Processing natural language queries to disambiguate named entities and extract users' goals: application to e-Tourism

Sanjay Kamath, Lorraine Goeuriot, Marie-Christine Fauvet

Processing natural language queries to disambiguate named entities and extract users' goals: application to e-Tourism

Sanjay Kamath, Lorraine Goeuriot, Marie-Christine Fauvet

LIG (MRIM), University of Grenoble Alpes, CNRS, Grenoble, France

ABSTRACT. *This paper presents a study which is part of a broader project. This latter aims at providing mobile users with context-aware personalised services. The E-tourism Project deals with a variety of queries submitted by a tourist, such as booking a hotel room, getting the weather conditions for the next day, or booking tickets in a museum in the neighbourhood and worth to visit. This paper focuses on the query management and processing. The module described analyses and structures the query by splitting it, identifying the named entities, solving ambiguities... To process the query, the system uses various external knowledge bases and Natural Language Processing tools to understand the named entities and proper context of the query using disambiguation techniques.*

RÉSUMÉ. *Cet article présente une étude qui s'inscrit dans le cadre d'un projet plus large qui porte sur la conception et la réalisation d'un système visant à fournir à des utilisateurs mobiles des services personnalisés, dépendant de leur contexte, et adaptés à leurs besoins. Par exemple, un utilisateur peut vouloir des informations sur la météo du lendemain, ou bien réserver des billets d'entrée à un musée voisin, ou encore réserver une table dans un restaurant italien et obtenir les indications pour s'y rendre en voiture. Dans cet article, nous étudions plus particulièrement les problèmes liés à l'extraction de requête fournie par l'utilisateur des paramètres nécessaires à son traitement, ainsi que du contexte de l'utilisateur. Le système que nous proposons est basé sur le traitement de requêtes exprimées en langage naturel. Afin d'extraire de la requête les entités nommées ce système s'appuie sur des bases de connaissances et des outils de désambigüisation.*

KEYWORDS: *e-Tourism, Query Processing, Context Aware Personalised Services*

MOTS-CLÉS : *e-Tourism, Traitement de Requêtes, Services Personnalisés Dépendants du Contexte*

1. Introduction and Motivating Example

With the development of pervasive computing and mobile technologies, mobile applications are getting more and more attention. Distributed computing systems based on context awareness have been proposed in several domains such as health-care, logistics and tourism. Most of the existing service providers in the tourism domain focus only on a few specific goals, such as booking a restaurant, or searching a hotel. These applications, running on mobile devices, are challenged to locate and deliver the right service to the right person, with the appropriate rendering.

The work reported in this paper is part of a broader project which aims at designing and implementing a framework which provides context-aware personalised services for mobile users according to their needs, profile and context. The issues to be addressed in this project are related to system design, software architecture, distributed and heterogeneous resource access and integration, information retrieval and recommender systems. This paper focuses on the user's queries management and processing.

To illustrate the challenges tackled in this paper, let us consider the following scenario. Alice is an American tourist visiting Paris in France. She picks up her smartphone and issues the query, once connected to our e-Tourism system: *Want to book a table tonight at the closest restaurant to the Eiffel Tower and directions to get there*. The study reported in this paper has the following contributions:

– Recognition and disambiguation of **names entities**: according to Alice's context the value returned by the GPS embedded in her smartphone, she is located in Paris, Eiffel Tower is then recognized as a named entity in Paris (nor in Las Vegas, Nevada, neither in Brisbane, Australia).

– Extraction of user's **goals**: in the example, Alice has two main goals: the first one is *"I want to book a table at the closest restaurant to Eiffel Tower for tonight"*, and the second one *"get directions to reach the restaurant"*.

This paper is organized as follows: we describe the architecture of the whole system in Section 2; while we present in Section 3 some related work. in Section 4 we propose our approach. Section 5 details the implementation of this approach and evaluation of our system. Eventually, in Section 6 we conclude ans sketch some further work.

2. Architecture of the eTourisme System

The study described in this paper is conducted as part of a broader project whose main goal is provide mobile users with services according to their needs (Na-Lumpoon *et al.*, 2013).

We call a goal the task requested by the user (e.g. movie booking, events nearby...). A query might contain one or more goals. The *Context* includes spatial, temporal,

physical, and environmental properties that could be collected by sensors embedded on the devices used to submit the queries. The *Profile* captures users' personal details, preferences and centers of interest.

Figure 1 sketches the overall architecture of this project dedicated to e-tourism (Fauvet *et al.*, 2015). The role of each module and the flow of information are rapidly introduced below.

