Context-based Hybrid Method for User Query Expansion

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

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

Abstract— **Today, there is a real challenge in accessing relevant information on the Web according to the user's needs and the context. There are always certain needs behind the user query and these queries are often ambiguous and shortened (especially in the case of mobile users), thus we need to handle the user queries intelligently to provide personalized results in a particular context. For improving user query processing, we present a context-based hybrid method for query expansion that automatically generates context-related terms. It considers the context as the actual state of the task that the user is undertaking when the information retrieval process takes place. The method uses the UML state diagram for modeling the current task and for detecting the transitions at time intervals with the task state changes. Furthermore, we introduce a new concept of SRQ (State Reformulated Queries), which is used to reformulate queries according to the user task context and the ontological user profile. Using experimental study, our approach has proved its relevance for certain contexts, the preliminary results are promising.**

Keywords- query reformulation; context; task modeling; Information Retrieval; user profile.

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 becomes 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. Recent studies have tried to dynamically enhance the user query with the user's preferences by creating a user profile for providing personalized results [1]. However, a user profile may not be sufficient for a variety of queries of the user. For example a tourist and a programmer may use the same word "java" (Java Island in Indonesia, Java programming language, the Java Coffee, etc.), in some situations the programmer may need information about the Java island that is not found in his preferences. 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.

On the other hand, new devices are constantly appearing and becoming a principle part of our daily lives. the multitude of devices (PC, PDA, cellular phone, etc.) including diverse platforms, the different user knowledge levels, characteristics and expectations, and the various work environments, have created new considerations and stakes to be satisfied [2]. To overcome the previous problems, studies taking into account the user context are currently undertaken. As a result, the information needs of mobile users are related to contextual factors such as user interests, user current task, location, direction, etc.

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: physical, social, personal, professional, technical, task, etc. Fig. 1 shows these factors and examples for each one [3].

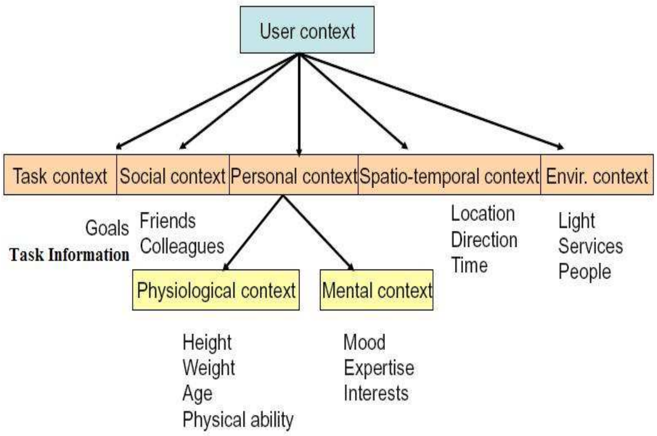

Figure 1. A context model from Kofod-petersen.

The problems to be addressed here include how to represent the context, how to determine it at runtime, and how to use it to influence the activation of user preferences. It is very difficult to take into consideration all the contextual factors in one information retrieval system, so the researchers often define the context as certain factors (location for example).

Thus in this paper our definition of the context is that the context describes the user current task, its changes over time and its states, i.e., we take into account the user current task which the user is undertaking when the information retrieval process occurs.

Queries, especially short one, do not provide a complete specification of the information need. Many relevant terms can be absent from queries and terms included may be ambiguous. Typical solution includes expanding query

 ISBN: 978-1-61208-104-5

representation by exploiting semantic resources [4] or user profile [5]. That refers to methods of query reformulation, i.e., any kind of transformation applied to a query to facilitate a more effective retrieval.

This paper present a method to reformulate user queries depending on the user profile, containing his interests, together with the user context which is considered as the actual state of the user current task in order to provide personalized results in context. Moreover we will consider that the user queries are related to the task at hand, indeed that are part of it. We combine knowledge about query (linguistic knowledge, using WordNet and semantic knowledge using ODP ontology, Open Directory Project, www.dmoz.org) and knowledge about user (user profile and user task context) into a single framework in order to provide the most appropriate answer for a user's information needs in the search time and task state.

