Toward an Ontology-Based Chatbot Endowed with Natural Language Processing and Generation

Amine Hallili

Toward an Ontology-Based Chatbot Endowed with Natural Language Processing and Generation

Amine Hallili

Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
hallili@i3s.unice.fr

Abstract. With the last evolution of the web, several new means of communication have showed up. In the commercial domain, chatbot technologies are now considered as essential for providing a wide range of services (e.g. search, FAQ, assistance) to the end-user, and to make a client a faithful customer. In this paper, we propose an on-going work on the definition and implementation of SynchroBot, an ontology-based chatbot that relies on Semantic Web and NLP models and technologies to support user-machine dialogical interaction in the e-commerce domain.

Keywords: Chatbot, Artificial Intelligence, Natural Language Processing, Natural Language Generation, Semantic Web

1 Introduction

During the last decades, our way of consuming information has totally changed with the emergence of new means of communication (e.g. forums, FAQ, social networks, semantic search engines, mobile applications, and text to speech systems) which provide us with different possibilities of handling and dealing with information on the web. At the same time, researchers in Natural Language Processing (NLP) and Semantic Web domains have proposed new approaches to model and implement more and more complex systems capable of interpreting natural language, of reasoning, and of assisting end-users (e.g. Chatbots [1], Expert Systems [10] , multi agent systems [15], and Question Answering systems [9]). Besides covering both open and close domains (e.g. social, commercial, scientific), such systems aim to be autonomous, self-learning and they can replace humans in performing several tasks. My PhD research proposal, whose preliminary works I present in this paper, focuses on chatbot systems, which we classify in two different categories: Question Answering Systems (QA) and Dialog Systems (DS). On the one hand, Question Answering systems aim at finding answers to factual queries in either a Knowledge Base (KB) or raw text and to return them to the user. The answer can be just a textual string (e.g. [4]) or it can be enriched by other meta-information or well-formed sentences, obtained by applying Natural Language Generation (NLG) techniques (e.g. [2,5]). In spite of their efficiency in retrieving the information, such systems lack the capability of handling the links between sequential questions as in a conversation. On the other hand, Dialog Systems aim at keeping in memory the links between consecutive questions in order to ensure a logical conversation mode with the user (e.g. [13,7]). Nevertheless, most of these systems do not rely on robust and flexible KBs allowing them to extract information from multiple sources and to reason over the data. The goal of our work is to combine the strenghts of the two categories of systems discussed above, and to propose a dialog system that relies on i) a rich KB for data extraction and reasoning, ii) NLP tools to interpret user's question, and iii) NLG techniques to generate well-formed sentences. The system will ensure the following type of conversation:

<User>	*Give me the price of a Nexus 5!*
<System>	*the price of Nexus 5 is 400$*
<User>	*and who sells it?*
<System>	*several sellers were found. The main one is Google! Do you want to see other sellers?*
<User>	*No, show me the white version, sold by Google and located in France!*
<System>	*here are the images of Nexus 5 white version, sold by Google and located in France...*

The remainder of this paper is organized as follow: Section 2 presents our preliminary approach and implementation. In Section 3 we describe our ongoing works along with our perspectives for future works.

2 SynchroBot : A Preliminary Approach

The approach we propose relies on the Semantic Web[1] paradigm, which covers structuring, linking, sharing and reusing data through applications, enterprises and communities. For that, it provides a number of information modeling frameworks e.g. Resource Description Framework (RDF) and RDF Schema (RDFS). The preliminary approach we propose here toward an ontology-based chatbot covers three aspects, namely i) Knowledge based System ii) Question Interpretation iii) Natural Language Generation. Currently we focus on modeling and implementing an efficient and robust QA system that will be the corner stone for our future Dialog System.

2.1 A Knowledge Based System

Our approach relies on the use of exiting tools, resources and information (e.g. FAQ, API, system logs) in order to create a KB in RDF, which means that the data will be represented as triples: `<Subject, property, Value>`. For example, the following sentence "Google sells Nexus 5" can be expressed in RDF as `<sbr:Google, sbo:sells, sbr:Nexus_5>`. We have created an ontology that

[1] www.w3.org/2001/sw/

describes the classes (e.g. Product, Category, Seller, etc.) and properties (e.g. sells, price, locatedIn, etc.) of the KB in the e-commerce domain (the focus of Synchrobot). For instance Google type is `sbo:Seller` and has the `sbo:sells` property. Likewise, Nexus_5's type is
`sbo:Product` and has the `sbo:locatedIn` property. Also, with our ontology we infer, among others, that Nexus 5 is sold by Google (<`sbr:Nexus_5`, `sbo:soldBy`, `sbr:Google`>). Furthermore, every property is annotated in both French and English, by a number of labels that have the same meaning (e.g. `sbo:sells` will have “sell”, “trade”, “vend”, “commercialize”, “market”, etc. as labels), which will be used to match the terms in the question, in order to identify the queried property. The current version of the KB is composed of 500000 product descriptions that we retrieved by using eBay APIs to transform eBay data to RDF.