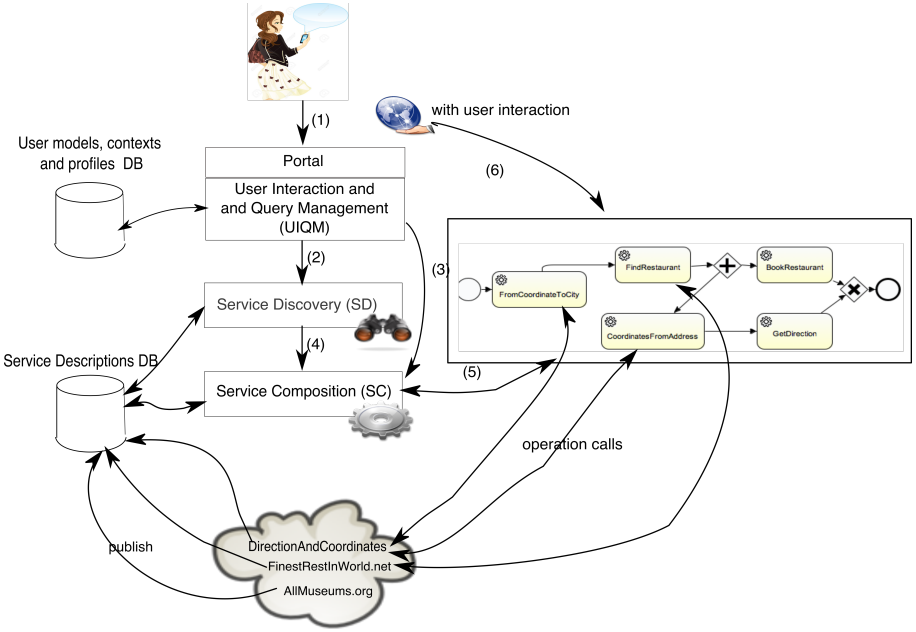

Figure 1. *A context-aware discovering system for mobile users.*

1) **User interaction and query management module**: aims at managing user interactions by handling the queries submitted by users on their mobile device. A user's query and her identification are received in this module in the data flow (1). This module extracts from the query, the information necessary for the choice and composition of the service components. With data flow (2) it sends single goal queries to the Discovery module and with data flow (3) the users' goals to the Composition and execution system. This paper focuses only on this module.

2) **Discovery module**: given the user's query received in the data flow (2), her profile and context, this module is responsible for retrieving services among a repository of service descriptions, that once composed can potentially meet the user's goals expressed in her query. The retrieved services are then sent, in the data flow (4), to the next module. For details see (Caicedo-Castro, 2015; Caicedo-Castro *et al.*, 2015).

3) **Composition and execution system**: eventually this module is in charge of automatically compose and execute services returned by the discovery phase. The automated composition is meant to satisfy users' goals. This module results in a BPM model whose each task refers to a service operation by the data flow (5) (for details see (Na-Lumpoon, 2015)). During the execution of the resulting process, some interactions with the user might be necessary (see data flow (6)).

3. Related Work

3.1. *Ontology based approaches*

Various approaches have been used in creating an ontology or using one for the purpose of tourism. (Tomai *et al.*, 2005) discusses a case study on trip planning using two ontologies to represent the user profile and an ontology for tourism data such as shopping malls, cinemas, activity spots nearby etc. Assumption of the user's location is always the city centre according to the above approach hence the user's device is not constantly looking for GPS signals and User profile related information is collected through a questionnaire to the user.Using these information the context matching algorithm is used to determine the feasibility of the plan in accordance to the time the traveler has.

Building an ontology is expensive and time consuming. The approach proposed by (Faria *et al.*, 2014) is a new approach for automatic ontology population that uses an ontology to automatically generate rules that extract instances from text and classify them in ontology classes.These rules can be generated from ontologies of any domain, making the process domain-independent.

Ontology population and the evaluation for tourism domain corpus is discussed in (Ruiz-Martınez *et al.*, 2011) which proposes a methodology for extracting semantic content from textual web documents to automatically instantiate a domain ontology.Semantic contents are obtained using a framework and considered as instance candidates which undergo disambiguation for semantic ambiguities and then related with their ontology entities.

Although generating an ontology or using an existing one makes it easier to build a system upon because of the markups provided by the ontology which can be accessed by OWL [1], the generation of ontology involves domain expertise and constant updating of the data relationships which is a costly operation and not fully automated.