For example, if a user has to organize a workshop, many states for this task exist, such as the choice of the workshop topics and the choice of the program committee members, etc. Submitting two equivalent queries in two different states, the relevant results to each task state will be different, so the proposed system has to provide the different relevant results to each state.

The rest of the paper is organized as follows: Section 2 shows the related work; Section 3 introduces the models and algorithms to reformulate user's queries; section 4 presents the architecture of our system; Section 5 shows the experimental study and examples Finally, Section 6 gives the conclusion and future work to be done.

II. RELATED WORK

Query expansion is the process of augmenting the user's query with additional terms in order to improve results by including terms that would lead to retrieving more relevant documents. Many works have been done for providing personalized results by query reformulation.

Two main approaches based on the user profile to reformulate a query have been proposed: query enrichment process which consists in integrating elements of the user profile into the user's query [6], the user profile is defined as a list of disjunctive predicates, including selections and joints. Given such a profile, the query enrichment process consists in reformulating the initial user query by adding predicates from this profile. The second approach based on a user profile is the query rewriting process which translates the query to access the real data sources [7].

The limitation of these approaches is that they do not take into consideration the user context for activation the elements from the user profile.

Studies on query reformulation by relevance feedback are proposed, the aim is to use the initial query in order to begin the search and then modify it from the judgments of the relevance and irrelevance to the user. The new complaint obtained in each iteration feedback, can rectify the direction of the research [8]. Because relevance feedback requires the user to select which documents are relevant, it is quite common to use pseudo-relevance feedback.

Furthermore the techniques of disambiguation aim to identify precisely the meaning referred by the terms of the query and focus on the documents containing the words quoted in the context defined by the corresponding meaning [9]. But this disambiguation may cause the query to move in a direction away from the user's intention. For example the query "windows" might be about actual windows in houses or the Microsoft Windows operating system. A system might choose an interpretation different from the user's intention and augment the query with terms related to the wrong interpretation.

Many approaches like [4] try to reformulate the web queries based on semantic knowledge about different application domains from Research-Cyc for example, others use sense information (WordNet in general) to expand the query [10].

Many approaches, for example [11], expand the user initial query by using ontology in order to extract the semantic domain of a word and add the related terms to the initial query. But sometimes these terms are not related to query terms. More precisely they are related to the query but only under a particular context of the specific query.

This paper presents a new approach for improving user query processing. We propose a hybrid query expansion method that automatically generates query expansion terms from the user profile and the user task. In our approach we exploit both a semantic knowledge (ODP Ontology) and a linguistic knowledge (WordNet) to learn the user's task, and we exploit an UML states diagram for one task to learn user current state.

III. MODELS AND ALGORITHMS

Our aim is to provide context-based personalized results. For that, we improve the user web-queries intelligently to address more of the user's intended requirements. We generate a new query language model for the purpose of query reformulation based on the user context and an ontological user profile. We consider the user current task as a contextual factor. Here we will describe our models for detecting the user current task, constructing an ontological user profile and generating the reformulated queries.

A. General Language Model

We construct here a new general language model for query expansion including the contextual factors and user profile in order to estimates the parameters in the model that is relevant to information retrieval systems. In the language modeling framework, a typical score function is defined in KL-divergence as follows [15]:

$$Score(Q,D) = \sum_{t \in V} P(t \mid \theta_Q) \log P(t \mid \theta_D) \propto -KL(\theta_Q \parallel \theta_D) \quad (1)$$

Where: θ_D is a language model created for a document D, θ_Q a language model for the query Q, generally estimated by relative frequency of keywords in the query, and V the vocabulary.

