An Italian Question Answering System for Structured Data based on Controlled Natural Languages

Lucia Siciliani and **Pierpaolo Basile** and **Giovanni Semeraro**
Department of Computer Science, University of Bari Aldo Moro, Italy
firstname.lastname@uniba.it

Matteo Mennitti
Sud Sistemi srl, Italy
mennittim@sudsistemi.it

Abstract

Question Answering over structured data represents one of the main challenges in the field of Natural Language Processing since it requires to render natural language, which is used by people every day, into a formal language, which can be processed by a machine. This task is particularly tricky due to the gap between the vocabularies adopted by users and the formalism that characterizes any query language. For this reason, although its birth as a discipline dates back to the late sixties, Question Answering over structured data is still accomplished to an unsatisfying degree. This result is even more critical if we take into account languages different from English, for which the amount of available resources is limited. In this paper we present MULIB, a Question Answering system capable of answering questions in Italian over both Knowledge Bases and databases.

1 Introduction and Motivation

Question Answering (QA) over structured data has the aim to interpret a natural language question issued by the user and retrieve an answer from a structured data source. Nowadays, the task of QA over structured data is usually performed over Knowledge Graphs (KGs), which encode an enormous amount of information and can thus provide a broad knowledge on many different domains.

However, QA over structured data has its roots in the late sixties as an attempt to make databases easily accessible even by non-expert users. For this reason, QA systems were initially referred to as "Natural Language Interfaces".

Apart from the technical differences existing between KGs and databases, they still share the same properties hold by any structured resource: a Data Representation Language (DRL) allows describing the data in a data source, and a Data Query Language (DQL) is then used to retrieve the data. The standard DQL for databases is SQL, while its correspondent for KGs is SPARQL. The main goal of a QA system is to bridge the so-called *lexical gap* existing between the vocabulary adopted by the user and the labels used within the structured data source. In this way a QA system can allow users to have access to the information stored in the structured data source with no need for mastering a DQL: the system has to take over the management to this translation, hiding it to the user.

Due to its complexity, the majority of works available at the state of the art exploit a combination of several NLP techniques to process the question and transform it into its DQL equivalent. For this reason, the results available at the state of the art appear even more critical when looking for relevant solutions for non-English languages.

This problem is accentuated even more by the shortage of multilingual datasets. For example, the QALD evaluation campaign[1], starting from its third edition, has included a task for Multilingual Question Answering over DBpedia. The dataset created for this task provides each question in seven different languages (i.e. English, German, Spanish, Italian, French, Dutch, and Romanian) along with its SPARQL translation. Even if the dataset actually includes non-English languages, the SPARQL translation always makes use of the resources of the English version DBpedia since many properties and entities do not have a label for the aforementioned languages.

[1]http://qald.aksw.org/

Other datasets for Question Answering over Structured Data, like Simple Question (Bordes et al., 2015) and Web Question (Berant et al., 2013) are focused only on the English language and do not provide the translation for other languages. The same issue affects also the datasets available for the evaluation of Natural Language Interfaces for databases like the U.S. Geography database (Geoquery[2]) or IMDb[3].

For all these reasons, there are only a few systems which propose an approach applicable for Italian. FuLL (Bombara et al., 2005) is a NLI for geographical data banks. FuLL exploits a fuzzy engine and a dialog manager to interpret the question inserted by the user and handle subjective elements (like the magnitude of adjectives) and ambiguous requests. However, in order to make the system more accurate, the authors have focuses only over a specific domain.

QAnswer (Diefenbach et al., 2017) is one of the few QA systems with an architecture completely independent from the language thus it can process many different languages including Italian. The system splits the question in n-grams and tries to match them with the resources of the underlying knowledge graphs. Based on the retrieved resources, it generates all the possible queries that could satisfy the user's information need. Multilingualism is obtained by avoiding the usage of any NLP tool which could affect the performance of the system, especially for those languages where the accuracy of those tools is still very low. On the other hand, the main disadvantages of this approach are that the identification of relations is based just on the dictionary and the syntax of the question is ignored thus meaning that the lack of resources in a certain language can deeply affect the results.

Based on these observations, we decided to develop a QA system for the Italian language. Our approach is based on the one adopted in CANaLI (Mazzeo and Zaniolo, 2016) which obtain the best results within the QALD-6 evaluation campaign (Unger et al., 2016). CANaLI makes use of controlled natural languages and an auto-completion mechanism to guide the user toward the formulation of a natural language question which is then processed using a finite state automaton. By analyzing the advantages and the limitation of this approach, we developed a new system which is capable of reducing the lexical gap and extended it to cover the Italian language and to support queries over traditional databases.

