IADIS International Journal on WWW/Internet
Vol. 15, No. 1, pp. 80-95
ISSN: 1645-7641

DBPEDIA BASED FACTUAL QUESTIONS ANSWERING SYSTEM

Maksym Ketsmur[1], Mário Rodrigues[2] and António Teixeira[1]
[1]*Department of Electronics Telecommunications and Informatics / IEETA, University of Aveiro. Portugal*
[2]*Águeda School of Technology and Management / IEETA, University of Aveiro. Portugal*

ABSTRACT

The creation of generic natural language query and answering (QA) systems is an active goal of the Semantic Web since it would allow people to conduct any query using their native language. Current solutions already handle factual questions mostly in English. The aim of this work was to develop a QA system to query knowledge bases (KB) such as DBpedia, in a first version, using factual questions in Portuguese, and in a second version, factual questions in English, French and German. This involves representing queries in terms of the KB ontology using SPARQL. The process of constructing a SPARQL query representing the natural language input involves determining: (1) the type of answer that is being sought - a person, a place, etc. - which is done by looking at the *wh*-words of *wh*-questions; (2) the main topic of the question - which person, place, etc. - obtained by morphosyntactic analysis to discover the potential subjects of the question; and (3) the properties that can be mapped to the KB ontology for creating a SPARQL query as precise as possible.
The first version of the system, working for Portuguese, was tested by with 22 questions without guarantee that the answer was in the KB, and the multilingual version was tested with 30 random questions from QALD 7 (Question Answering over Linked Data) training set. The correctness of the answers was verified as well if the answer exists in the KB when the system did not produced results. A correct answer was produced for 67% of the questions for the Portuguese version and up to 55% (for English) of times for multi-language version considering that the answer existed in the KB. Results show that this approach is promising and further investigation should be done to improve it. The robustness observed, and capability to handle several languages, fosters future work to expand the system to answer questions of other types.

KEYWORDS

Question Answering, Multi-lingual, Knowledge Base

1. INTRODUCTION

The predominant form of search on the internet is the use of keywords. However, the use of this form of information search is often inadvertent to express the true intention of the user (Song, D. et al., 2015). The traditional keyword searching returns an extensive list of possible results, where the user must spend some time to find the desired information. On one hand, a greater amount of information allows users to compare results and convey some conclusion, yet, it makes the search more difficult due to the quantity and growth of online information nowadays. So, systems that allow anyone to perform structured search become increasingly needed (Hirschman, L. et al., 2001; Navigli, R. et al., 2012).

Using a well-structured search allows to better infer users' intent, returning the desired answer and not a list of documents as in traditional approaches. With this, a need to develop semantic search systems arise. Such systems are intended to allow users carrying out structured searches using natural language, and should identify users' intentions, generating formal semantic representations that are able to combine distinct sources of information to obtain an answer. Increasingly, the information is provided in the form of Linked Data, which consists of using the Web to interconnect different sources of information. These can be as diverse as the databases of two distinct organizations in different geographic locations. Technically, Linked Data refers to data published on the Internet in a way that can be understood by both humans and computers (Damljanovic, D. et al., 2011; Unger, C. et al., 2012).

Semantic web data is published in Resource Description Framework (RDF) form, which is a standard for expressing data graphs and sharing them with other people and, perhaps more importantly, with machines. A large collection of tools and services emerged around the RDF (Bizer, C. et al., 2009; Segaran, T. et al., 2009). RDF is a language to express data models using triples, which are constituted by subject, predicate, and object. In addition, it adds several important concepts that make these models more accurate and robust, removing the ambiguity when transmitting semantic data between machines (Segaran, T. et al., 2009).

The main goal of this study is to develop two versions of a QA system that allows users to get the desired information using natural language as a query. The characteristic points of this system are the use of Portuguese language in the first version, being expanded to handle English, French, and German languages in the second version, and being designed to operate directly over already existing knowledge bases (KB), being for the moment instantiated in the popular DBpedia.

The paper continues in section two with the background and the relevant related work. The third section explains how the two version of QA system works and the fourth section shows representative results of each version and discusses them. The paper ends with the conclusions in section 5.