$P(t|\theta_D)$: The probability of term t in the document model,

$P(t|\theta_Q)$: The probability of term t in the query model,

$P(Q \mid D) = \prod P(t \mid \theta_D)^{c(t;Q)}$ $c(t;Q)$: Frequency of term t in query Q;

The basic retrieval operation is still limited to keyword matching, according to a few words in the query. To improve retrieval effectiveness, it is important to create a more complete query model that represents better the information need. In particular, all the related and presumed words should be included in the query model. In these cases, we construct the initial query model containing only the original terms, and a new model SRQ containing the added terms. We generalize this approach and integrate more models for the query. Let us use θ_Q^0 to denote the original query model, θ_Q^T for the task model, θ_Q^S for the contextual state model, and θ_Q^U for a user profile model. θ_Q^0 can be created by MLE (Maximum Likelihood Estimation)[3].

Given these models, we create the following final query model by interpolation:

$$P(t \mid \theta_Q) = \sum_{i \in X} \alpha_i P(t \mid \theta_Q^i) \quad (2)$$

Where: X= {0, T, S, U} is the set of all component models and a_i (with $\sum_{i \in X} a_i = 1$) are their mixture weights.

Thus the (1) becomes:

$$Score(Q,D) = \sum_{t \in V} \sum_{i \in X} \alpha_i P(t \mid \theta_Q^i) \log P(t \mid \theta_D) = \sum_{i \in X} \alpha_i Score_i(Q,D) \quad (3)$$

where:

$$Score_i(Q,D) = \sum_{t \in V} \alpha_i P(t \mid \theta_Q^i) \log P(t \mid \theta_D) \quad (4)$$

is the score according to each component model.

The remaining problem is to construct task model, contextual model and user profile model and to combine all the models.

B. *Constructing Task Model*

The task model is used to detect and describe the task performed by the user, when he submits his query to the information retrieval system. We consider the task as the contextual factor of the user. In this paper we depend on study questionnaires [16], which were used to elicit tasks that were expected to be of interest to subjects during the study. A generic classification was devised for all tasks identified by all subjects, producing the following nine task groupings:

Academic Research; News and Weather; Shopping and Selling; Hobbies and Personal Interests; Jobs/Career/Funding; Entertainment; Personal Communication; Teaching; Travel.

For example, the task labels "viewing news," "read the news," and "check the weather" would be classified in Group 2: "News and Weather."

We generate a UML states diagram for each task in order to detect the changes in the task-needs over time and for describing all the sequences of the performed task. This generated diagram contains the task states and at least one attribute for each one. Accordingly, an index is built for: the terms of the tasks, the terms of its states including the state attributes, and the related task concepts from ODP. Thus this index consists of r terms. We will use this index when using the term vector model.

The user task can be identified in two different ways:

1) Manually, by the user who selects one task from the proposed tasks and assigns the selected task to his queries.

2) Automatically, by taking advantages of existing linguistic (WordNet) and semantic resources (ODP Ontology) for assigning a task to user query.

Here, we use the second way in order to facilitate the process to users. For applying the second way, we apply the following *algorithm*:

Let q be a query submitted by a specific user at the current task denoted A_*. This query is composed of n terms; it can be represented as a single term vector:

$$\vec{q} = \langle t_1, t_2, \ldots, t_n \rangle$$

For this query $\vec{q}$ a current task A_* is built by a single term vector:

$$\vec{A_*} = \langle a_{s1}, a_{s2}, \ldots, a_{si} \rangle$$

Where: a_{S1}, a_{S2}, …a_{Si} the terms that represent the state attributes of the task states s_1, s_2, …s_i for the current task A_*. For example, if the actual state is "Find a Restaurant", then the state attribute will be "Restaurant" and a value from the user profile (such as vegetarian) will be assigned to this state attribute in order to personalize the query.

The initial query q is parsed using WordNet in order to identify the synonymous terms and to build the baseline query:

$$\vec{q_w} = \langle t_{w1}, t_{w2}, \ldots, t_{wn} \rangle$$

The baseline query $\vec{q_w}$ is queried against the ODP ontology in order to extract a set of concepts ($c_1, c_2 \ldots, c_m$ with $m \geq n$) that reflect the semantic knowledge of the user query. These concepts of the user query and its sub-concepts are represented as a single term vector

$$\vec{C_q} = \langle c_1, c_2, \ldots, c_m \rangle$$

Then the concepts are compared with the previous nine tasks, to do this, we compute the similarity weight between $\vec{C_q}$ and the proposed nine tasks, depending on the task index which is previously explained:

$$SW(A_1) = \mathrm{Cos}(\vec{C_q}, \vec{A_1})$$

$$SW(A_2) = \mathrm{Cos}(\vec{C_q}, \vec{A_2})$$

…….