The paper is organized as follows: in Section 2 we will introduce and describe our system MULIB a QA system capable of answering natural language questions written in Italian over an underlying structured data source, in Section 3 is described the evaluation we performed to assess MULIB's effectiveness, finally in Section 4 we will discuss the results obtained by MULIB and outline the future directions for our work.

2 Methodology

2.1 Bridging the lexical gap

As stated in Section 1, QA systems like CANaLI can achieve good results if the syntactic structure of the question is compliant with the controlled natural language.

The main drawback of this approach lies in the vocabulary that can be accepted by the finite state automaton. In fact, it is created by collecting the labels of the resources in the KG and a match exists only if there is a complete string matching, hence only those labels can be employed in the question. A simple example is represented by the question *Who is the writer of the Divine Comedy?*. Since there is no string matching between the words "writer" and the label of the property "author", CANaLI is not able to retrieve the right answer.

This method appears to be in contrast with what discussed in Section 1 regarding the lexical gap since it requires the user to know in advance how data is stored in the data source. In order to cope with this problem, we extended the vocabulary using an approach based on distributional semantics methods, i.e. Word2Vec (W2V) (Mikolov et al., 2013). The vector space was built upon Wikipedia abstracts in order to obtain representation which could be suitable with an open domain scenario. In this way, if the data source is changed, there is no need to re-train the model to adapt it to a specific topic. During the phrase mapping step, the system not only checks if there is a match with one of the labels of the KG like in its vanilla version, but it also computes a ranked list of phrases which are semantically similar to the original one. Therefore, the system substitutes in an iterative fashion the phrase in the question with the ones retrieved

[2]http://www.cs.utexas.edu/users/ml/geo.html

[3]https://www.imdb.com/interfaces/

using W2V until the right one is found. Since the word "writer" has a high semantic similarity with the word "author", using our methodology we can easily retrieve the correct answer.

A second problem occurs when the automaton enters a deadlock state. This happens when a token is misinterpreted, i.e. the automaton applies a wrong transition rule and shifts into a state where no other rules can be fired. For example, let us consider the question *Which are the prizes of Albert Einstein?*. After recognizing the starting phrase "Which are the", the automaton shifts in a state where it can accept an entity or a class. The word "prize" is erroneously matched by the system to the class `dbr:Prize` and so the automaton proceeds in the following state where, however, it can not accept an entity such as "Albert Einstein". For this reason, the procedure is forced to stop, returning as overall output an empty result set. To prevent this behavior, we introduced a backtracking algorithm that, in combination with the semantic matching mechanism described above, allows the automaton to reconsider the previous choices thus leading to the correct resource which is `dbr:award`.

2.2 Processing Italian Sentences

The main problem to deal with in order to adapt this kind of solution for a different language, like Italian, is to modify the automaton since it is designed specifically keeping in mind the English grammar. For example, the English automaton was not able to recognize a question not beginning with a "question start" token, e.g.: *Give me the*, *Who is the*, *Is*, *Are*, while this syntactic structure is relatively common in Italian. To overcome this problem, we modified the transition rules related to the state S0 so that there is a transition to the state S1 either if a question token is recognized or if the first token represents an entity. In this way, we are capable to answer to question like *Matrix è un film? (Is Matrix a film?)* or *L'ordine #1123 è in stato concluso? (Order #1123 is in Finished state?)*.

Another important difference between the syntactic structure of English and Italian sentences regards the positioning of adjectives: in English adjectives are usually placed before the noun they refer to, while in Italian they can appear also after the noun. In order to handle both these configurations, we added to the automaton another transition which allows it to shift from the state S_2 to the state S_1 if the incoming token is a property. This allowed a correct recognition of requests like: *Dammi tutti i film in lingua inglese. (Give me all the English films).* In Figure 1 is shown the updated version of the automaton i.e. capable to process sentences written in Italian.

