A Query Refinement Mechanism for Mobile Conversational Search in Smart Environments

Beibei Hu
MAS Laboratory, Ecole Centrale Paris
92295 Chtenay-Malabry Cedex, France
beibei.hu@ecp.fr

Marie-Aude Aufaure
MAS Laboratory, Ecole Centrale Paris
92295 Chtenay-Malabry Cedex, France
marie-aude.aufaure@ecp.fr

ABSTRACT

A key challenge for dialogue systems in smart environments is to provide the most appropriate answer adapted to the user's context-dependent preferences. Most of the current conversational search is inefficient for locating the target choices when user preferences depend on multiple attributes or criteria. In this paper, we propose an architecture which incorporates a context-dependent preference model for representing weighted interests within utility functions, and a query refinement mechanism that can incrementally adapt the recommended items to the current information needs according to user's critiques. Our preliminary evaluation results based on a scenario of interactive search demonstrate that the query refinement mechanism supported by our architecture can enhance the accuracy of mobile search.

Author Keywords

preference modeling; context-awareness; query refinement.

INTRODUCTION

Recent trends in Spoken Dialogue Systems (SDS) are towards personalized and interactive search in smart environments, which can recommend items that may be of interest to mobile users (e.g., tourists) given their target topics, such as hotels and transportation schedules. One of the main challenges is to exploit preferential information and adapt the answers to user's context over interaction cycles. Two crucial subtasks to this end are: (i) modeling user's context-depended preferences by considering not only hard constrains (e.g., the price should not be more expensive than 50 euros), but also soft constrains (e.g., I prefer French food, otherwise Italian food is also fine); and (ii) improving the accuracy of conversational search according to user's critiques made on the current recommended items.

Modeling user preferences plays a major role in the design of adaptive recommendation systems which provide information or services personalized for user's needs. Typically, user preferences, such as "prefers A over B", could be soft constraints which are represented by a list of alternatives, and thus enable to achieve the optimal answers regarding to the available objects, even if there are no exact match. To adapt the answers to user's current context in mobile environments, it is necessary for dialogue systems to exploit user preferences in a given contextual situation. Since user preferences can be expressed via item ratings, especially for those recommender systems based on collaborative filtering, much attention has been focused on assessing and modeling the relationship between contextual factors and item ratings [1]. In this paper, we aim to model user preferences by taking into account the contextual information and assess the preference weights given multiple attributes.

Given that users are likely to make their feedback on the recommended items during conversational search, this style of user-system interaction requires the dialogue systems to provide better answers adapted to user's critiques. Therefore, the mechanism of preference adjustment formed in critiques-based recommender systems is critical for improving the accuracy of recommendations, so as to ensure a natural and intelligent interaction. Based on Multi-Attribute Utility Theory (MAUT) [9], a critique generation method has been presented to assist users in making critiques according to their stated and potentially hidden preferences [2]. Relying on the above work, we focus on the design of a mechanism that can adjust user preferences and accordingly refine the queries based on the critiques.

Most of the current search engines and recommender systems perform well if a user has a single search criterion and does not have multiple trade-offs to explore. However, few of them provides efficient solutions for finding the user's target choice relying on multiple service properties and their values. To tackle this challenge, we have been involved in the European PARLANCE project[1], which aims to design and develop mobile, interactive, "hyper-local" search through speech in a number of languages [5]. In the context of PARLANCE, our research aims to realize preference-enabled querying for mobile conversational search. Our contributions in this paper are two-folds: (i) we present an ontological architecture of preference-enable querying for smart dialogue systems, which allows to represent user preferences by taking into account the contextual information; and (ii) we propose a query refinement mechanism which can incrementally adapt the retrieved items to user's context given the critiques.

IUI 2013 Workshop: Interacting with Smart Objects, March 18, 2013, Santa Monica,CA, USA

[1] http://sites.google.com/site/parlanceprojectofficial/

In this paper, we discuss the related work on context awareness, user preference modeling and interaction strategies of dialogue systems. By presenting a motivating scenario, we highlight the requirements that led us to design the architecture. After presenting the overall architecture, we elaborate our preference model and query refinement mechanism. Finally, we demonstrate our preliminary evaluation results and outline the future research.

