

PQMPMS: A Preference-enabled Querying Mechanism for Personalized Mobile Search

Beibei Hu, Yves Vanrompay, Marie-Aude Aufaure

► **To cite this version:**

Beibei Hu, Yves Vanrompay, Marie-Aude Aufaure. PQMPMS: A Preference-enabled Querying Mechanism for Personalized Mobile Search. Web Reasoning and Rules, RR 2013, LNCS 7994, Jul 2013, MANNHEIM, Germany. pp.235–240, 2013. <hal-00831400>

HAL Id: hal-00831400

<https://hal-ecp.archives-ouvertes.fr/hal-00831400>

Submitted on 6 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PQMPMS: A Preference-enabled Querying Mechanism for Personalized Mobile Search

Beibei Hu, Yves Vanrompay, and Marie-Aude Aufaure

MAS Laboratory, Ecole Centrale Paris, France

`{beibei.hu,yves.vanrompay,Marie-Aude.Aufaure}@ecp.fr`

Abstract. A key challenge for personalized mobile search is to tailor the answers to the specific user by considering her contextual situation. To adapt the retrieved items to user's context, this paper presents a preference-enabled querying mechanism for personalized mobile search. By exploiting the user's dialogue history, we infer the weighted user preferences and interests. To further compute personalized answers, we aim to continuously collect the ratings given by the user's friends regarding relevant topics from stream-based data sources such as Twitter. An experiment shows that our approach allows to compute the most relevant answers, providing an increased quality of search experience for the user.

Keywords: preference queries, preference inference, stream reasoning.

1 Introduction

Advances in Spoken Dialogue Systems (SDS) have allowed the provision of context-dependent information and services for mobile users in ubiquitous environments. For example, a tourist can ask her smart phone to recommend the points of interest (POIs) that may be of relevance for her, such as suggesting which museums to visit or which restaurants to go to. One of the critical issues is how to provide desirable answers to the user given her contextual situation by exploiting preferential information. Two crucial subtasks to this end are: 1) modeling the user's preferences by exploiting background knowledge; and 2) reasoning and query answering by the continuous processing of social semantic data. Most of the current search engines and SDS focus on delivering information according to a single search criterion specified by the user, but few of them provide efficient solutions for proactively providing relevant answers given user's preferences and the social context.

Let us consider a motivating scenario, *a tourist Maria asks her dialogue system for recommending the best restaurant near the Eiffel Tower in Paris*. First of all, the system infers her preferences given multiple attributes from the dialogue history, e.g., she prefers French restaurants at medium prices. Then, according to her current location, the system can select a list of nearby restaurants that might be interested by Maria. In order to determine the most relevant answers, the system also utilizes the opinions (e.g., positive) of Maria's friends collected from the social media streams to influence the ranking. From this scenario, we can

learn that the system is requested to support advanced inferences and reasoning over background knowledge and also streams of changing information.

To tackle the above challenge, we have been involved in the European PARLANCE project¹, which aims to design and develop mobile, interactive, “hyper-local” search through speech. We focus on presenting a preference-enabled querying mechanism, which is able to infer the weighted interests by exploiting the dialogue history and further offer relevant answers by considering the ratings of the user’s followers as well as following with regard to POIs. Our contributions in this paper are two-folds: 1) we present an architecture for preference-enabled querying which can infer the weighted interests from the dialogue history and encode those interests into the queries; 2) we propose a mechanism which can determine the most relevant answers by exploring both persistent knowledge base and streaming data from social networking websites such as Twitter.

In the rest of this technical communication we overview the state of the art of preferences queries and stream reasoning. Then we present the overall architecture and explain the interactions of different components. Finally, we demonstrate our preliminary experiments and outline the future research.

