Knowledge-Based Systems 22 (2009) 443–448

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Question answering based on pervasive agent ontology and Semantic Web

Qinglin Guo [a,b,*], Ming Zhang [a]

[a] Department of Computer Science and Technology, Peking University, Beijing 100871, China
[b] School of Computer Science and Technology, North China Electric Power University, Beijing 102206, China

ARTICLE INFO

Article history:
Received 15 March 2008
Accepted 4 June 2009
Available online 10 June 2009

Keywords:
WWW
Pervasive agent ontology
Semantic Web
Question answering

ABSTRACT

Semantic Web technologies bring new benefits to knowledge-based question answering system. Especially, ontology is becoming the pivotal methodology to represent domain-specific conceptual knowledge in order to promote the semantic capability of a QA system. In this paper we present a QA system in which the domain knowledge is represented by means of ontology. In addition, personalized services are enabled through modeling users' profiles in the form of pervasive agent ontology, and a Chinese Natural Language human–machine interface is implemented mainly through a NL parser in this system. An initial evaluation result shows the feasibility to build such a semantic QA system based on pervasive agent ontology, the effectivity of personalized semantic QA, the extensibility of pervasive agent ontology and knowledge base, and the possibility of self-produced knowledge-based on semantic relations in the pervasive agent ontology.

1. Introduction

Semantic Web technologies bring new benefits to knowledge-based question answering system. Especially, ontology is becoming the pivotal methodology to represent domain-specific conceptual knowledge in order to promote the semantic capability of a QA system.

Specific research in the areas of QA has been advanced in the past couple of years particularly by the QA track of the TREC–QA competitions [1]. The TREC–QA competitions focus on open-domain systems, i.e. systems that can potentially answer any generic question. Therefore, these competitions are based on large volumes of unstructured text, which makes deep text analysis become resource-consuming tasks. In contrast, a QA system working on a specific technical domain can make use of the specific domain-dependent terminology to recognize the true meaning included in a segment of natural language text. So we realize that the terminology plays a pivotal role in a technical domain such as Java programming. A great deal of work has been done representing domain-specific concepts and the terminology by means of Ontology, i.e. UMLS [2]. Recent research advancements on knowledge representation with Semantic Web and pervasive agent ontology have proved that this methodology is able to promote the semantic capability of a QA system.

The Semantic Web is a Web that includes documents, or portions of documents, describing explicit relationships between things and containing semantic information intended for automated processing by our machines. It operates on the principle of shared data. When you define what a particular type of data is, you can link it to other bits of data and say "that's the same", or some other relation. For example, "zip" in my QA system based on Semantic Web is the same as "zip" in my friends. Although it gets more complicated than this, which is basically what the Semantic Web is all about, sharing data through ontologies, and processing it logically. Trust is also important, as the trust of a certain source is fully in the hands of the user. This is a fully decentralized system: "you can not make something be the centre of all knowledge". Although the Semantic Web is a Web of data, it is intended primarily for humans; it would use machine processing and databases to take away some of the burdens we currently face so that we can concentrate on the more important things that we can use the Web for.

For example, recent research in information processing has focused on health care consumers [3]. These users often experience frustration while seeking online information, due to their lack of understanding of medical concepts and unfamiliarity with effective search strategies. We are exploring the use of semantic relationships as a way of addressing these issues. Semantic information can guide the lay health consumer by suggesting concepts not overtly expressed in an initial query. For example, imagine that a user submits a full question to a search system in the health care domain to find out whether exercise helps prevent osteoporosis. The semantic relationship prevents in the proposition representing the question, namely "exercise prevents osteoporosis", can support this effort; prevents might be used with osteoporosis to determine additional ways of avoiding this disorder. We present an analysis of semantic relationships that were

* Corresponding author. Address: School of Computer Science and Technology, North China Electric Power University, Beijing 102206, China.
E-mail addresses: qlguo@pku.edu.cn, qlguo88@sohu.com (Q. Guo), mzhang@pku.edu.cn (M. Zhang).

0950-7051/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2009.06.003

manually extracted from questions asked by health consumers as well as answers provided by physicians. Our work concentrates on samples from Ask-the-Doctor Web sites. The semantic network from the unified medical language system (UMLS) [4] served as a source for semantic relationship types and this inventory was modified as we gained experience with relationship types identified in the health consumer texts.

A semantic relationship associates two (or more) concepts expressed in text and conveys a meaning connecting those concepts. A large variety of such relationships have been identified in several disciplines, including linguistics, philosophy, computer science, and information science. Some researchers have organized hierarchies of semantic relationships into meaningful but not formal structures. Others examine specific relationships in depth, for instance, subsumption, temporality, and meronymy. In addition, ontologies contain semantic relationships that are elements of the overall system. WordNet, for example, contains these primary relationships between concepts: hypernymy, antonymy, entailment, and meronymy (part–whole). A number of projects have involved the study of semantic relationships specifically within the domain of medicine. Work on the GALEN common reference model examined part–whole relationships and other aspects of "tangled" taxonomies. Other ontology projects, such as the foundational model of anatomy (FMA), are central to the delineation of relationships for use in specific types of applications, in this case representation of anatomical structures.