1. Web Ontology Language, `http://www.w3.org/2001/sw/wiki/OWL`

3.2. *Tourism datasets*

Similarly to DBpedia [2], some efforts are made in building TourPedia (Cresci *et al.*, 2014), which includes point of interests, accommodations, restaurants and attractions using various commercial data sources. However, TourPedia still covers only a part of Europe, it will take long for it to cover world wise.

4. Proposed Approach

As explained earlier in Section 2, we focus here on the Query Management Module which is depicted in Figure 2.

Each module described in the architecture is discussed in details below: Paragraph 4.1 discussed the pre-processing we apply on queries. Then Section 4.2 details how **Named Entities Recognition** is performed. Section 4.3 discussed how we deal with disambiguation of name entities. Eventually, Section 4.4 described how we extract goals from the user's query. At different stages of the query, we use *User Context* and *Profile* which are metadata returned by sensors embedded in the device used by the user to issue her query. External knowledge bases are accessed to help disambiguation of Named Entities.

4.1. *Pre-Processing of the Queries*

The short queries considered in tourism domain consists of various keywords which are of different parts of speech in English. Some scenarios might have conditional keywords in them. The conditional keywords in our corpus of queries are simple keywords like **if, then, else, otherwise, if not**, etc. And in certain cases the queries might have multiple goals in them, these type of queries will have keywords like **and, along with, and also**, etc.

Our proposed approach does the shallow parsing on the queries for searching keywords mentioned above, and splits the queries into separate single queries. The conditional keywords (**and, or**) are used to split the query into individual queries. Part of the speech tagging is used to determine the **verbs** present in the query, which are more likely to be the user's goals.

4.2. *Named Entity Recognition*

Determining the Named Entities from a query or short text is important because the entity can have a great influence over the query intent. There are many tools

2. `http://wiki.dbpedia.org`

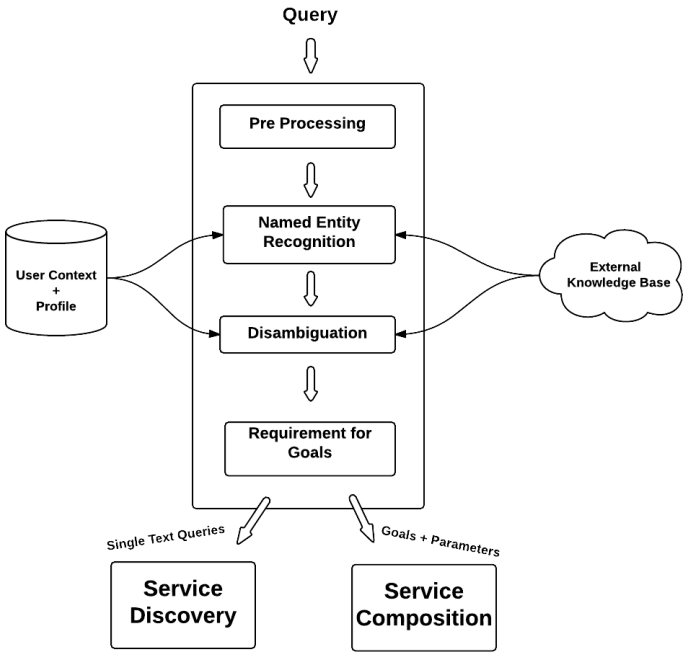

Figure 2. *Architecture of Query Management Module*

which can be used to determine the named entities in the text. Named Entity Recognition (NER [3]) is a well known task of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organisations, locations, expressions of times, quantities, monetary values, percentages, etc. Statistical models are used to detect named entities from text.

Using NER alone is not sufficient specially in Tourims applications. Detection of named entities will lead the system to determine also the kind of activity or goal which a tourist would like to do with that named entity. Hence we need some external knowledge base to clarify the named entity with more details. Our focus is on the queries where the NER results are not sufficient to determine the goal of the query.

Below, we discuss an example case that shows the importance of named entities and how we solve ambiguities between them: *I would like to book a ticket for the grand budapest hotel nearby for tonight.*

Having no insight about the query, a human might interpret it as a hotel booking query in Budapest but, the real context of the query is a movie booking. The phrase

3. `https://opennlp.apache.org/documentation/manual/opennlp.html#tools.namefind.recognition`

"the grand budapest hotel" refers to a movie which was released in 2012. Having a keyword "ticket" and the phrase which is a movie name, refers to a movie ticket booking and has nothing do with a hotel in Budapest.

The NER provides the output "budapest" and "hotel" because the keywords "The" and "Grand" are *article* and *adjective* respectively. This is often the case with movie titles, as they have *verbs*, *adjectives* and various other part of speech tags in them making the NER fails to recognize them.