RELATED WORK

This section overviews some important issues concerned with the design and implementation of context-aware dialogue systems in smart environments.

Context awareness. In the field of Ambient Intelligence (AmI), context-awareness is exploited as a technique for developing applications which are flexible, adaptable, and capable of acting autonomously on behalf of users. The most acknowledged definition of context provided by [4] is: "any information that can be used to characterize the situation of an entity (...) relevant to the interaction between a user and an application, including the user and application themselves". As can be observed from this definition, besides the external context such as location and temporal context, it is necessary to take into account the internal context, include current user state, inferences on user behavior and long-term user properties (such as preferences in interaction style) that are relevant to the interaction between a user and a system [15]. Ontology-based context modeling has advantages in terms of sharing a common understanding of the structure of context information among users, devices as well as services, and reusing the domain knowledge, and also describing contexts at a semantic level.

User preference modeling. Given that the user can be considered to be part of the contextual information, it is essential to employ user models to differentiate user types and adapt the performance of dialogue systems accordingly [11]. The General User Modeling Ontology (GUMO) provides a uniform interpretation of distributed user profiles in intelligent semantic web enriched environments [6]. From the quantitative perspective, preferences are rating and defined as a function μ that captures the satisfaction or appealingness of an item $i \in I$ to user $\mu \in U$ within a scale of numerical values [13], usually the real interval [1, 1], i.e., $\mu : U \times I \rightarrow [-1, 1]$. On the other hand, preferences are also viewed as qualitative descriptions of a set of properties that specify user interests, which can be added to the queries as constraints [8]. A preference model based on MAUT proposed in [3] is represented as: a pair $(\{V_1, ..., V_n\}, \{w_1, , w_n\})$, where V_i is the value function for each attribute A_i, and w_i is the relative importance of A_i. The utility of each product $(\langle a_1, a_2, ..., a_n\rangle)$ can be hence calculated as:

$$U(\langle a_1, a_2, ..., a_n\rangle) = \sum_{i=1}^{n} w_i V_i(a_i)$$

Based on the above methods, we are targeted at assessing the preference weights within utility functions and updating the weight values in the user profile.

Interaction strategies of dialogue systems. Dialogue systems in smart environments consist of a series of interconnected computing and sensing devices which surround the user pervasively in his environment and are invisible to him, providing a service that is dynamically adapted to the interaction context [11]. Based on user feedback, dialogue systems conduct preference adjustments in order to modify actual user behavior [10]. For example, a mixed interaction strategy is explored by a context-aware dialogue system named SmartKom [16], so that both the user and the system can take the initiative in the dialogue. In addition, critiquing-based recommendations [12], which elicit critiques that users made on the recommended items, allow users to adjust their preferences during the user-system interaction. However, regardless of the importance of user preference adjustments, it is not a trivial task, particularly in the dynamic environment and mobile applications. This paper adopts the concepts of the above characteristics, and designs a query refinement mechanism, aiming at incrementally enhancing the accuracy of conversational search.

MOTIVATING SCENARIO AND REQUIREMENTS

Consider "Bob Ferris", who is a male student at Ecole Centrale Paris. He is looking for a restaurant that is good for having his birthday dinner. An example of interaction between Bob (U) and his dialogue system (S) can be shown as follow:

(1) U: Where can I have my birthday dinner with 5 to 6 friends tonight?
(2) S: Well, I suggest the medium size restaurants near to your place. I have found some French, Chinese and also Italian restaurants. Which cuisine do you prefer?
(3) U: I prefer French food, otherwise Italian food is also fine.
(4) S: I have found 10 French restaurants. The closest one is "Le Bon Plat", next to the Secaux station. The price is medium, costing 35 euro per person. Do you like it?
(5) U: Sounds good, but is there any cheaper one?
(6) S: Yes, a cheaper French restaurant "Le Gourmet" costing 25 euro per person is at the Antony center. But you need to take the bus 379 for 3 stops.
(7) U: Oh, no. A bit far, I want to a closer one.
(8) S: In this case, there is no suitable French restaurant. How about an Italian restaurant? "Bel Cibo" is on St. Velpeau, two kilometers far from your place. It costs 28 euro per person.
(9) U: Nice, I will choose this one.

