Expert Systems With Applications 53 (2016) 172–191

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A multi-agent conversational system with heterogeneous data sources access

Eduardo M. Eisman[a,*], María Navarro[b], Juan Luis Castro[a]

[a] Department of Computer Science and Artificial Intelligence (DECSAI), University of Granada, C/ Daniel Saucedo Aranda, s/n, 18071 Granada, Spain
[b] CEMSE Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Kingdom of Saudi Arabia

ARTICLE INFO

Keywords:
Natural language interfaces
Virtual assistants
Embodied conversational agents
Multi-agent systems
Semantic grammars
Ontologies

ABSTRACT

In many of the problems that can be found nowadays, information is scattered across different heterogeneous data sources. Most of the natural language interfaces just focus on a very specific part of the problem (e.g. an interface to a relational database, or an interface to an ontology). However, from the point of view of users, it does not matter where the information is stored, they just want to get the knowledge in an integrated, transparent, efficient, effective, and pleasant way. To solve this problem, this article proposes a generic multi-agent conversational architecture that follows the divide and conquer philosophy and considers two different types of agents. Expert agents are specialized in accessing different knowledge sources, and decision agents coordinate them to provide a coherent final answer to the user. This architecture has been used to design and implement SmartSeller, a specific system which includes a Virtual Assistant to answer general questions and a Bookseller to query a book database. A deep analysis regarding other relevant systems has demonstrated that our proposal provides several improvements at some key features presented along the paper.

1. Introduction

As many studies reveal (Chai et al., 2001a; 2001b; Kaufmann & Bernstein, 2010; Zhou, Mohammed, & Zhang, 2012), there is a clear preference of users for full natural language query interfaces rather than keywords, formal query languages, or menu driven interaction. In addition, the interest of users in a particular site decreases exponentially with the increase in the number of mouse clicks (Huberman, Pirolli, Pitkow, & Lukose, 1998). This fact is emphasized even more if we talk about mobile devices, where traditional input interfaces are very limited. Natural language systems are able to improve the perceived usefulness, ease-of-use, and efficiency, which in turn account for positive attitude and intention to use those systems.

From an economical point of view, the increase of e-commerce spending supposes a great opportunity for natural language interfaces. According to comScore[1] (one of the most important Internet marketing research companies), Q1 2014 saw desktop-based U.S. retail e-commerce spending rise 12% year-over-year to $56.1 billion, marking the 18th consecutive quarter of positive year-over-year growth. M-commerce spending on smartphones and tablets added $7.3 billion for the quarter, up 23% vs. year ago, for a digital commerce spending total of $63.4 billion in the first quarter. In addition, some events like Alibaba's[2] Singles' Day sales confirm that m-commerce is more and more important. During that day, the world's biggest online retail sales day, sales exceeded predictions at $9.3 billion, shipping 278 million orders, 43% of which were placed on mobile devices.[3] This change in consumer behavior reflects the necessity of defining more intelligent ways of interacting with websites and product databases rather than traditional keyword search.

This paper focuses on the problem of accessing heterogeneous data sources using natural language dialog. As we will see in Section 2, most of the natural language interfaces (NLIs) that can be found in literature are usually oriented to solve very specific problems in which the knowledge is stored in just one type of source (e.g. NLIs to relational databases, NLIs to ontologies...). However, the reality is usually quite different.

As can be seen in Fig. 1, when users interact with an NLI, they have a single objective in mind: getting the information they want at that moment. From their point of view, it does not matter if

* Corresponding author. Tel.: +34 958 246163; fax: +34 958 243317.
E-mail addresses: eisman@decsai.ugr.es (E.M. Eisman), mariaisabel.navarrojimenez@kaust.edu.sa (M. Navarro), castro@decsai.ugr.es (J.L. Castro).

[1] https://www.comscore.com/Insights/Press-Releases/2014/5/comScore-Reports-56-1-Billion-in-Q1-2014-Desktop-Based-US-Retail-ECommerce-Spending-Up-12-Percent-vs-Year-Ago

[2] http://www.alibaba.com/

[3] http://www.bbc.com/news/business-29999289

http://dx.doi.org/10.1016/j.eswa.2016.01.033
0957-4174/

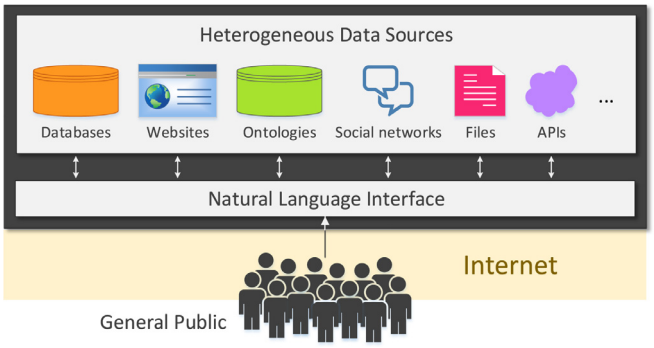

Fig. 1. Natural language interface for general public.

it is an NLI to a database, an ontology, or an XML file, nor if it is a virtual assistant that can instruct them in using a website or answer general questions. What is really important is that, if we want these systems to be applied to many different fields and be extensively used by the general public and not only by expert users or database administrators, they must put all that heterogeneous information at the service of users in an integrated, transparent, efficient, effective, and pleasant way.