2.2 Question Interpretation

As regards the natural language question interpretation, our approach focuses on textual information as input and relies on the work described in [3] which requires identifying three aspects i) the Expected Answer Type (EAT), which is the type of the resource that we are looking for, ii) the property, representing the relation linking the entity on which the question is asked to its answer, and iii) the Named Entity (NE) representing the subject of the given question. In this example “Who sells Nexus 5?”, the EAT is `sbo:Seller`, the property is `sbo:sells` and the NE is `sbr:Nexus_5`.

Named Entity Recognition : To identify the NE, we aim at using natural language processing techniques (e.g. Named Entity detection and linking) to retrieve all possible NEs from the KB by matching the user'question to KB property values (e.g. name, description, etc.). Then, relevant NEs will be used in querying the KB according to the assigned retrieving score which we determine by using the folowing strategy :

In order to assign a score to the quality of the search, our strategy relies on several aspects : First, all KB properties used for the search are ordered depending on their relevance. For instance, matching the user's question to the `sbo:hasLegalName` property will be more accurate than matching it to the `sbo:hasDescription` property and so on. Based on that, a relevance coefficient is assigned to each property and used in both the score computing and the determination of relevant NEs. Second, a score is assigned to the accuracy of the matching of the user’s question and the resources found in the KB. In other word, the bigger the matched words, the higher score will be (an exact match will result of a perfect score and the found resource will be directly used). Finally, we focus on the number of the retrieved resources to determine the precision of our result. This means that the fewer resources we find, the higher the precision of the search.

Property Detection : We detect the property by matching its annotated labels with the user's question [6] and following a scoring strategy we pick up the relevant property.

We are also able to recognize questions with two relations as shown below in figure 1. For instance, the following example "Give me the address of the Nexus 5's sellers!" contains two properties (i.e. `sbo:address`, `sbo:sells`) meaning that the question can be divided in two sub-questions: "Give me the Nexus 5's sellers!" and "Give me their addresses!". This can be done thanks to the fact that the properties in our ontology have domain and range that allow the detection of 2-relation (n-relations in general). Concretely we aim at constructing a relational graph representing the user's question that contains the identified properties along with the found resources (e.g. Named Entities), while comparing both the NE type and the identified property domain. In the given example, the property `sbo:address`, with domain `sbo:Seller`, will have the best score along with the NE `sbr:Nexus_5` that have type `sbo:Product` which differs from `sbo:Seller`, this leads to the creatation of a relational graph with two nodes which are two properties namely `sbo:address` and `sbo:soldBy` and means that the question has more than one relation. These elements (NE, property and EAT) will be used to generate a SPARQL query that retrieves results from the KB.
Note that the system will select the property `sbo:soldBy` instead of `sbo:sells` during the process time. This is due to the fact that the system will use inference to pick up more properties to be able to construct a relational graph that represents the user's questio. In our example, only the `sbo:soldBy` property which is the inversed property of the selected one `sbo:sells` will give results when creating the relational graph.

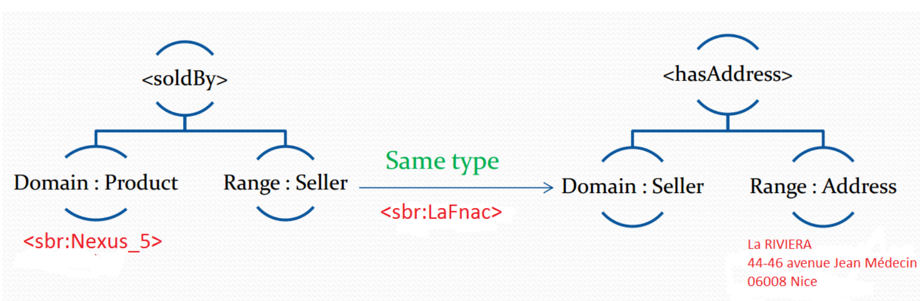

Fig. 1. Relational graph for 2 relations

Expected Answer Type (EAT) Detection : After detecting relevant properties, we aim at using their respective `Domain` types. For instance, as shown in figure 1, the following properties' `Domains` (`sbo:Product`, `sbo:Seller`, `sbo:Address`) will be used along with the score assigned to their respective property. This allows us to sort all the detected EATs before adding them in the generated query that the system use to retrieve results from the KB.