2 Related Work

Preference queries. From the quantitative perspective, preferences are ratings 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, usually the real interval $[1, 1]$, i.e., $\mu : U \times I \rightarrow [-1, 1]$ [7]. On the other hand, preferences can also be expressed as qualitative descriptions of a set of properties that specify user interests, which can be added to the queries as constraints [5]. To query the Semantic Web with preferences, the PREFERRING clause has been provided in the SPARQL which states preferences among values of variables [8]. However, this clause cannot encode the social context that may impact on the ranking. In contrast to the extension of the standard query language, we focus on modeling preferences as a type of weighted interest according to user’s dialogue history.

Continuous querying social semantic data. Social data as feeds and microblogs are adapted to publish information in real-time stream through social networking Web site [1]. This trend towards the interlinking of Social Web with semantics using vocabularies such as Semantically-Interlinked Online Communities (SIOC)², Friend-of-a-Friend (FOAF)³ and Simple Knowledge Organization System (SKOS)⁴. Much research focuses on the Linked Stream Data [6] and Stream Reasoning⁵ with the aim to realize transient streams and continuous reasoning tasks. C-SPARQL as an extension of SPARQL allows existing reasoning mechanisms to be further extended in order to support continuous reasoning over

¹ <http://sites.google.com/site/parlanceprojectofficial/>

² rdfs.org/sioc/spec/

³ xmlns.com/foaf/0.1/

⁴ www.w3.org/2004/02/skos/

⁵ streamreasoning.org/

data streams and rich background knowledge [2]. A mobile search method proposed in [3] aims to compute novel and effective features for ranking fresh URIs by using Twitter data. Also, a stream reasoner named BOTTARI [1] has been developed to continuously analyse twitters to understand how the social users collectively perceive the POIs in a given area. We aim to investigate how to use the ratings from the social users to influence the relevance of retrieved items.

3 Architecture and Components

We designed a Preference-enabled Querying Mechanism for Personalized Mobile Search (PQMPMS), which supports providing personalized answers tailored to user preferences by taking into account the social context. The main components and their interactions are elaborated below (and shown in Figure 1):

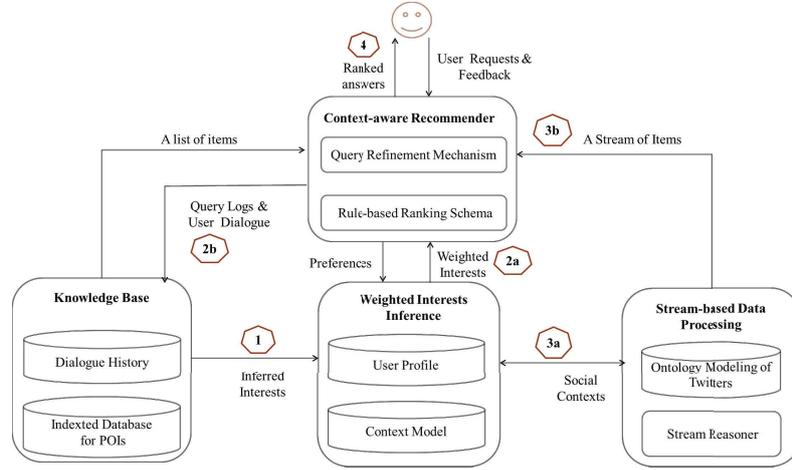


Fig. 1. PQMPMS Architecture

Context-aware recommender. This component fulfills the core functionality of recommending the desirable items in two steps. First, it generates the preference-enabled queries and further refines them according to user’s feedback relying on our query refinement mechanism [4]. Referring to our motivating scenario, Maria’s request is translated to a query, which encodes her preferred cuisine recorded in the user profile and the current location represented by the context model. The generated query for *selecting the French restaurants near to the Eiffel Tower within 5 kilometers and ranking the results by the distance* is shown in Figure 2. Second, the relevance of the retrieved items is assessed based on the ranking schema. To represent the relevance of a given item for a specific user, the concept item has the rating object with its properties *isForUser* and *hasRating*. The ranking of item i for user j is represented as:

$$R_{i(j)} = \alpha \frac{1}{n} \sum_{z=1}^n w_z + \beta(\lambda F + (1 - \lambda)P) \quad (0 < \lambda < 1, \alpha + \beta = 1) \quad (1)$$

```

SELECT ?subject ?sub ?label ?subgeo WHERE {
db: Eiffel_Tower geo:geometry ?eiffelgeo.
?subject geo:geometry ?subgeo .
?subject rdfs:label ?label.
?subject par:foodType par:FrenchFood .
?subject dcterms:subject ?sub.
FILTER(bif:st_intersects(?subgeo,?eiffelgeo, 5) &&
?sub = dbc:Restaurants_in_Paris &&
lang(?label) = "en"
).
} ORDER By DESC (?subgeo)

```

Fig. 2. A Preference-enabled Query

where w_z represents the inferred interest scores, F is the normalized number of followers/following, and P is the percentage of positive opinions. Once a new relevance value is computed as formula (1) by considering the weighted interests and the ratings from user's friends, the SPARQL CONSTRUCT queries are used as rules to infer new triples for updating the relevance of a given item.

Knowledge base. It is composed of the dialogue history and an indexed database containing POIs. The dialogue history consists of an abstract representation (so called dialogue act units) of all past dialogues between the user and the system. The second part of the knowledge base is the indexed database with POIs. The preference-enabled query, being the query extended with the interests of the user obtained from the user profile component, will be executed to search in the database for the set of POIs that are in correspondence with the query.

Weighted interests inference. It consists of a user profile and a context model. The user profile contains modular ontologies that are annotated with scores representing the interests of a user on different levels. The history will be used to infer the interests of the user on several levels. First, the interest of the user on the *attribute type* level can be inferred. For example, if a user in his past interactions with the system has asked a lot about the food type of a restaurant but not about the price, we infer that the food type attribute is of more interest to the user than the price. Second, we infer the interests of the user on the *attribute value* level, meaning that for one specific attribute type like food type, the user prefers Italian or Chinese food above Mexican. Third, the interests on these 2 levels are combined to get the interest of a user at the *instance* level, i.e. the interest in specific restaurants. It should be noted that these interests evolve in time. When the user has interacted with the system, the dialogues are analyzed and the weights in the model are updated accordingly. Each ontology module corresponds to one domain of interest, such as restaurants or transportation. The recommender component calls the appropriate modular ontology, depending on the subject of the dialogue (e.g. searching for restaurants), and analyzes the associated scores to build the preference-enabled queries. The context model consists of characteristics of the user that can be relevant for recommendation, and of the current location of the user. Also the social context of the user is included in FOAF format, representing a.o. the followers of the user on Twitter.

Stream-based data processing. To model the Twitter data, we adapt the SIOC vocabulary which defines `TwitterUser` as a special case of `UserAccount`. The model defines the notion of POI as `NamedPlace`, which is enriched with a categorization and count of positive/negative/neutral ratings. It defines the object property `talksAbout` and its sub-properties for positive, negative and neutral opinions, so that the opinions of a tweet about POIs can be represented. We utilize C-SPARQL queries to continuously analyze the Twitter data for collecting the items rated by user’s followers/following. A query example for creating a stream of the restaurants that are positively talked by the followers of Maria in the last week is shown as:

```
REGISTER STREAM RestaurantsMariaFollowersLike COMPUTE EVERY 5h AS
CONSTRUCT { ?friend sr:talksAboutPositively ?poi}
From STREAM <?http://www.streamreasoning.org/stream#> [RANGE 7d Step 5h]
WHERE {
?poi a sr:NamedPlace ;
      dterms:subject dbc:Restaurants_in_Paris .
?friend sr:following ?user .
        sr:posts ?tweet .
?user foaf:name "Maria" .
?tweet sr:talksAboutPositively ?poi .
}
```

Our architecture works as follows. First of all, the weighted interests are inferred from user’s dialogue history and are modeled in the user profile (noted as 1 in Fig.1). Then, the user’s requests are translated to the corresponding SPARQL queries by considering user preferences and contextual information (noted as 2a). Accordingly, a set of items defined as I_q is retrieved by sending preference-enabled queries to our knowledge base to get POIs (2b). Next, the component of stream-based data processing exploits the social context (3a) to provide a stream of the items (defined as I_s) that have been positively rated by the user’s friends by executing the C-SPARQL queries (3b). Then, the intersection of I_q and I_s is computed and the relevance rating of those items needs to be obtained. The final relevance rating of a certain item is determined by the weighted interests and opinions of user’s friends (4). Finally, the top item with the highest relevance rating is recommended to the user.