....

$$SW(A_9) = \mathrm{Cos}(\vec{C_q}, \vec{A_9})$$

Finally, the task $\vec{A_*}$ corresponding with the maximum similarity weight (Max (SW (A_*))) is automatically selected as the current task. Fig. 2 shows the various vectors.

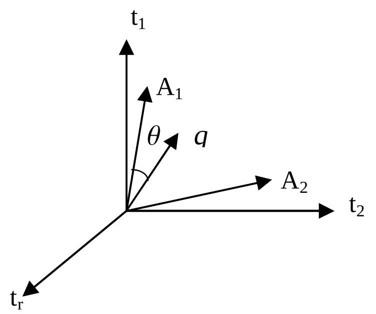

Figure 2. Representation of the tasks and the query as term vectors.

Where: query terms: $t_1, t_2, \ldots, t_n$.

Terms of task index: $t_1, t_2, \ldots, t_r$.

Terms of task state attributes: $a_{S1}, a_{S2}, \ldots, a_{Si}$.

Each term's weight is computed using tf * idf weighting scheme.

For example if the user submits the query *q= {Tourism in Toulouse}*, then the steps of our approach for detecting the user task are shown in Table 1:

TABLE I. APPLYING TASK MODEL TO THE QUERY Q

Description	Knowledge used	Result
parsing the initial query using WordNet	WordNet	A set of query terms (t_1,.., t_n) (tourism, Toulouse) and its synonymous terms (that will be used as the baseline query(services to tourists, touring, travel, city in France)
The concepts in ontology that represent the baseline query terms are identified.	Ontological information from ODP ontology.	A set of the concepts ($c_1, \ldots, c_m$ with $m \geq n$) relevant to the baseline query. (Travel Guides, Travel and Tourism, Vacations and Touring, Touring Cars, Weather, Food, Maps and Views, hotel, University of Toulouse, Commerce and economy)

So, the task that assigned to the user query *q* is: "*travel*" as it has the most similarity weight number.

C. *Contextual State Model*

The contextual state model is responsible for determining and analyzing the actual state of the current task. We suppose that the different states of the current task are modeled using an UML state diagram. There is at least one relevant attribute as_i for each detected state S_i. Because mobile device moves with the user, it is possible to take into account the actual task state in which the user is in when submitting certain queries to the information retrieval system IRS. Such contextual information may come automatically from various sources such as the user's schedule, sensors, entities that interact with the user; it may also be created by the user.

According to our assumption, we have defined 9 UML state diagrams for the main pre-defined tasks. After the user's query is submitted to our platform, the related task is assigned automatically to the user query and a set of SRQ (State Reformulated Queries) related to each state is presented to the user. The user is then asked to choose the appropriate SRQ according to his state. Finally, the contextual model will follow the UML state diagram to present the next SRQ.

D. *Ontological user profile model*

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts so the basic building blocks of ontology are concepts and relationships. Concepts (or classes or categories or types) appear as nodes in the ontology graph.

A *user profile* is a collection of personal data associated to a specific user. The *Ontological user profile* is constructed by the representation of the user profile as a graph of related concepts of the ODP ontology, inferred using an index of user documents. Here, a dynamic ontological user profile is considered as semi-structured data in the form of attribute-value pairs where each pair represents a profile's property.

The properties are grouped in categories or concepts using ODP Ontology, this allows us to help users to understand relationships between concepts, moreover, to avoid the use of wrong concepts inside queries. e.g., for a query "looking for a job as a Professor", ontology suggests relevant related terms: teaching, research etc. for example in the proposed ontological user profile we can find global category (language, address, age…etc.) and local category (preferences of restaurants, hotel, travel, music, videos, etc.), i.e. the annotating of each concept in ODP ontology is done by giving value for each attribute in the ontology concept based on an accumulated similarity with the index of user documents, a user profile is created consisting of all concepts with non null value.