As the above interactions show, a smart dialogue system should have the following features:

1. The systems can select interesting objects for users by considering their context-dependent preferences. For example, since "Bob" has a social activity (i.e., birthday dinner), the system infers that fast food is not his current preference. Moreover, user preferences are interpreted as both hard constraints (e.g., price less than 35 euro) and soft constraints (e.g., preferring French food to Italian food).

2. The systems is able to explore multi-attribute tradeoffs for assisting users in locating their target objects. In case that there is no answer can meet user's all desired attributes

(e.g., cheaper and nearer), the system can provide a list of suboptimal alternatives and further adjust their ranking.

3. The systems should have a refinement mechanism, allowing to improve the accuracy of recommended items according to user's critiques. As the scenario shows, the preferences are adjusted and queries are accordingly refined over interactive dialogue turns.

ARCHITECTURE OF PREFERENCE-ENABLED QUERYING FOR SMART DIALOGUE SYSTEMS

Based on the above derived requirements, we designed an architecture of preference-enabled querying for smart dialogue systems (PEQSDS). As shown in Figure 1. The main components and their interactions are elaborated below:

Ontological knowledge base. An ontology-based knowledge base is constructed to represent the background knowledge. It consists of the geographic knowledge exploited from DBpedia for representing location information and also the domain-specific ontologies, e.g., a tourism ontology that can capture the concepts of point of interests. These background knowledge is exploited by the query refinement mechanism to enrich the user preferences.

Context-dependent preference model. In order to represent preferences in a given contextual situation, go beyond the physical context like current location that can be measured by hardware sensors, we also consider the logical context, such as user's activities. For example, a user typically prefers to drive highways for commuting (*activities & time*), but he wants to drive through country roads to go to a supermarket during the weekend. The relations between those context dimensions are represented by the properties in the Resource Description Framework (RDF) schema[2], such as a user *performs* an activity, so that the system can decide the appropriate information (e.g., navigation information) that is adapted to the user's current context. We rely on RDF as data model to formalize information as it facilitates the integration of multiple data sources and the representation of information in a flexible way.

Query refinement mechanism. The user's requests are firstly processed by the *dialogue management* component relying on the technologies of speech recognition, while designing this component is out of the scope of this paper. After receiving the processed requests, the query refinement mechanism is triggered to generate corresponding queries that encodes both hard and soft preference constraints. The initial query set is further enriched and refined according to the user's critiques until the retrieved items can meet the information needs.

User preference model

Semantic Web vocabularies such as the Friend-Of-A-Friend (FOAF)[3] and the Weighted Interests Vocabulary[4], facilitate usage and integration of user profiles in different applications.

[2] `www.w3.org/TR/rdf-schema/`

[3] `http://xmlns.com/foaf/spec/`

[4] `http://smiy.sourceforge.net/wi/spec/weightedinterests.html`

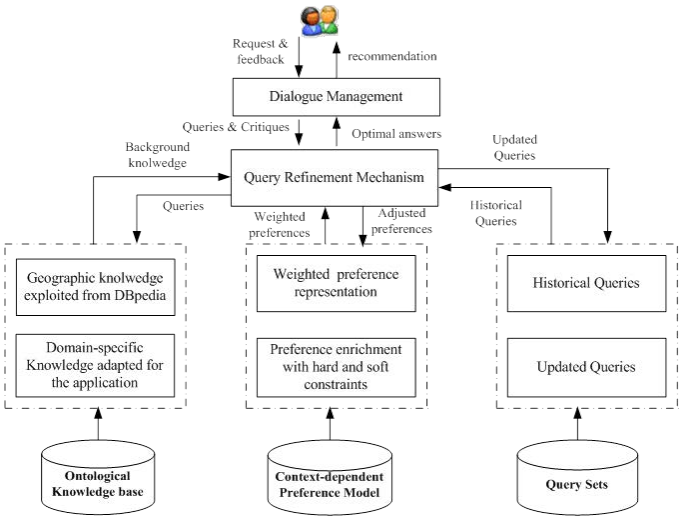

Figure 1. Architecture of PEQSDS

We make use of the standard ontologies and further describe the weighted preferences within utility functions.