4 Evaluation

We evaluated the effectiveness of PQMPMS based on our motivating scenario. Firstly, we queried DBpedia⁶ to search all the restaurants that are near to the location of the Maria (e.g., *the Eiffel Tower*), and thus collected a list of twenty-two items. According to the inferred interests, i.e., preferring nearby French restaurants, three candidates were retrieved by executing a preference-enabled query. Next, we constructed a synthetic set of tweets for these restaurants (conforming to reality by looking at ratings for them on the Web), which specifies the number of followers and the percentage that talks positively about these restaurants. The final ratings (calculated by formula 1 in Section 3) as shown in Table 1, is the weighted sum of the normalized number of followers and the

⁶ <http://dbpedia.org/sparql>

percentage of positive comments. As can be seen, there is e.g. a big difference in both the number of followers and the percentage of positives between restaurant “Taillevent” and “Ledoyen”, resulting in a significantly different final rating. Thus, the ranked answers computed by considering user’s preferences and also the ratings of followers can better meet user’s needs.

Table 1. The Relevance Rating of Three Restaurant Items

Subject	Followers	Positive	Rating
db: Le_Boeuf_sur_le_Toit_(cabaret)	274	87	0.94
db: Ledoyen	127	58	0.52
db: Taillevent_(restaurant)	48	68	0.43

5 Conclusions and Future Work

We have presented our preference-enabled querying mechanism, which can provide personalized answers by inferring user preferences from dialogue history and explore the ratings from the social users given POIs to determine the ranking of answers. Our future work mainly focuses on utilizing dialogue act unites provided by YAHOO! Local to infer user’s preferences and also computing personalized answers by continuous analysis of social media streams. We will conduct evaluation studies based on steaming data and measure the rank-biased precision achieved by our system.

References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental reasoning on streams and rich background knowledge. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) *ESWC 2010, Part I. LNCS*, vol. 6088, pp. 1–15. Springer, Heidelberg (2010)
2. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Querying rdf streams with c-sparql. *SIGMOD Rec.* 39(1), 20–26 (2010)
3. Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., Zha, H.: Time is of the essence: improving recency ranking using twitter data. In: *Proceedings of the 19th International Conference on World Wide Web*, pp. 331–340. ACM (2010)
4. Hu, B., Aufaure, M.-A.: A query refinement mechanism for mobile conversational search in smart environments. In: *Proceedings of the IUI 2013 on Second Workshop on Interacting with Smart Objects*, pp. 1–6 (2013)
5. Karger, P., Olmedilla, D., Abel, F., Herder, E., Siberski, W.: What do you prefer? Using preferences to enhance learning technology. *IEEE Transactions on Learning Technologies* 1(1), 20–33 (2008)
6. Le-Phuoc, D., Xavier Parreira, J., Hauswirth, M.: Linked stream data processing. In: Eiter, T., Krennwallner, T. (eds.) *Reasoning Web 2012. LNCS*, vol. 7487, pp. 245–289. Springer, Heidelberg (2012)
7. Polo, L., Mínguez, I., Berrueta, D., Ruiz, C., Gómez, J.: User preferences in the web of data. *Semantic Web*, 1–9 (2012)
8. Siberski, W., Pan, J.Z., Thaden, U.: Querying the semantic web with preferences. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006. LNCS*, vol. 4273, pp. 612–624. Springer, Heidelberg (2006)