2. BACKGROUND AND RELATED WORK

The first step in the development of QA systems is to understand how users search information and what kind of answer they expect (Hirschman, L. et al., 2001). Different types of questions can be asked and can be identified per the type of answer: factual, opinion, and abstract (Hirschman, L. et al., 2001). The different types of questions can be done in different

ways, such as indirect requests or even commands. "*I would like to list all presidents of Portugal*" is considered as an indirect request and "*Name all presidents of Portugal*" is considered as a command (Hirschman, L. et al., 2001). However, this can bring some difficulties to systems that heavily depend on identifying terms such as *Who*, *Where*, *When* and *What* in users' queries. For example, by identifying the word *Who* in the query, the system will determinate that the user wants to know about some human or organization, or the word *Where* that will indicate that the answer will be about some location. These types of systems cannot respond to the questions asked in indirect request form or even commands. Also, the user is not expected to perform search in a structured way, which in this case makes the autosuggest a good feature during the construction of the query, as seen in the TR Discover system.

Not all types of questions are similar, there is evidence that some types are more difficult than others, such as *Why* and *How* tend to be more difficult to analyze because they require understanding of text and relations between entities (Hirschman, L. et al., 2001). The identification of the correct expected answer type reduces the number of possible answers, making the system more efficient and precise (Breck, E. et al., 2000). Different systems use distinct approaches to form the answer that is shown to the user. Answers can be formed in two ways, extraction or generation. In the first, the fragment that contains the answer is extracted from the original document and presented to user, while in the second the answer is extracted from multiple sentences or from several documents (Hirschman, L. et al., 2001).

The ability of identifying users' intention implies a natural language analysis where the main challenge is to understand the users' intent that may be ambiguous. In addition, natural language processing aims to achieve a natural language analysis like humans, using different techniques that allow to identify entities, actions, relation between entities, etc. in a natural language sentence (Hirschman, L. et al., 2001; E. Liddy. et al., 2001). Many evaluations of semantic search systems referred that users prefer to use free natural language during the search rather than controlled inputs or view-based searches. Although the flexibility offered by this approach is a significant advantage, it can also become very complex (Kumar, A. et al., 2016). Allowing users full freedom in choosing terms increases the difficulty of these tools to disambiguate the search and understand the users' intent (Elbedweihy, K. et al., 2013).

Word sense disambiguation aims to understand what the terms mean. While for humans it is easy to understand the meaning of a word, for a machine it is a difficult task (Guha, R., et al., 2003; Lopes, L.S. et al. 2005; Elbedweihy, K. et al., 2013). Therefore, the use of semantic search tools is essential to match the document content with users' intention (Hoffart, J. et al., 2011). Semantic search systems adopt different search approaches, ranging from natural language (free or guided) to visualization-based interfaces (forms or graphs). Each of these strategies provides different levels of user flexibility, query language expression and support during the query formulation (Elbedweihy, K. et al., 2013).

However, as mentioned earlier, flexibility makes it difficult to map terms with concepts, properties and ontological entities. One of these difficulties is polysemy (a word with more than one meaning) and synonymy (multiple words with the same meaning). While the first affects accuracy by providing false correspondences, the second affects recall by causing the lack of true semantic correspondences. In this way, both use the word disambiguation techniques (Elbedweihy, K. et al., 2013). For example, to answer the question "*How tall is ...?*", the query term *tall* needs to be mapped to the *height* property in the ontology. However, the term *tall* is polysemous and has different meanings including (from WordNet): "*Great in vertical dimension*", "*High in stature*", "*Tall people*", "*Tall buildings*", etc. (Elbedweihy, K. et

al., 2013). In this way, the term must be disambiguated, and the correct meaning identified before gathering the related terms, among which there may be named entities, which must also be disambiguated in case multiple matches are found.

Furthermore, as in the example of the term *tall*, named entities can refer to several real-world entities, requiring disambiguation, which is usually done using the query context and the structure of the ontology in which the term occurs. That is, to identify the meaning of the term, is necessary to consider the phrase in which it occurs and the structure of the ontology. For example, in the question "*What river does the Brooklyn Bridge cross?*" the terms *Brooklyn Bridge* and *Brooklyn* would be mapped to the resources of DBpedia, *res:Brooklyn_Bridge* describing the bridge and *res:Brooklyn* describing the city in New York, respectively. The first resource can be correctly selected based on an ontological structure that shows that crossings of properties connect a river to a bridge and not to a city (Elbedweihy, K. et al., 2013). In fact, identifying the correct meaning of a polysemic word is necessary for the process of query expansion, which is often used by search systems to find matches between search terms and ontology (Elbedweihy, K. et al., 2013).

