Query Parsing, (2) Query Expansion, (3) Query Refinement, and (4) Query Submission.

- The query parsing phase involves parsing the initial query using WordNet and identifying from the initial query the synonym terms. From this, a baseline query is created.
- In the query expansion phase, the initial query terms and their synonym terms (the baseline query) $Q=\{t_1, t_2, \ldots, t_n\}$ are queried anew against DAML ontology library to find the related concepts $(c_1, \ldots, c_m)$ with $m \geq n$. These concepts are compared with the user profile concepts $(c_{u1}, \ldots, c_{um})$ in order to find the common concepts that allow finding the user profile attributes, whereas these attributes are selected from the user profile categories that are determined by previous common concepts, and then these attributes are added to the initial query. Finally the relevant state attributes $A_{S_i}$ from UML state diagram, are also added to the initial query.
- The query refinement phase: Sometimes there may be an irrelevant terms in the selected user profile categories, thus for choosing the relevant user profile attributes $A_{S_i}$ for the current task state $S_n$, a similarity function, between the current state attributes and the user profile attributes from selected categories, is applied. Equation (1) is used for producing State Reformulated Query SRQ to better focus on the requirements of the user.
- The query submission phase creates the final query according to the syntax required by the search engine used and submits the query SRQ and provides the results back to the user. After each step, the user is asked if the query reflects his intention. If so, the final query is constructed using the appropriate syntax and submitted to the search engine.

VI. Example

In order to illustrate the challenge we have to face with this approach, a simple state diagram as shown in Figure 2 is taken here; the task will be "organize a visit to Toulouse to see the main monuments such as the Capitol and the Basilica of Saint Cernin". This task is composed of three main states: before the visit, during the visit and after the visit and every main state has also many sub-states as shown in Figure 2. To perform this task let us consider the following queries for example: $Q_1=\{\text{Toulouse}\}$ or $Q_2=\{\text{Monuments in Toulouse}\}$ or $Q_3=\{\text{Toulouse Travel Guide}\}$ or $Q_4=\{\text{Tourism in Toulouse}\}$ or \ldots etc.
TABLE I. THE STATE REFORMULATED QUERIES FOR THE FIRST QUERY Q1.

<table>
<thead>
<tr>
<th>Query</th>
<th>Terms</th>
<th>S1RQ1</th>
<th>S2RQ1</th>
<th>S3RQ1</th>
<th>S4RQ1</th>
<th>S5RQ1</th>
<th>S6RQ1</th>
<th>S7RQ1</th>
<th>S8RQ1</th>
<th>S9RQ1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Toulouse &quot;Toulouse&quot; + &quot;Airline&quot; OR Book Ticket OR Inexpensive</td>
<td>&quot;Toulouse&quot; + &quot;hotel&quot; +2 stars OR single</td>
<td>&quot;Toulouse&quot; + &quot;Monuments&quot; OR Weather</td>
<td>&quot;Toulouse&quot; + &quot;Plan OR &quot;Map&quot; OR &quot;Metro&quot;</td>
<td>&quot;Toulouse&quot; + &quot;restaurant&quot; OR Italian Cuisine OR Vegetarian Food</td>
<td>&quot;Toulouse&quot; + &quot;restaurant&quot; OR Italian Cuisine OR Vegetarian Food</td>
<td>&quot;Toulouse&quot; + &quot;clube&quot; OR &quot;capitol&quot; OR &quot;disco&quot;</td>
<td>&quot;Toulouse&quot; + &quot;News&quot; OR &quot;Photos&quot;</td>
<td>&quot;Toulouse&quot; + &quot;News&quot; OR &quot;Photos&quot;</td>
<td>&quot;Toulouse&quot; + &quot;News&quot; OR &quot;Photos&quot;</td>
</tr>
</tbody>
</table>

VII. CONCLUSION AND FUTURE WORK

As yet we presented in this paper a method which is tested manually to reformulate user queries depending on his profile and the state of the task that he is undertaking when the information retrieval process is performed. To produce the reformulation we suppose that a user query done during a task is related to this task and we choose UML state diagrams to model the task in order to detect the transitions at time intervals with the task state changes. We introduced the concept SRQ (State Reformulated Queries) and how we produce them from the initial user query. We have illustrated on an example that the results obtained by SRQ queries, which are reformulated depending on the current task state and user profile, are more relevant than the results obtained with the initial user queries. In the future we plan to evaluate this method by using a collection of tests. Knowing how we can detect the predicates of each state implicitly is another future aim.

REFERENCES


