# Semantic Hyperlocal Search for Parlance Mobile Spoken Dialogue System

Panos Alexopoulos[1], Marie-Aude Aufaure[2], Nesrine Ben-Mustapha[2],
Hugues Bouchard[3], José Manuel Gomez-Pérez[1], James Henderson[4],
Beibei Hu[2], Joel Lang[4], Peter Mika[3], and Yves Vanrompay[2]

[1] iSOCO Madrid Spain
[2] Ecole Centrale Paris
[3] Yahoo! Research Barcelona
[4] University of Geneva

## 1 Introduction

Current spoken dialogue systems (SDS) for mobile search are mostly domain-specific and make use of static knowledge. Consequently they do not take into account the interests, location and contextual situation of the concrete user. We propose PARLANCE, which is a more dynamic and personalized SDS that incorporates 1) a dynamic knowledge base consisting of modular ontologies that are enriched incrementally with information extracted from the Web; 2) and an evolving user profile. This allows the system to provide answers that are more tailored to the concrete user and to exploit the Web as a source of information, which can improve the quality of experience for the user. The PARLANCE SDS aims to guide the user in his search for information by providing answers that are: (1) Hyperlocal: The current geographical location of the user is taken into account to provide points of interest (POIs) in the neighborhood; (2) Dynamic: New concepts and entities are learned at runtime and included in the appropriate modular ontologies; (3) Personalized: Potential relevant answers adapted to user's queries are selected and ranked according to user preferences. Complementary, a form of social search is performed by looking at interests of similar user in the neighborhood (i.e. collaborative filtering). The central component in the PARLANCE architecture is the Interaction Manager (IM) which probabilistically decides on the most appropriate next answer to be provided to the user. The IM exploits information from the Semantic Web by interacting with the Knowledge Base (KB), the Web Content Analyzer (WCA) and the Local Search (LS) components, which will be detailed in the next section.

## 2 Hyperlocal, Dynamic and Personalized Spoken Dialogues

*Modular Ontologies and User Profile.* The Knowledge Base consists of the Ontology Manager (OM) and User Model (UM) and is informed by the Web Context Analyzer. The OM is responsible for the construction and enrichment of the

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7955, pp. 271–272, 2013.

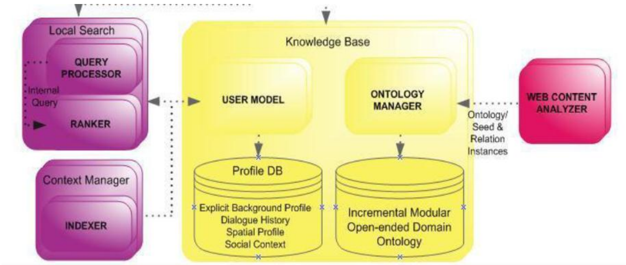

**Fig. 1.** Hyperlocal semantic search components of PARLANCE

modular ontologies in the form of xsd-schemas specifying RDF documents. In addition, we propose a method of ontology enrichment by searching web snippets related to each entity associated with a core ontology module schema, focusing on one attribute at a time.Web snippets are considered as short sentences, which can be parsed to retrieve linguistic patterns by the WCA. These patterns are then ranked according to web-based co-occurrence measures by querying the Web using the extracted linguistic patterns. The User Model contains user interests, social and contextual information that is exploited to provide tailored answers to the user.

*Web Content Analyzer.* The goal of the WCA is to extract information from web content. The extracted information is written to the Knowledge Base. Specifically, the WCA is concerned with extracting relation tuples for relations in the domains of interest. In our approach the target concept (either an entity class or a relation) is defined in terms of a relatively small number of seed instances (seed set), which are then used to extract further instances belonging to the concept. We developed a novel graph-based seed set expansion method, in which candidate instances are ranked according to their distance/similarity to the seed instances in the graph, where distance is measured in terms of truncated random walk hitting times.

*Local Search.* The LS consists of an Indexer, Query Processor and Ranker. The ontology provides information at index time to the Indexer to determine the metadata fields to be applied to the POIs, for example, that the POI is a restaurant, which serves Chinese food and is cheap. The Query Processor determines the geographical scope and personalization and the index is organized into small geographical partitions, populated with data instances and embedded with metadata associated with the Knowledge Base. The Indexer builds the necessary index structures for scalable matching and ranking. The Indexer is an offline component that builds an index from RDF data on disk, which is used at runtime for matching results. Thus the Indexer does not support dynamic updates of the data or the schema but instead rebuilds the index periodically from the data feeds received from the WCA. The Ranker performs runtime ranking based on the internal query received from the Query Processor. The goal of the Ranker is to order the results retrieved from the index according to the relevance of the original query.