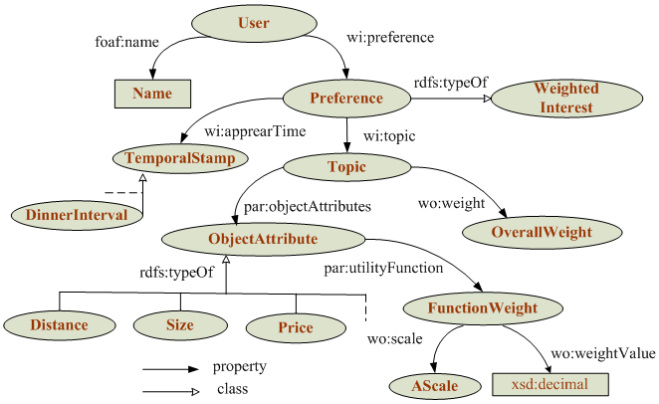

Figure 2. A User Preference Model

As Figure 2 shows, the concept preference has the subject topic and the temporal stamp. The former concept specifies preferred topic in a given domain, while the latter one describes the temporal dynamics of preferences. In particular, the object Topic has two properties: the property *overallWeight* is re-used from the Weighted Interests Vocabulary, while the property *objectAttribute* is defined by us to represent multiple attributes with utility functions. Further, the *objectAttribute* has a property *utilityFunction*, which allows to assign a function weight for each attribute. Thus, a quantitative model of preferences is constructed to compare all alternatives within a topic.

Query Refinement Mechanism

Our algorithm of query refinement mechanism can be described in three main steps: generating initial queries according to user preferences that are enriched by exploiting the knowledge base; adjusting user preferences according to the critiques and further assessing the preference weights based on the MAUT; and refining the queries with respect to the

adjusted preferences until the retrieved items meet user's information needs. In the following, we explain the sub-steps of our algorithm regarding how it generates a list of possible relevant items and uses the top candidate to learn user's criticisms, and how it adjusts the preference weights and further adapts the retrieved items to user preferences.

1. Query generation. To translate a user's request to an appropriate query, the preferences are enriched by exploiting our knowledge base, and an initial query set can be generated by reusing the SPARQL[5] *query generator* we developed for our Relevance Assessment Rule Engine (RARE) [7]. For example, an initial request in our scenario can be expressed as: *selecting near restaurants at medium prices for Bob having his birthday diner*. By taking into account the contextual information, including user's current location, target event and also preferred cuisine recorded in the user profile, a SPARQL query can be generated as:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wi:  <http://purl.org/ontology/wi/core#> .
@prefix par: <http://mas.ecp.fr/parlance/rdf/> .