The problem also exists when there are conflicting named entities in the same domain which refers to two different point of interest, like "Hotel" or "Movie" in our case. Hence we use external knowledge sources or datasets and represent it as a list of results S.

4.3. *Disambiguation*

In the case of an ambiguity, we need methods to rank the potential entities. We use an algorithm for the ranking function R to do so using the user and the entity locations. The main steps of this algorithm are:

```
Inputs: NE = Named Entity of the query
       SRC = User's location obtained using GPS
       Treshold = the maximum distance between the user and POIs
                  Treshold value must be agreed by the user

R(NE, SRC, Treshold):
     K=1;                              \\ one result needed
     Generate S with external data
       for (K=1; K--;) {
        if distance (SRC, location(S)) > Treshold then K++
       }
Outputs:
        Top K(K=1) results from List S.
```

We apply comparing Top K lists (Fagin *et al.*, 2003), as the output obtained is ranked against the location values. We use K=1 because we need the best ranked result from the list S.

4.4. *Requirements for goals*

Each goal has specific parameters required for composition and execution of services, as shown in Table 1. For instance, a movie booking has as parameters the movie name, cinema name, number of people, the city where the cinema is located and day and time of the booking.

From observing classical Web applications such as *booking.com*, *tripadvisor.com*, etc., we simply assume certain mandatory parameters for specific goals which are common in the domain of tourism.

Goals	Parameters
Movie booking	movie name, cinema name, city name, people number, day and time of booking
Hotel booking	hotel name, city name, people number, checking day, checkout day
Museum booking	museum name, city name, people number, visit day
Direction	mode, source, destination

Table 1. *Expected parameters depending on goals.*

In cases where the system has conflicts in filling in the parameters, the interaction module asks the user to input that particular parameter again rather than making wrong assumptions.

We encounter the *synonymy problem* in this phase. Words like 'Book' and 'Reserve' both refer to the same verb. To tackle this problem, we use *Wordnet* [4].

The results from the query processing module is the parameters required for the composition and execution of services.

5. Implementation and Evaluation

The Figure 2 (see page 6) sketches the overall architecture of the *User Interactions and Query Management module*. The role of each sub module is discussed in details below.

The query is obtained from the user using a *Graphical User Interface* which is web application (see `http://lig-membres.imag.fr/etourism/en/`). Our system is still in implementation phase.

5.1. *Pre Processing*

Apache OpenNLP [5] is a Java machine learning toolkit for Natural Language Processing (NLP). It supports the most common NLP tasks. OpenNLP includes rule

4. `https://wordnet.princeton.edu`, Wordnet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members.

5. `https://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html`

based and statistical Named Entity Recognition. There are many other toolkits which are off the shelf named entity recognisers for English text. For processing the queries we use NLP tools such as Tokenizer and POS Tagger [6]. Tokenizer splits the queries into tokens and the POS Tagger appends the tokens with the part of the speech corresponding to that word in the query. This part of the speech value can be used to determine the goals, as goals are always a verb in our domain.

5.2. *Named Entity Recognition and Disambiguation*

Apache OpenNLP toolkit also provides an NER tool to determine named entities. If there are no named entities in the query, it is probably because the tool fails to recognise them. Hence we use knowledge sources listed below to determine named entities.

The online knowledge sources we use are WolframAlpha [7], Results by Google places [8], Foresquare [9], Google search [10]. We use the above data because they represent most of the things related to tourism domain and also they are easy to obtain. Our system can make use of knowledge source or datasets like (Marchetti *et al.*, 2013) for extracting named entities for the places in Europe, but this system is limited to some of the locations in Europe and not yet complete for all the places around the world. Hence, currently we just use the search results for named entities from these sources. The System parses the output from the sources to determine the different named entities using the ranking function as described in the Disambiguation Algorithm.

5.3. *Requirements for goals*

We perform the POS Tagging and disambiguation of the named entities to extract goals, we use Wordnet data to compare the results with goals. We classify the requirements mentioned above using a key for representing each goal. This key is stored along with the set of requirements. For example, "Book" is a key for all bookings-based queries. Whenever a *verb* is encountered in the query, Wordnet is accessed to get the Verbs which are synonyms to the input verb. Data obtained from Wordnet are matched with keys to determine the key of the set of requirements which it matched.

We compare Wordnet synonyms with the keys of the requirement class to understand the goal requested by the user when the user uses synonyms. Once we obtain the goal, we substitute the data present in the query for the parameters we assumed.