2. Semantic Web and agent-based Semantic Web services query

Making the Web more meaningful and open to manipulation by software applications is the objective of the Semantic Web initiative. Knowledge representation and logical inference techniques form the backbone. Annotations expressing meaning help software agents to obtain semantic information about documents [5]. For annotations to be meaningful for both creator and user of annotations, a shared understanding of precisely defined annotations is required. Ontologies – the key to a Semantic Web – express terminologies and semantic properties and create shared understanding. Ontologies consist of hierarchical definitions of important concepts in a domain and descriptions of the properties of each concept, supported by special logics for knowledge representation and reasoning. Web ontologies can be defined in DAML + OIL – an ontology language based on XML and RDF/RDF Schema. Some effort has already been made to exploit Semantic Web and ontology technology for the software engineering domain [6]. DAML-S is a DAML + OIL ontology for describing properties and capabilities of Web services, which shows the potential of this technology for software engineering. Formality in the Semantic Web framework facilitates machine understanding and automated reasoning. DAML + OIL is equivalent to a very expressive description logic [7]. This fruitful connection provides well-defined semantics and reasoning systems. Description logic is particularly interesting for the software engineering context due to a correspondence between description logic and dynamic logic (a modal logic of programs). We propose to define a semantic interface definition language IDL and a reasoning technique for component matching in form of ontology. The connection between description logic and modal logics allows us to introduce reasoning about component and service matching within a Semantic Web framework.

In the conventional Web services approach exemplified by WSDL or even by DAML services, the communicative intent of a message is not separated from the application domain. This is at odds with the convention from the multi-agent systems world, where there is a clear separation between the intent of a message, which is expressed using an agent communication language, and the application domain of the message, which is expressed in the content of the message by means of domain-specific ontologies. This separation between intent and domain is beneficial because it reduces the brittleness of a system. If the characterization of the application domain (the ontology) changes, then only that component which deals with the domain-specific information need change; the agent communication language component remains unchanged. In addition, the domain-neutral performatives may be combined to form common patterns of interaction such as contract nets, markets or auctions that enable the behaviour of a system to be considered in more abstract terms.

When the service in the QA example is invoked, the value of the input parameter should be an instance of the class restriction that is given as the input parameter types in both the profile and the process descriptions. For the various query performatives (query-if, query-ref and subscribe), this input parameter contains the query expression that would be contained in the message content in a conventional agent-based system. However, there is as yet no standard query language for RDF, DAML + OIL or OWL, although there are several under development, including DAML rules [8] (which builds on DAML + OIL and expresses queries as Horn clause-like structures), the DAML query language.

As an example, the domain ontology that we have designed for this application is centred on events and reports of events. We have taken the approach that communication in the system will be about these events and reports (rather than about any persistent world state which the reports might suggest), so the queries can be expressed using the anonymous resource technique by specifying the properties that the report (and the event it contains) must possess. It should be noted, however, that we did not specifically design the ontology in this report to circumvent the expressive limitations of our chosen query language, but rather that the query language was chosen because it was appropriate for use with the domain ontology that we had already designed.

3. Building Web service domain ontologies

Several Web service tasks can be automated by using semantic descriptions. For example, service offers and requests can be matched automatically [9]. This matchmaking is flexible because it allows retrieving services that only partially match a request but are still potentially interesting. For example, the hotel booking service will be considered a match for a request for accommodation booking services, if the used domain ontology specifies that hotel is a kind of accommodation. This matchmaking is superior to the keyword search.

A basic requirement for being able to perform complex reasoning with multiple Semantic Web service descriptions is that (many of) these descriptions should use concepts of the same (or a small number of) domain ontology. If each Web service description uses different domain ontology then a mapping between these ontologies has to be performed before any reasoning task can take place. However, pervasive agent ontology mapping itself is a difficult and largely unsolved problem in the Semantic Web. Therefore, quality domain ontology will reflect a high percentage of the domain's terminology so that many Web services in that domain can be described with its concepts. This requirement makes the building of the domain ontologies difficult, as it is evident in the next section where we present an analysis of the pervasive agent ontology building process in two concrete research projects.

The creation of Semantic Web service descriptions is a time consuming and complex task whose automation is desirable, as signaled by many researchers in this field. This task can be broken down in two smaller tasks. First, acquiring a suitable Web service domain ontology is a prerequisite when creating Semantic Web