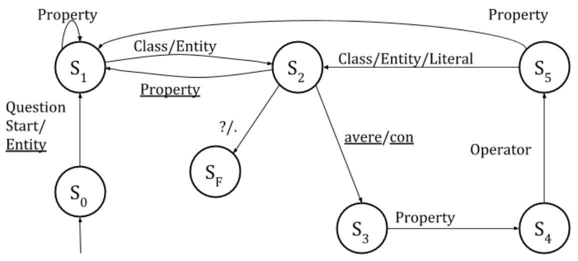


Figure 1: Structure of the automaton for the Italian language. The underlined text indicates the transition rules specifically added for the Italian language.

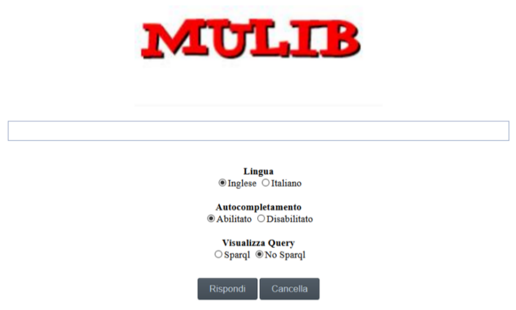
2.3 Mapping databases

One of the main features of MULIB is its capability to query not only Knowledge Graphs but also relational databases. In order to make a database compliant with the structure of the finite state automaton, we employ a particular framework called D2RQ (Bizer and Seaborne, 2004) which is developed under the Apache License [4]. This tool is essential for our system since the database, once converted using RDF can be queried both in English and Italian.

To generate an RDF graph compatible with MULIB, it is necessary to create the first mapping by using D2RQ and then modify it using its Mapping Language[5]. For examples, new labels can be defined by simply using the properties *d2rq:classDefinitionLabel* and *d2rq:propertyDefinitionLabel* for classes and properties respectively.

To express a join using D2RQ, it is necessary to create an object of type *d2rq:PropertyBridge* which allows creating a mapping between one or more database columns and a custom RDF property.

[4]http://www.apache.org/licenses/LICENSE-2.0.html
[5]http://d2rq.org/d2rq-language



Lingua
[✓ Inglese] Italiano

Autocompletamento
[✓ Abilitato] Disabilitato

Visualizza Query
Sparql [✓ No Sparql]

Figure 2: MULIB web interface.

2.4 Web Interface

We developed a Web Interface in order to allow users to interact with it and test the system in a real-world scenario (details about this experiment will be discussed in Section 3). A screenshot of the actual interface is shown in Figure 2.

We decided to design an interface as simple as possible in order to not insert elements which could confuse the users and make the interaction with the system unnecessarily difficult. The interface is composed of a text box, where the user can insert her questions and a list of options.

Since MULIB is a multilingual system, one of the options allows the user to switch from English to Italian. The system leaves to the user also the possibility to disable the auto-completion mechanism and freely insert a question without any suggestion. In this case, the system will bridge the lexical gap existing between the question inserted by the user and the database using W2V and the backtracking mechanism. Finally, the last option can enable the visualization of the SPARQL query which translates the question along with the final answer.

3 Evaluation

As stated in Section 1, in the literature there is a lack of resources for non-English languages which makes the creation and evaluation of novel approaches troublesome. It is very hard to create a solution completely language independent which allows achieving good results and NLP tools for English usually perform better than the others.

For the evaluation of our approach, we conducted an in-vivo experiment involving Sud Sistemi srl, a company that has expressed its willingness to participate in the experiment. The company made available one of its databases to be integrated and queried by MULIB. In this way, we could actually test the effectiveness of MULIB in a real-world scenario. Only the tables useful for the purposes of the experiment were used in the mapping, namely: Personal data, Articles, Agents. In the conversion, some fields were omitted, due to the sensitive data contained or to their limited significance with the purposes of the experiment.

The in-vivo experiment involved a total of 25 subjects. Participants were selected accordingly to their degree of knowledge with SQL so that the ratio between expert and non-expert user would be balanced. The experiment was composed of the following four phases:

- Phase 1: gathering personal information, i.e.: age and gender;
- Phase 2: gathering information about the participant's skills in IT and SQL;
- Phase 3: participants are asked to interact with the system and complete some simple tasks;
- Phase 4: survey about the system, to collect feedback coming from the participants.

From the second phase of the experiment emerged that the 52% of the participants declared that they had low-mid IT skills and the 48% of them declared having none or little knowledge of SQL.

During phase 4, we asked the participants to express their overall opinion about the system using a 10 point Likert scale, which ranged from a minimum of 1, that expressed the lowest liking, to a maximum of 10. The 80% of the participant assigned a score greater than five, thus corroborating the effectiveness of MULIB as a Natural Language Interface.

