Expert Systems with Applications 40 (2013) 3649–3664

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

SWSNL: Semantic Web Search Using Natural Language

Ivan Habernal [a,*], Miloslav Konopík [b]

[a] Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
[b] NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic

ARTICLE INFO

Keywords:
Semantic search
Natural Language Understanding
Semantic Web
Statistical semantic analysis

ABSTRACT

As modern search engines are approaching the ability to deal with queries expressed in natural language, full support of natural language interfaces seems to be the next step in the development of future systems. The vision is that of users being able to tell a computer what they would like to find, using any number of sentences and as many details as requested. In this article we describe our effort to move towards this future using currently available technology. The Semantic Web framework was chosen as the best means to achieve this goal. We present our approach to building a complete Semantic Web Search Using Natural Language (SWSNL) system. We cover the complete process which includes preprocessing, semantic analysis, semantic interpretation, and executing a SPARQL query to retrieve the results. We perform an end-to-end evaluation on a domain dealing with accommodation options. The domain data come from an existing accommodation portal and we use a corpus of queries obtained by a Facebook campaign. In our paper we work with written texts in the Czech language. In addition to that, the Natural Language Understanding (NLU) module is evaluated on another domain (public transportation) and language (English). We expect that our findings will be valuable for the research community as they are strongly related to issues found in real-world scenarios. We struggled with inconsistencies in the actual Web data, with the performance of the Semantic Web engines on a decently sized knowledge base, and others.

1. Introduction

Keyword-based search engines have proven to be very efficient on collections of unstructured textual content, e.g. web pages (Wilson, Kules, schraefel, & Shneiderman, 2010). However, if users want to find information in a structured content, e.g. in a database, the basic keyword search might not be expressive enough (Demidova, Fankhauser, Zhou, & Nejdl, 2010). A simple, yet sufficient solution on the Web can be provided by a form-based user interface which typically combines keywords with some other restrictions, according to the specific domain structure (He, Meng, Yu, & Wu, 2005). Form-based interfaces are user friendly in the sense that they do not require the user's prior knowledge of the underlying data structures. The structure is typically shown as multiple forms or menus that allow further specification of the user request. Nevertheless, it is obvious that from the user's point of view, a simple keyword input is a more straightforward approach than filling forms.

A different approach to retrieving sought information consists in using a domain-specific query language. However, as pointed out by Kaufmann and Bernstein (2007), querying structured data using a domain-specific query language, e.g. SQL or SPARQL (Prud'hommeaux & Seaborne, 2008), can be complicated for casual users. Most of the existing query languages use a precise syntax and the users are expected to be familiar with the back-end data structures. Evidently, this allows exact queries to be formulated by domain experts but on the other hand this is a big obstacle for casual users.

One step beyond the above-mentioned traditional approaches are Natural Language Interfaces (NLI). The key feature of such interface is that users can search for the required information by posing their queries using natural language (NL). It is assumed that posing NL queries is more precise than the keyword approach and at the same time more natural than the form-based approach. Recent studies (Elbedweihy, Wrigley, & Ciravegna, 2012; Elbedweihy, Wrigley, Ciravegna, Reinhard, & Bernstein, 2012) indeed confirmed that NLI offers a better user experience than the form-based search for casual users.

In the article we present a realistic approach to designing and developing a complete Semantic Web Search Using Natural Language (SWSNL) system. The system builds on the Semantic Web paradigm and thus uses ontologies as a main vehicle for storing the domain structure and all the data (the Knowledge Base, KB). Furthermore, ontologies, due to their ability to precisely capture the semantics, are also used to describe the meaning of user queries that are expressed in natural language (NL). The system can

* Corresponding author. Tel.: +420 377632456.
E-mail addresses: habernal@kiv.zcu.cz (I. Habernal), konopik@kiv.zcu.cz (M. Konopík).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.070

be seen as a search engine which accepts NL queries and performs the search in the back-end KB.

Our SWSNL system also has a few aspects that distinguish it from other related work. Firstly, we allow users to formulate their NL queries using more than a single sentence. Secondly, the Natural Language Understanding (NLU) module incorporates a stochastic semantic analysis model and does not require any syntactic parser preprocessing. The NLU module is trained from annotated data and it is language independent. Thirdly, the ontology for describing natural language queries is different from (independent of) the KB ontology. This independence enables switching between various KBs without affecting the NLU module.

The system was thoroughly tested and evaluated on three different domains. Primarily, the end-to-end evaluation was performed on the accommodation options domain using actual data acquired from the Web as well as the corpus of actual NL queries in the Czech language. The other two domains, the Czech public transportation domain and a subset of the English ATIS dataset, were used to verify the portability to different domains and languages.

The initial impulse to build the presented system was an idea to enrich an existing form-based application with a newly developed NLI. The expected benefits were targeted into better user experience and into testing the NLI approach in practice. The selected domain of accommodation options followed the idea as our preliminary survey which revealed a promising potential of such a domain. The selection of the Czech language as the domain language was not a random choice. The decision was meant to verify the technology in a different language than English.