SELECT ?user ?item
WHERE {
        ?user foaf:name "Bob" .
        ?user par:locatedIn ?curLoc .
        ?user par:isInvolvedIn   ?event  .
        ?event par:socialEvent  par:BirthdayDinner .
        ?item   par:location  ?loc.
        ?loc    par:Nearby ?curLoc .
        ?item   wi:topic  par:Restaurant .
        ?item   par:Size  par:Medium .
        ?item   par:Price  par:Moderately .
        ?item   par:Name ?name .
        {?item  par:foodType par:ItalianFood} UNION
        {?item  par:foodType par:FrenchFood} UNION
        {?item  par:foodType par:ChineseFood} .
        }
            ORDER BY ?name
```

2. Preference adjustment. According to the preference-based organization (Pref-ORG) algorithm proposed in [2], each alternative item will be turned into a tradeoff vector (i.e., critique pattern) comprising a set of (*attribute, tradeoff*) pairs. The tradeoff indicates whether the attribute of an item is improved or compromised compared to the same attribute of the top candidate. Enlighten by the above critique generation approach based on the *Apriori* algorithm, we explore the user's stated critiques not only to adjust the preference order, but also to determine the tradeoffs between attributes, so that the user preferences about the relative importance of each attribute can be reflected. For example, in dialogue turn (3), a preference order regarding the cuisine is adjusted as: $Preference_{foodType} = French \succ Italian$; and in turn (5), the price constraint is modified as: $Preference_{price} = price < Medium$; and also in turn (8), the relative importance of attributes are adjusted as: $priceRange_{weight} \succ distance_{weight} \succ foodType_{weight}$.

3. Query refinement. According to the adjusted preferences, the hard as well as soft preference constraints encoded in the queries are refined. The query with negation expressions can be encoded to filter out irrelevant results. For example, in dialogue turn (3), the expression $MINUS\{?cuisine = res : ChineseFood\}$ is to meet to the constraint of cuisine preference, and in turn (5) the expression $FILTER(?price < 35)$ is to meet the constraint of price preference. We also implemented our algorithm of *preference weights updating* that uses the CONSTRUCT query to infer new triples for updating the weight values. In order to collect the items with higher preference weights, an *ORDER BY DESC (?overallWeight)* clause is specified to sort the results in descending order given the overall weight values.

IMPLEMENTATION

This section explains how the preference weights can be assessed and further updated in the user profile.

Weight assessment for user preferences

We present the user preferences over all preferable attributes within utility functions relying on the MAUT. MAUT uses utility functions to convert numerical attribute scales to utility unit scales, thus allowing direct comparison of diverse measures. Applying this approach involves the following steps, which are described below: 1) normalizing attribute scores in terms of each attribute's Best-to-Worst range, and defining the utility function of the ith attribute ($U_i(x_i)$); 2) specifying the tradeoffs between attributes to reflect user preferences about the relative importance of each attribute and defining k_i as the weight of the ith attribute; and 3) for a consequence set that has values $x_1, x_2, ..., x_m$ on the attributes of m objectives, its overall utility is computed as: $U(x_1, x_2, ..., x_m) = k_1U_1(x_1) + k_2U_2(x_2) + ... + k_mU_m(x_m) = \sum_{i=1}^{m} k_iU_i(x_i)$, where $(k_1 + k_2 + ... + k_m = 1)$, and $0 \leq U_i(x_i) \leq 1$, and $0 \leq U(x_1, x_2, ..., x_m) \leq 1$.

We illustrate how the preference weights can be assessed by applying the above approach in our scenario. Regarding to the three alternatives and their attributes selected in a specific interaction cycle, i.e., a French restaurant named "Le Bon-Plat" (35 euro per person; 1 kilometer distance), a French restaurant named "Le Gourmet" (25 euro per person; 3 kilometer distance) , and an Italian one "Bel Cibo" (28 euro per person; 2 kilometer distance), we firstly set the best value (1) and the worst value (0) by comparing their property values as: $U_{price}(BonPlat) = U_{price}(35) = 0, U_{price}(Gourmet) = U_{price}(25) = 1; U_{distance}(Gourmet) = U_{distance}(3) = 0, U_{distance}(BonPlat) = U_{distance}(1) = 1$. Then, according to the following formula: $U(x) = \frac{x - x_i^-}{x_i^+ - x_i^-}$, where x_i^- denotes the worst value of X_i, and x_i^+ denotes the best value of X_i, utility functions can be defined as: $U_{price}(BelCibo) = 0.6, U_{distance}(BelCibo) = 0.5$. Further, the ratio of the weights is specified according to user's critiques, e.g., $K_{distance} = 2/3K_{price}$. Finally, the overall weight can be assessed as: $U(Bon) = 3/5 \times 0 + 2/5 \times 1 = 0.4; U(Gourmet) = 3/5 \times 1 + 2/5 \times 0 = 0.6; U(Cibo) = 3/5 \times 3/5 + 2/5 \times 1/2 = 0.56$. It can be seen that in this

[5] **www.w3.org/TR/rdf-sparql-query/**

interaction cycle "Le Gourmet" with the highest weight can be recommended.

SPIN for calculations

We defined SPARQL rules in order to calculate the overall weights and also update the values in the preference model. The SPARQL Inferencing Notation (SPIN)[6] provides the syntax to attach SPARQL queries to resources in an RDF-compliant way using RDF properties spin:rule and spin:constraint. The spin:rule property accepts SPARQL CONSTRUCT queries as value and can be used to infer new triples on the basis of the statements in the query's WHERE clause [14]. SPIN adoption is supported by tools as TopBraid composer[7]. SPIN allows us to define a function: *calOverall-Weight (value, functionWeight, overallWeight):float*, for calculating the preference weight of a given item. As Figure 3 shows, the CONSTRUCT clause infers new triples that represent the updated value of the preference weight, and the LET clause specifies how the value is computed by assigning the ratio of weight.