Some approaches consider all meanings of a polysemic word and use their related terms for query expansion. However, this increase the noise and irrelevant matches, affecting accuracy. On the other hand, in the disambiguation approach described by (Lopez, V. et al., 2006), a specific synset of WordNet (list of synonyms) is considered relevant only if one of its meanings exists in the synonyms, hyperonyms, hyponyms, holonyms or meronyms of an antecedent or a descendant of synset (Elbedweihy, K. et al., 2013). Typically, in these systems, only lemmatization is performed, which consists of a morphological transformation that modifies the word for its base form or dictionary form, the lemma (Liu, H. et al., 2012), and part-of-speech (POS) tagging, whether it is a noun, an adverb, an adjective, etc. (Elbedweihy, K. et al., 2013; Berant, J. et al., 2013). After, each term is stored with its lemma and POS tag, except previously recognized named entities that are not lemmatized. In addition, the position of each term is stored in relation to the rest of the query (Elbedweihy, K. et al., 2013).

Semantic search systems use semantic databases that can be queried using SPARQL language, since it is the recommended language by W3C for RDF (Resource Description Framework) querying. The terms generated in the disambiguation stage of the query go through the process of matching properties and ontology concepts. After gathering all correspondences of candidate ontologies that are syntactically like a query term, they are sorted using two string comparison algorithms described by (Winkler, W., 1990; Philips, L., 2000). The first depends on comparing the number and order of common characters by assigning a high score to the terms that are parts of each. This is useful since concepts and ontology properties are usually named in this way. For example, the term *population* and the *totalPopulation* property receive a high similarity score using this algorithm (Yao, X. et al., 2014). At this stage, the query can be interpreted in terms of a set of ontology concepts, properties, and instances that need to be interconnected. The semantic data are presented in the form of triples, constituted by subject, predicate and object. In this way, given the object (the query) are the predicates and subjects, where the predicates present the relation between object and subject.

Given that first version of our system is intended for Portuguese, we have analyzed some systems working for Portuguese. The question classification done by Branco et al. (2008) is per semantic category of the right answer. The major factor that differentiates the (Branco et al., 2008) system from other systems, is the way the answer is obtained. This system acts as a

client of search engines, submitting the list of keywords obtained in previous phase and retrieving relevant documents. The most similar system to ours is the one of Quaresma, P. et al. (2004) since the semantic representation of the query is obtained using the ontology of concepts and a KB with some general world knowledge. Finally, the system IdSay (Carvalho et al., 2008), is an open domain QA system for Portuguese. Its current version can be considered as baseline version, using techniques from the Information Retrieval (IR) area.

These systems are intended to answer the factual or definition questions made in Portuguese. Although they use some identical methods such as natural language processing and question type classification, they differ from the developed system in terms of the form of answer obtaining, extracting them from the text or collection of texts, whereas our system uses the semantic database as DBpedia.

3. DEVELOPED SYSTEM

This section aims to present and describe the successive phases of analysis and processing of information that will be conducted to answer users' question.

The architecture of the developed system consists of 5 processing steps. Figure 1 presents these steps, at right, and a representation of their application to an example question "*Onde nasceu o Albert Einstein?*" (Where was Albert Einstein born?), to help better understand the processing steps description of this section.