Using ontology as the basis of the profile allows the initial user behavior to be matched with existing concepts in the domain ontology and relationships between these concepts [12]. When the ontological user profile is created, its query-related concepts must be activated. This is done by mapping the query context $C_q = \langle c_1, c_2, \ldots, c_m \rangle$ on this ontological user profile (note that, the query context is calculated during the construction of the task model). This allows to activate for each query context concept its semantically related concepts from the ontological user profile, following our contextual approach depending on the relevant propagation [13]. Hence, the relevant user profile attributes that are determined by the previous activated concepts are found. This attributes with its values are used to reformulate the user query.

E. *SRQ Model (State Reformulated Queries)*

Query expansion is the process of adding relevant terms to the original query [14]. However, in a more general sense,

 ISBN: 978-1-61208-104-5

it also refers to methods of query reformulation, Thus we look for a relevant terms to use it in query expansion. But what do we mean by *relevant terms*?

The terms are relevant if they are related to the query, the user, and the task state in the same time and don't contain unrelated terms. The initial user query is reformulated depending on these relevant terms in order to produce SRQ (State Reformulated Query) to improve the retrieval performance. The two aspects for producing SRQ are query expansion and query refinement.

Query expansion: the initial query is expanded with two type of generated terms:

- The terms that represent the state attributes, from UML state diagram, for the current task A_* (denoted a_{S1}, a_{S2}, …,a_{Si}) One state attribute for each task state.
- The query-relevant attributes from the ontological user profile with its values. (<attribute a_{u1}, value>, <attribute a_{u2}, value>, …,<attribute a_{uj}, value>)

Query refinement: Query refinement is the process of transforming a query into a new query SRQ that more accurately reflects the user's information need. Sometimes irrelevant attributes may be present in the selected user profile concepts. In order to keep only the relevant user profile attributes for the current task state S_i, we compare between these generated attributes and the current state attributes, next we exclude from the generated user profile attributes these non similar with the state attributes. We must also exclude the duplicated terms if they exist in the resulting SRQ.

Another method for filtering the previous terms is by asking the user to choose the relevant terms before adding them to the query.

Finally SRQ is built according to the syntax required by the used search engine in order to submit the query SRQ and to provide back results to the user.

Let *q* an initial query which is composed of many terms $\{t_1, t_2, ..., t_n\}$ and related to the task at hand. The state reformulated query in the task state S_i and for a specific user profile P_j is: $S_iRQ<Q,P_j,S_i>$, The relevant results D_i in the states S_i are produced by applying $S_iRQ<Q,P_j,S_i>$ on an information retrieval system. We expect that the results D_i in the task state S_i are more relevant than the normal results produced by using the initial query *q* in S_i, to check that an experimental study will be performed.

IV. SYSTEM ARCHITECTURE

Fig. 3 presents the system architecture. It combines the several models described in the previous section: the task model, the contextual state model, the ontological user profile model and the SRQ model.

V. EXPERIMENTAL STUDY

Here we first suppose that the queries we are considering are related to some current task at hand and secondly, the tasks are modeled by UML state diagrams. We can show that our system works depending on the following practical consequent steps:

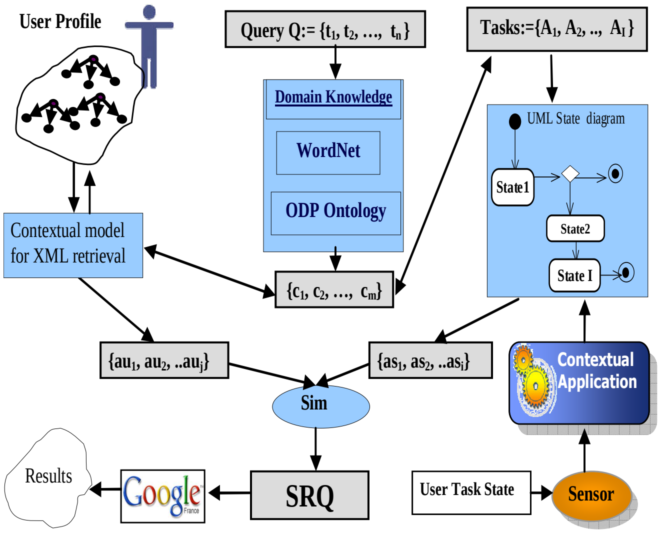

Figure 3. System architecture.