```
CONSTRUCT { ?this  wi:overall_weight  ?ow .
            ?ow   wo:weight_value   ?wv  .
          }

WHERE { ?p  foaf:name "Bob" .
        ?p    wi:preference  ?topic .
        ?topic wi:topic    ?this .
        ?this wo:weight  ?w .
        ?w    a  wo:FunctionWeight ;
              wo:priceWeight  ?pw .
              wo:distanceWeight ?dw .
Let  (?wv := 0.6 * ?pw + 0.4 * ?dw ) .
      }
```

Figure 3. A SPIN Example

PRELIMINARY EVALUATION

We preliminarily evaluated our architecture PEQSDS, in particular the query refinement mechanism based on the data constructed from our scenario.

Dataset construction. To collect a list of items that the user "Bob" may be interested in, we exploited a web search engine, namely Paris Digest[8], which is acted as a city guide for tourism and can recommend points of interests such as restaurants and hotels. By specifying the constraints of location, price, date and cuisine, 22 restaurants were recommended by the search engine. In our evaluation setting, those items were used for re-ranking relying on the query refinement mechanism. We also established a preference model that can specify how much a user is interested into a certain topic. The profile snippet shown in Figure 4 expresses that the user "Bob" is interested into an restaurant named "Bon". For this interest, the utility functions of price and distance are 1.0 and 0.0, separately; and the preference weight is 0.33.

Evaluation measures. We measured the precision of retrieved results in each interaction cycle, in order to assess how likely user's target choice could have been located in the

[6] **http://spinrdf.org/**

[7] **http://www.topquadrant.com**

[8] **http://www.parisdigest.com**

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix wi:   <http://purl.org/ontology/wi/core#> .
@prefix wo:  <http://purl.org/ontology/wo/core#> .
@prefix par: <http://mas.ecp.fr/parlance/rdf> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://www.ecp.fr/people/studentBob>
  a foaf:Person ;
  foaf:name "Bob Ferris" ;
  wi:preference  [
      a  wi:WeightedInterest  ;
      wi:topic (
            <http://example.org/Restaurant_bon>)
  ];

       <http://example.org/Restaurant_bon>
       par:ObjectAttribute   [
            par:UtilityFunction   [
                  a   wo:Weight ;
                  par:priceFunctionWeight  1.0  ;
                  par:distanceFunctionWeight  0.0 ;
                        ] ;
                  ];
      wi:overall_weight  [
           a   wo:Weight ;
           wo:weight_value  0.33 ;
           wo:scale ex:AScale  ;
          dcterms:modified "2012-12-04T10:01:00+01:00"^^xsd:dateTime
                        ] .
```

Figure 4. A Part of Preference Model in the Scenario

recommended items once the query has been refined. The precision denotes the ratio of correctly retrieved items over all retrieved items and can be defined as the following formula: $Precision = \frac{|\{retrieved\ items\} \cap \{user\ target\ items\}|}{|\{retrieved\ items\}|}$. Further, the interaction effort reduction, which is used to assess how effectively the system could potential reduce users' objective effort in locating their target choice [2], is defined as: $EffortReduction = \frac{1}{NumUsers}(\sum_{i=1}^{NumUsers} \frac{actualCycle - targetCycle}{actualCycle})$, where $actualCycle$ denotes the number of cycles a user actually experienced and $targetCycle$ denotes the number of cycle until user's target choice first appeared in the recommended items. In addition, a baseline is defined as: choosing the candidates by specifying a single search criterion (i.e., choosing the restaurants for the user within 2 kilometers). We compared the results of the refined queries to those of the baseline.

Results. In the beginning of the experiment, among a set of n potential interesting items ($n = 22$), after refining the preference order on the cuisine, two items were filtered out and the remained 20 items were sorted by the distance. Then, the top candidate was used to collect user's critiques (e.g., cheaper). In the second interaction cycle, 3 alternatives were retrieved after modifying the preference constraint on the price range. Accordingly, the precision is enhanced to 0.33, compared to the precision achieved by the baseline ($Precision_{baslineline} = 0.25$). By executing SPARQL rules to assess the preference weight, the overall utility was recalculated for each alternative, and the corresponding queries were executed to rank the alternatives by the updated weight values. In the third cycle, the target choice was ranked as the second candidate. This result requests another interaction cycle to re-compute the preference weights by adjusting the

ratio of the weight. Finally, the target choice appears in the fourth cycle ($EffortReduction = 0.25$). The above preliminary results show that our approach can satisfy multiple search criteria and enable end-users to more efficiently target their best choices. However, the limitation is that the target choice was not ranked as the top candidate after re-computing the tradeoff values between attributes. In the next step, our approach should be improved by computing the preference weights as accuracy as user's critiques.