6. Parts of the Speech Tagger provided by Apache OPENNLP: `http://opennlp.apache.org`
7. `http://www.wolframalpha.com`
8. `http://developers.google.com/places`
9. `https://foursquare.com/`
10. `http://www.google.com`

Some of the parameters present in the query match the parameters assumed in the system, and some parameters might be missing. The system signifies it as missing parameter and passes it to the service composition module. This module does not invoke an interaction with the user for the missing parameters, because the service operations might need few parameters and some of the parameters obtained from the user might not be used. Hence invoking the user for appropriate missing parameter is solely dependant on service execution.

5.4. *Results*

The results generated from this module are: 1) Single Text queries or Single Goal queries which are the inputs for the Service Discovery module. 2) Goals + Parameters which are the inputs for the Service Composition module of the whole eTourism System.

6. Conclusion and Future Work

It is hard to evaluate this approach yet because the output of this system is a part of the system of eTourism implementation, and eTourism project has a combination of many other modules which interact with each other to produce a collective result.Hence a dataset has to be generated which consists of user profiles and corpus of queries. One of the main concerns about the above approach is privacy. Some users might hide their locations and make the system in worst case scenario.We assume that the user allows the system to access his location provided by the GPS Module. In certain cases, the result from a service is passed as the input to another service, the execution of the system shall wait for a parameter which has to be passed on to it. These type of issues deal with the causality of events which are discussed in detail by (Allen, 1984) and there has to be some work done on this which relates to service composition and execution.

Proactiveness of a system mainly depends on the data of the user and use of algorithms which can predict the user's actions in prior, hence there is a huge scope for making the system proactive by providing context aware recommendations.Proactiveness can be more accurate when there is more profile information,history,social media profiles etc.

Also, the temporal information which are hidden in named entities can be used to determine the time related information and compute the intervals between the events to make the system more dynamic to hidden timed entities and hidden time intervals in them.

Complex scenarios involving previous booking information which will influence the future queries. The subset of requirements which we assume in our sysem can be made dynamic by computing the service requirements for each instance.

7. References

Allen J., "Towards a general theory of action and time", *Artificial intelligence*, vol. 23, nº 2, p. 123-154, 1984.

Caicedo-Castro I.-B., "Sniffer: A text description-based service search system", , PhD Dissertation, University of Grenoble, 2015.

Caicedo-Castro I.-B., Fauvet M.-C., Lbath A., Duarte-Amaya H., "Toward the highest effectiveness in text description-based service retrieval", *Document Numérique*, vol. 2-3, p. 155-177, 2015.

Cresci S., D'Errico A., Gazzé D., Lo Duca A., Marchetti A., Tesconi M., "Towards a DBpedia of Tourism: the case of Tourpedia", *Proceedings of the 2014 International Conference on Semantic Web-Poster and Demo Track, ISWC2014*, 2014.

Fagin R., Kumar R., Sivakumar D., "Comparing top k lists", *SIAM Journal on Discrete Mathematics*, vol. 17, nº 1, p. 134-160, 2003.

Faria C., Serra I., Girardi R., "A domain-independent process for automatic ontology population from text", *Science of Computer Programming*, vol. 95, p. 26-43, 2014.

Fauvet M.-C., Kamath S., Caicedo-Castro I., Lbath A., Goeuriot L., "Offering Context-Aware Personalised Services for Mobile Users", *Proc. of the Inter. Conf. on Service Computing Conf. (ICSOC) demo session*, Goa, India, 2015.

Marchetti A., Tesconi M., Abbate S., Lo Duca A., D'Errico A., Frontini F., Monachini M., "Tour-pedia: A web application for the analysis and visualization of opinions for tourism domain", *The 6th Language & Technology Conference on Human Language Technology*, p. 594-595, 2013.

Na-Lumpoon P., "Toward a Framework for Automated Service Composition and Execution: E_Tourism Applications", , PhD Dissertation, University of Grenoble, 2015.

Na-Lumpoon P., Lei M., Caicedo-Castro I., Fauvet M., Lbath A., "Context-Aware Service Discovering System for Nomad Users", *Proc. of the 7th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2013.

Ruiz-Martınez J., Minarro-Giménez J., Castellanos-Nieves D., Garcıa-Sánchez F., Valencia-Garcia R., "Ontology population: an application for the E-tourism domain", *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 7, nº 11, p. 6115-6134, 2011.

Tomai E., Spanaki M., Prastacos P., Kavouras M., "Ontology assisted decision making–a case study in trip planning for tourism", *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, Springer, p. 1137-1146, 2005.