As we will see in Section 3, the architectures used by this type of systems present some advantages and certain disadvantages. And although many of these systems have been more or less usable in practice for different application domains, they present several problems. Many of them are not interactive or their interactivity is very limited. However, users often do not know where to find the information, or their requests are vague and they need to complete them little by little with additional information, which could be provided in a dialog. For this reason, the lack of memory and a dialog management module to keep track of the conversation and use the context in an appropriate way is a big problem. Other issues regarding these systems are that most are not ready to work with heterogeneous data sources at the same time and in a transparent way. Moreover, the majority cannot handle fuzzy concepts such as "*cheap*" or "*recent*", or temporal queries involving relative dates, which is essential. On the other hand, they do not use 3D characters that can emotionally engage in conversation with users, making the interaction process friendlier. Finally, sometimes they are simple prototypes that are far from being a powerful and easy to use system that can be used in a real environment with a large number of users of any type.

Our proposal to address these problems and improve what other systems provide is to use a multi-agent approach that allows applying a divide and conquer philosophy. In this way, a set of expert agents specialized in concrete domains facilitates the access to different knowledge sources, and a series of decision agents interacts with them to coordinate them and provide a unique final answer to the user. This proposal will be analyzed in depth in Section 4, and it will include the design and development of a specific conversational system for a bookstore. This system will be used in Section 5 to make a comparative analysis in order to confirm that our proposal provides several improvements in the aforementioned features. Section 6 will present some details about the implementation and usage of the system. Finally, Section 7 will include some general conclusions and directions for future work.

2. Related work

The problem of Natural Language Interfaces (NLIs) started more than four decades ago. NLI systems are divided into three main categories, depending on how knowledge is structured:

- NLIs to structured data: usually closed-domain, they can work either on relational databases translating the query to SQL (the so called Natural Language Interfaces to Databases, or NLIDBs), or on ontologies translating the query to SPARQL, for example.
- NLIs to non-structured or semi-structured data: usually open-domain, they process huge amounts of documents to find the answer to a question.
- interactive NLIs: used in dialog systems, they have memory to remember previous questions.

In this paper we will focus on NLIs to structured data and interactive NLIs. In order to be able to better understand the benefits of our proposal, we will show the main features of some NLIDB systems that can be found in literature, from classic implementations to state-of-the-art prototypes.

Since their appearance at the end of the 60s, many different systems have been proposed. During the first years, all the systems were domain specific. They were tied to a particular database and they were difficult to port. They used to be based on pattern matching techniques and syntactic trees. During the seventies and eighties, this kind of systems were improved with the use of new techniques such as semantic grammars or intermediate representation languages, which were independent from the database. However, it was not until the end of that period that some commercial systems started to appear (Sijtsma & Zweekhorst, 1993) (e.g. Intellect, Natural Language, Q&A, LanguageAccess by IBM, English Query by Microsoft, or English Wizard by Linguistic Technology Corporation). In spite of their initial popularity, these systems gradually disappeared due to several problems: bad performance, significant effort required to develop specialized systems for individual databases that could not be easily adapted to work with different domains, language coverage not obvious to users, ambiguity of natural language, and so on. Nevertheless, after some years in the oblivion, they started to live a second youth with the arrival of the new century, and nowadays they are mature enough as Apple Siri,[4] Google Now,[5] Microsoft Cortana,[6] or Amazon Echo[7] show, although these systems are much more complex than a simple NLIDB.

We will talk more about the different types of architectures of these systems in Section 3. But first, we will briefly describe some specific examples. Although in Section 5 we will make a thorough analysis considering some of these systems, more details on any of them can be found in the corresponding paper.

LUNAR (Woods, Kaplan, & Nash-Webber, 1972) is one of the most well known NLIDB systems. It presented a natural language interface to a database containing chemical analyses of Apollo-11 moon rocks. LUNAR used an Augmented Transition Network (ATN) parser, Woods' procedural semantics, and two databases for chemical analysis and literature references. Although it was not broadly used, its performance was impressive for that time, with an accuracy of 78%, or 90% if dictionary errors were corrected.

RENDEZVOUS (Codd, 1974) was intended to engage users in free (relatively unrestricted) dialogs to help formulate queries over relational databases. It placed special emphasis on query paraphrasing and engaging users in clarification dialogs. The objective was to make sure that the intended meaning of the user's question had been correctly captured before routing the formal query to the relational DBMS.

LADDER (Hendrix, Sacerdoti, Sagalowicz, & Slocum, 1978) was designed to access information about US Navy ships. It could be used with large distributed databases and different database

[4] https://www.apple.com/ios/siri/
[5] https://www.google.com/landing/now/
[6] http://www.windowsphone.com/en-US/how-to/wp8/cortana/
[7] http://www.amazon.com/oc/echo