CONCLUSIONS AND FUTURE WORK

In this paper, we presented an ontological architecture of preference-enabled querying for smart dialogue systems (PEQSDS). Our architecture is able to enrich user preferences by exploiting an ontological knowledge base and represent the weighted interests within utility functions relying on a preference model. Also, it provides a query refinement mechanism for adapting the retrieved results to user's context. Contrast to the most of existing approaches that are too restricted to a single search criterion, our approach allows for selecting relevant objects by considering both hard and soft preference constraints. Moreover, it is able to adjust user preferences and further refine the queries by learning the user's critiques. Our preliminary evaluation based on a scenario of mobile conversational search shows that the query refinement mechanism offered by PEQSDS can incrementally improve the accuracy of recommendations.

We currently investigate the enhancements to our approach by applying critique generation and association rule mining techniques. We will show how the tradeoffs between desirable attributes can be determined according to the critiques. Relying on our ontological knowledge base, we will further demonstrate how our system is adaptable for dynamic contextual situations. We will also implement SPARQL rules and improve our algorithm to enhance the ability of computing the preference weights. Our future work mainly focuses on conducting evaluation studies based on large and realistic datasets to show the feasibility of our approach in a pervasive environment. We plan to exploit user's profile information and the search history provided by YAHOO! Local, and further evaluate the system behavior in terms of ranked-biased precision and also the interaction effort reduction.

ACKNOWLEDGMENTS

This work has been supported by the European FP7 PARLANCE project. We gratefully acknowledge valuable comments from Dr. Pearl Pu.

REFERENCES

1. Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing 16*, 5 (2012), 507–526.
2. Chen, L., and Pu, P. Experiments on the preference-based organization interface in recommender systems. *ACM Transactions on Computer-Human Interaction (TOCHI) 17*, 1 (2010), 5:1–5:33.
3. Chen, L., and Pu, P. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction 22* (2012), 125–150.
4. Dey, A. K. Understanding and using context. *Personal Ubiquitous Computing 5*, 1 (2001), 4–7.
5. Hastie, H., Lemon, O., and Dethlefs, N. Incremental spoken dialogue systems: Tools and data. *SDCTD 2012* (2012), 15–16.
6. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. Gumo–the general user model ontology. *User modeling 2005 3538* (2005), 428–432.
7. Hu, B. *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2011.
8. Karger, P., Olmedilla, D., Abel, F., Herder, E., and Siberski, W. What do you prefer? using preferences to enhance learning technology. *Learning Technologies, IEEE Transactions on 1*, 1 (2008), 20–33.
9. Keeney, R., and Raiffa, H. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, 1993.
10. Kuo, M., Chen, L., and Liang, C. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications 36*, 2 (2009), 3543–3554.
11. López-Cózar, R., and Callejas, Z. Multimodal dialogue for ambient intelligence and smart environments. *Handbook of ambient intelligence and smart environments* (2010), 559–579.
12. McCarthy, K., Reilly, J., McGinty, L., and Smyth, B. Experiments in dynamic critiquing. In *Proceedings of the 10th international conference on Intelligent user interfaces*, ACM (2005), 175–182.
13. Polo, L., Mínguez, I., Berrueta, D., Ruiz, C., and Gómez, J. User preferences in the web of data. *Semantic Web*, 0 (2012), 1–9.
14. Spohr, D., Cimiano, P., McCrae, J., and O'Riain, S. Using spin to formalise accounting regulations on the semantic web. In *International Workshop on Finance and Economics on the Semantic Web (FEOSW 2012)* (2012), 1–15.
15. Streefkerk, J., van Esch-Bussemakers, M., and Neerincx, M. Designing personal attentive user interfaces in the mobile public safety domain. *Computers in Human Behavior 22*, 4 (2006), 749–770.
16. Wahlster, W. *SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies)*. Springer, 2006.