2.3 Natural Language Generation

In order to answer questions with a generated sentence, we propose that each property in our ontology will be mapped with a list of generic response patterns. Our challenge is to be able to replace dynamically some particular parts of the pattern to return well-formed answers. For instance, we take the example "Give me the price of a Nexus 5!" and considering that the identified property `sbo:price` matches the pattern *[The price of {Product} is {Value}]*, so after replacing the {Product} and {Value} parts we can answer that "The price of Nexus 5 is 400$". We also use the `sbo:mediaType` property to show more interesting information to the user after giving the well-formed answer (e.g. image, video, map, etc.). For instance, when a user asks "Show me the white version of Nexus 5!", the system infers that the user is more interested in viewing images rather than just textual information, and as a result, images will be displayed in this case.

3 Ongoing and Future Work

This paper sketches out the PhD research project I have recently started, and describes the preliminary steps representing the bases of a number of directions that we consider for future work:

As ongoing and short term works we are planning to improve the NE recognition by using well-known and efficient algorithms (e.g. KNN, Similarity, N-Gram and TF-IDF scoring) in order to gain more precision, to spot complex and ambiguous resources, and to be able to diversify given answers. We have also begun to reuse other famous ontologies that exist in the literature and cover the commercial domain. we have started to use the schema.org [11] ontology but due to its parial coverage, we have decided to use more specific ontology (GoodRelations [8] ontology) which fully covers the commercial domain.
Furthermore, we consider answering n-Relation questions by the construction of a Relational Graph representing the question's NEs and properties. For that, we will study the possibility to generalize the 2-relation question answering approach explained in the previous section, to n-Relation questions. Moreover, we intend to integrate the relational pattern matching module of QAKiS system [4] that exploits Wikipedia pages to extract lexicalisations of ontological relations. More specifically, we will use website APIs, web services [14] and product pages to automatically extract and create generic property and response patterns. This will ensure more precision in detecting properties expressed in the user's question and it will allow to answer questions in different ways.

As middle term improvements, we intend to focus our work on the dialog mode part that we will integrate on top of our proposed approach, so that we can propose an approach that ensures our targeted scenario. For that, we will investigate the communicative behavior approaches (e.g. pause, resume, and to switch between interactive tasks [13]), dialog management systems (e.g. [7]) and in particular, the ontology-based dialog systems (e.g. [12] which correspond perfectly to the kind of system that we want to implement.

References

1. J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Toward conversational human-computer interaction. *AI Magazine*, 22(4):27–38, 2001.
2. A. Augello, G. Pilato, G. Vassallo, and S. Gaglio. A semantic layer on semi-structured data sources for intuitive chatbots. In *CISIS*, pages 760–765, 2009.
3. E. Cabrio, J. Cojan, A. P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon. Qakis: an open domain qa system based on relational patterns. In *International Semantic Web Conference (Posters & Demos)*, 2012.
4. E. Cabrio, J. Cojan, A. Palmero Aprosio, and F. Gandon. Natural language interaction with the web of data by mining its textual side. *Intelligenza Artificiale*, 6(2):121–133, 2012.
5. J. Chai, V. Horvath, N. Nicolov, M. Stys, A. Kambhatla, W. Zadrozny, and P. Melville. Natural language assistant: A dialog system for online product recommendation. *AI Magazine*, 23:63–75, 2002.
6. D. Damljanovic, M. Agatonovic, and H. Cunningham. Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In *ESWC (1)*, pages 106–120, 2010.
7. T. Heinroth and D. Denich. Spoken interaction within the computed world: Evaluation of a multitasking adaptive spoken dialogue system. In *COMPSAC*, pages 134–143, 2011.
8. M. Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *EKAW*, pages 329–346, 2008.
9. L. Hirschman and R. J. Gaizauskas. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300, 2001.
10. S.-H. Liao. Expert system methodologies and applications - a decade review from 1995 to 2004. *Expert Syst. Appl.*, 28(1):93–103, 2005.
11. P. Mika and T. Potter. Metadata statistics for a large web corpus. In *LDOW*, 2012.
12. D. Milward and et al. Ontology-based dialogue systems, 2003.
13. B. Pakucs. Towards dynamic multi-domain dialogue processing. In *INTERSPEECH*, 2003.
14. D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli, and N. Reithinger. Smartweb handheld - multimodal interaction with ontological knowledge bases and semantic web services. In *Artifical Intelligence for Human Computing*, pages 272–295, 2007.
15. M. J. Wooldridge. *An Introduction to MultiAgent Systems (2. ed.)*. Wiley, 2009.