The rest of the article is organized as follows. Section 2 outlines the context of the related work. In Section 3 we introduce our system in a schematic overview. Section 4 describes the target domain together with the techniques required to deal with actual Web data sources. A natural language corpus is also presented. Section 5 describes the formalism for capturing natural language query semantics. Section 6 proposes a statistical semantic model for analysing natural language queries and Section 7 deals with the semantic interpretation of a semantic annotation in order to perform a search in a particular KB. Section 8 thoroughly describes the evaluation of our SWSNL system. Open issues and future work are then discussed in Section 9.

2. Related work

Before we present the related work, we shortly discuss the terminology of the current research into Natural Language Interfaces. Typically, the term *Natural Language Interfaces* (NLI) is used when a system can be accessed using (written) natural language. Such a system mostly operates on structured information and, given the natural language question, it tries to find the correct answer. The family of NLI systems can be divided into various sub-classes. A *Natural Language Interfaces to Databases* (NLIDB) system holds information in a relational database. The principles of NLIDB have been adapted to the Semantic Web resulting into the *Natural Language Interfaces to Knowledge Bases* (NLIKB). In this case of NLI, the information is stored in the form of ontology, which plays the fundamental role in the Semantic Web.

We define our research as *Semantic Web Search Using Natural Language* (SWSNL). Although NLIKB covers a task similar to our approach, most of the existing NLIKB systems are designed to evaluate rather complex logical queries over the Knowledge Base. On the contrary, SWSNL can be viewed as a special case of a search engine that retrieves a set of results according to a natural language query, operating in the Semantic Web field. Furthermore, our SWSNL system does not belong to the family of question answering systems since these two fields have a very different motivation. Whereas a question answering system tries to answer a user's NL questions, the purpose of our system is to retrieve search results according to a user's NL queries.

2.1. Overview of recent NLIKB systems

A very recent system called *PowerAqua* (Lopez, Fernández, Motta, & Stieler, 2011) is an ontology-based NLI system which surpasses traditional systems by managing multiple ontology sources and high scalability. Since its NL processing module remains the same as in the previous *AquaLog* system (Lopez, Uren, Motta, & Pasin, 2007), we will review the *AquaLog* system. *AquaLog* is a portable NLIKB system which handles user queries in a natural language (English) and returns answers inferred from a knowledge base. The system uses GATE[1] libraries (namely the tokenizer, the sentence splitter, the POS tagger, and the VP chunker).

ORAKEL (Cimiano, Haase, Heizmann, Mantel, & Studer, 2008) is an ontology-based NLI system. It accepts English factoid questions and translates them into first-order logic forms. This conversion uses full syntax parsing and a compositional semantics approach. *ORAKEL* can be ported into another domain but such porting requires a domain expert to create a domain-dependent lexicon. The lexicon is used for an exact mapping from natural language constructs to ontology entities. A possible drawback of *ORAKEL*'s approach is that the system can neither handle ungrammatical questions nor deal with unknown words.

The *FREyA* system (Damljanovic, Agatonovic, & Cunningham, 2010) is an NLIKB system that combines syntactic parsing with ontology reasoning. It derives parse trees of English input questions and uses heuristic rules to find a set of potential ontology concepts (for mapping from question terms to ontology concepts) using GATE and the OntoRoot Gazetteer (Cunningham, 2011). The primary source for question understanding is the ontology itself. If the system encounters ambiguities, a clarification dialogue is offered to the user. The potential ontology concepts retrieved from the question analysis are then transformed into SPARQL. The system was tested on Mooney: Geography dataset (Tang & Mooney, 2001).

A portable NLIKB system called *PANTO* (Wang, Xiong, Zhou, & Yu, 2007) is based upon the off-the-shelf statistical parser Standford Parser and integrates tools like WordNet and various metrics algorithms to map the NL question terms to an intermediate representation called *QueryTriples*.[2] This semantic description is then mapped onto the *OntoTriples* that are connected to entities from the underlying ontology. This step involves a set of 11 heuristic mapping rules. Finally, OntoTriples are represented as SPARQL queries. The main idea of transforming a NL question into triples in PANTO is based upon the empirical observation that two nominal phrases from a parse tree are expected to be mapped onto a triple in the ontology. The system was evaluated on the Mooney dataset and the output of PANTO was compared to the manually generated SPARQL queries.

The *NLP-Reduce* system (Kaufmann, Bernstein, & Fischer, 2007) does not involve any advanced linguistic and semantic tools and depends on matching the query words to the KB instances. Its core part is a *query generator* which is responsible for creating SPARQL query given the words and the lexicon extracted from the KB. The major strength of the system is its good portability as it does not depend on any complex NLP query processing.

[1] http://www.gate.ac.uk.

[2] A mixed terminology (*triplets* vs. *triples*) appears in the literature. In this paper we use *triples* as proposed in the RDF standard, http://www.w3.org/TR/rdf-concepts/#section-triples.