When the user submits his query in our platform, the system will detect the user current task (described in task model, section III paragraph B) as the first step. Next, the UML state diagram for this task is retrieved (section III paragraph C). The system then uses the attributes associated with each state (in UML) and the user profile attributes for producing the relevant terms (methodology section III paragraph E). The irrelevant terms are excluded (The query refinement). Finally, the reformulated query denoted SRQ is submitted to Google to retrieve the relevant results.

For instance, Let us consider the query *q*= {Buy Laptop}, the task assigned to the user query *q* is: "Shopping and Selling". The contextual state model allows the proposition of several task states that are represented in UML state diagram as shown in the fig.4. For this task the system can produce the following SRQ:

- S_1 (Information about laptop models): S_1RQ: {"laptop"+ information}.
- S_2 (model choice): S_2RQ:{"laptop"+ HP OR Asus}.
- S_3 (comparing prices): S_3RQ:{ "laptop"+ price OR Inexpensive}.
- S_4 (choosing a computer shop): S_4RQ: {"laptop"+ address OR London}.

Table 2 presents the state reformulated queries SRQ for the query *q* and their relevance score using the first 20 retrieval results of Google. For example, at the first task state S_1 which is "general information about laptop models", there are 11 relevant results of 20 retrieved by Google using the user query *q* without reformulation, while there are 14 relevant results of 20 using the SRQ.

The evaluation of such systems is complicated due to the dynamic aspect of the system environment. So, we performed two manual evaluations, one to evaluate the detected task and another to evaluate the SRQ (State Reformulated Queries):

We asked 10 different users to submit 3 queries (for doing different tasks) the system then detects the task for each query. Next the users are asked if the queries were their tasks or not. We then got nearly 21 out of 30 positive responses (70%).

To evaluate the SRQ queries we asked the 10 users to submit different queries and we applied each one to the Google search engine at the different states of the task which was detected by our task model. We reformulated these queries by adding the relevant terms and then we reapplied them at the states using the same search engine. We compared the first 20 retrieval results produced in the two cases (by queries *q* and queries SRQ).

Results: we calculated the average number of relevant pages by queries *q* and SRQ on the first 20 results (P@20). We noticed that the precision of the relevant results using the initial query *q* is 0.17 and 0.59, respectively, by using SRQ queries which were reformulated depending on the current task state and user profile.

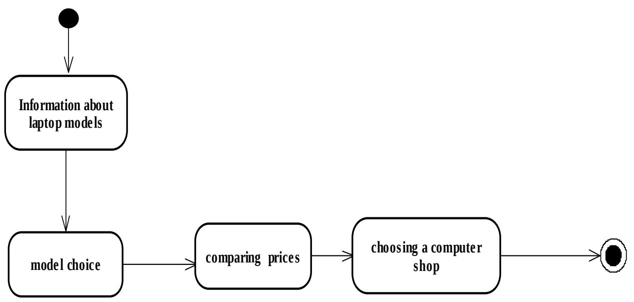

Figure 4. shows an example of a task that is modeled by UML state diagram.

TABLE II. THE STATE REFORMULATED QUERIES FOR THE QUERY Q.