The usage of MULIB's web interface has been considered easy to use by the 76% of the participants, while the remaining 20% of them judged it of mid/high difficulty. This result underlines how the simplicity of the User Interface that we designed for MULIB has been appreciated by the participants. In particular, what has been judged positively by the users is the auto-completion interface, which can guide them through the interaction with the system and allows to reduce the number of mistakes.

We asked the users to select a preference between SQL and Natural Language when querying the database after the interaction with the system. The majority of users expressed their preference for the natural language. This result is surely influenced by the presence among the participants of several users that have never used SQL, thus feeling more confident in using natural language rather than a DQL.

Another question asked if it was easy to perform the SQL *join* operation using natural language. The answer was affirmative in 89.5% of cases. In fact, thanks to D2RQ, a join is mapped to a simple property and make a question over a table which represents a join does not represent a problem. Of course, this flexibility can be obtained only by means of a careful mapping of the database structure to the final ontology.

The last set of questions was used to estimate to which extent MULIB could be useful within the context of a company. The 84% of participants think that a system like MULIB could actually be helpful and beneficial in such contexts, allowing to non-expert people to query the database without the need of knowing its underlying structure.

Finally, we asked the people involved in the experiment if MULIB managed to satisfy their information need and their expectations. In the case of a negative answer, we also proposed them to give us suggestions to improve the system. The 80% of participants declared that on average the system was able to satisfy their information need, while the remaining 20% was not completely satisfied and the main causes were the following: absence of data due to the General Data Protection Regulation, lack of aggregate data in the database, and failures caused by too complex queries.

Regarding the suggestions, they can be summarized in three main points: enhance the answer to the query with other details, make the system more flexible (i.e. extending the range of questions that the system can answer), and finally improve the User Interface of the system.

4 Results and Conclusions

From the answers to the questionnaires, it is clear that MULIB has been perceived positively by the users, which think that it would represent a powerful tool to support their interaction with a DBMS.

As future work, we could improve the graphical interface of our system, making it more appealing for the users and integrating some visualization tools which could help to provide a more complete answer by integrating complementary information coming from the database.

In conclusion, in this paper, we have presented MULIB, a QA system for Structured Data which is capable to answer questions formulated in English and Italian. We decided to adopt an approach based on Controlled Natural Languages, i.e. the one adopted in systems like CANaLI. By the analysis of the shortcomings of this approach, we designed a specific solution aimed at overcoming them.

First of all we adopted distributional semantics principles in order to cope with the lexical gap and we modified the algorithm to cover the issue represented by ambiguous words. Next we extended the approach to cover also the Italian language and allow to query databases as well as Knowledge Graphs.

By performing an in-vivo experiment along with 25 participants, we could actually evaluate how helpful user perceive our system.

Acknowledgment

This work is partially funded by the “DECiSION Data-drivEn Customer Service InnovatiON” project, POR Puglia FESR 2014-2020 - INNONET-WORK program - “Sostegno alle attività di R&S per lo sviluppo di nuove tecnologie sostenibili, di nuovi prodotti e servizi”.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Christian Bizer and Andy Seaborne. 2004. D2rq-treating non-rdf databases as virtual rdf graphs. In *Proceedings of the 3rd international semantic web conference (ISWC2004)*, volume 2004. Proceedings of ISWC2004.

Maurizio Bombara, Davide Calì, Ivana Calì, and Giuseppe Tropea. 2005. Servizi innovativi web gis: impiego di full (fuzzy logic and language) per l’accesso in linguaggio naturale ai db geografici. In *Proceedings of the 9th national conference of the Italian Federation of Scientific Associations for Territorial and Environmental Information (ASITA2005)*. Proceedings of ASITA2005.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Dennis Diefenbach, Kamal Singh, and Pierre Maret. 2017. Wdaqua-core0: A question answering component for the research community. In *Semantic Web Evaluation Challenge*, pages 84–89. Springer.

Giuseppe M Mazzeo and Carlo Zaniolo. 2016. Answering controlled natural language questions on rdf knowledge bases. In *EDBT*, pages 608–611.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Christina Unger, Axel-Cyrille Ngonga Ngomo, and Elena Cabrio. 2016. 6th open challenge on question answering over linked data (qald-6). In *Semantic Web Evaluation Challenge*, pages 171–177. Springer.