Query	Q				S_1RQ	S_2RQ	S_3RQ	S_4RQ
Terms	Buy laptop				"laptop"+ information	"laptop" + HP OR Asus	"laptop" + price OR Inexpensive	"laptop" + address OR London
P@20	S_1	S_2	S_3	S_4	14	15	8	7
	11	2	4	1				

VI. CONCLUSIONS

In this paper, we have proposed a hybrid method to reformulate user queries depending on a dynamic ontological user profile and user context for producing State Reformulated Queries (SRQ). The user context is considered as the actual state of the task that he is undertaking when the information retrieval process is performed. We have constructed a general architecture that combines several models for query expansion: the task model, the contextual model, the user profile retrieval model and SRQ model. We exploit both a semantic knowledge (ODP Ontology) and a linguistic knowledge (WordNet) to learn user's task, and we exploit a UML states diagram for this task to learn user current state. We have also constructed a new general language model for query expansion including the contextual factors and user profile. We have illustrated on an experimental study that the results obtained by SRQ queries are more relevant than those obtained with the initial user queries in the same task state. As a future work, we plan to evaluate this method by creating a test collection.

REFERENCES

[1] Micarelli, A., Gasparetti, F., Sciarrone, F., and Gauch, S.: Personalized Search on the World Wide Web. P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.), In: The Adaptive Web, LNCS 4321, pp. 195–230, Berlin, 2007.

[2] Liiv I., Tammet T., Ruotsalo, T., and Kuusik A.: Personalized Context-Aware Recommendations in SMARTMUSEUM Combining Semantics with Statistics. In: 2009 Third International Conference on Advances in Semantic Processing, Malta 2009.

[3] Kofod-Petersen, A. and Cassens, J.: Using Activity Theory to Model Context Awarenessa, In: American Association for Artificial Intelligence, Berlin, 2006.

[4] Conesa, J., Storey, V.C., and Sugumaran, V.: Using Semantic Knowledge to Improve Web Query Processing, In: NLDB 2006, pp. 106 – 117, Springer-Verlag Berlin, 2006.

[5] Chirita, P. A, Nejdl, W., Paiu, R., and Kohlschutter, Ch.: Using ODP Metadata to Personalize Search. In: Proc. of the 28th Annual Int'l ACM SIGIR Conf., 178–185, Brazil, 2005.

[6] Koutrika, G., and Ioannidis, Y. E.: Personalization of Queries in Database Systems. In: Proceedings of the 20th International Conference on Data Engineering, USA, 2004.

[7] Vidal, M.E., Raschid, L., Marquez, N., Cardenas, M., and Wu, Y.: Query Rewriting in the Semantic Web. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDEW, USA, 2006.

[8] Baeza-Yates, R., Hurtado, C., and Mendoza, M.: Query recommendation using query logs in search engines. In EDBT '04, 588-596, 2004

[9] Wakaki, H., Masada, T., Takasu, A., and Adachi, J.: A New Measure for Query Disambiguation Using Term Co-occurrences. In: Lecture Notes Computer Science, 2006, NUMB 4224, pages 904-911.

[10] Navigli, R. and Velardi, P.: An Analysis of Ontology-based Query Expansion Strategies. Workshop on Adaptive Text Extraction and Mining at the 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia (2003).

[11] Bhogal, J., Macfarlane, A., and Smith, P.: A review of ontology based query expansion, Information Processing and Management. In: an International Journal, v.43 n.4, p.866-886, July, 2007, doi: 10.1016/j.ipm.2006.09.003.

[12] Sieg, A., Mobasher, B., and Burke, R.: Representing context in web search with ontological user profiles. In: Proceedings of the Sixth International Conference on Modeling and Using Context, Roskilde, Denmark, August 2007.

[13] Asfari, O.: Modèle de recherche contextuelle orientée contenu pour un corpus de documents XML. In: CORIA 2008: 377-384, France.

[14] Asfari, O., Doan, B. L., Bourda, Y., and Sansonnet, J.-P.: Personalized access to information by query reformulation based on the state of the current task and user profile, In: The Third International Conference on Advances in Semantic Processing, SEMAPRO, Malta, 2009.

[15] Bouchard, H. and Nie, J.Y.: Modèles de langue appliqués à la recherche d'information contextuelle, In: Conf. en Recherche d'Information et Applications (CORIA), Lyon, 2006.

[16] W. White, R. and Kelly, D.: A Study on the Effects of Personalization and Task Information on Implicit Feedback Performance, In: CIKM'06, 2006, USA.