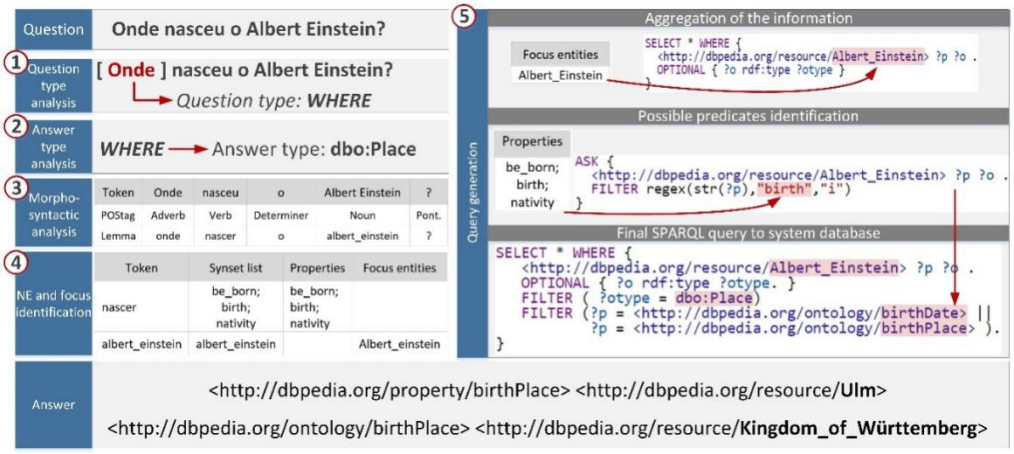

Figure 1. The question answering process exemplified with the question "Onde nasceu o Albert Einstein?" (Where was Albert Einstein born?)

Next subsections present information on the two instantiations of this architecture: a first version, for Portuguese; an enhanced version, capable of accepting questions in 4 languages (Portuguese, English, French, and German).

3.1 First Version

The first step is to identify the question type. Many different types of questions exist, among which are being considered three types: causal, list, and definition. The identification is based on word patterns made by combinations of *Who*, *When*, *Where* and *How much*.

The next step is for determining which kind of KB object classes are expected as an answer: for instance, if the question type is "*where*" then the answer expected if of class *Place* or equivalent. The answer classes are obtained using an answer type taxonomy and a set of rules mapping question types to answer classes, the same way Pasca, M. and Harabagiu, S. (2001) did. The answer class can be any class of the ontology or xml datatype. Possible and frequent top level classes and datatypes of DBpedia were identified and the answer type class set is: *Person*, *Agent*, *Place*, *Game*, *Dog_breeds*, *Eukaryote*, and *Abstract* class. Some mapping rules were defined to associate possible answer types to each question type. Some answer types have more than one ontology class or datatype, for example a question that contains word *Which*, that can be asked about anything, the mapping rule point to the list of ontology classes and/or datatypes. The ontology class is directly used in the SPARQL query generated at the end of this process to filter the possible answers and thus improve the answer accuracy.

The third step of processing is a morphosyntatic analysis of the input query phase including tokenization, lemmatization, POS tagging, number and named entity identification, and syntactic dependency parsing. One of the key features of this module is the dependency tree construction that allows to understand the relations between different words of the phrase which in turn will allow to identify the focus entity of the sentence, taking another step in identifying what is being queried.

After having the syntactic dependency tree and all input tokens characterized, the fourth processing step aims to identify all possible named entities or concepts in the input query. To perform this, BabelNet is used to obtain the synsets of the concepts present in the input phrase. This practice is common in QA systems since using other words representing the same concepts, that might exist in the KB but not necessarily in the input query, broadens the coverage of the system. The concept of synsets comes from WordNet and consists of unordered sets of cognitively synonymous words and phrases. BabelNet allows to obtain English synsets of words written in most languages, then used to query DBpedia. Querying DBpedia is only possible using English terms because its ontology is expressed in English. The focus entity is a synset that represents an entity of one of two types: (1) named like people, organizations, and cities; and (2) concepts such as light, water, star, etc. In case of multiple candidates then is selected the one closer to the dependency tree root.

The fifth step is about information aggregation and consists of constructing a first SPARQL to know the list of properties of the focus entity in DBpedia. The properties that can represent the remaining synsets identified in the input phrase will become filters in the second and final SPARQL. This final SPARQL also contains a filter to restrict the answers to the answer type ontology classes and/or datatypes identified in the second step. This query aims to retrieve the possible answers from the system's triple store and the answer is presented to user.

3.1.1 Software Tools

In our system, the most important part of processing is identifying of the user intent. For this, in Portuguese version two different Natural Language Processing tools were used, Freeling and Maltparser. FreeLing is a C++ library providing language analysis functionalities (morphological analysis, named entity detection, POS-tagging, parsing, Word Sense Disambiguation, Semantic Role Labelling, etc.) for a variety of languages among which Portuguese is available but not for syntactic dependency parsing. FreeLing also provides a command-line front-end that can be used to analyze texts and obtain the output in the desired format (XML, JSON, CoNLL). In our case the CoNLL output format is important to perform the correct output for the Maltparser.

MaltParser is a system for data-driven dependency parsing, which can be used to induce a parsing model from treebank data and to parse new data using an induced model. For the Portuguese version, the Maltparser was trained using CG-converted UD Portuguese treebank and Freeling output. The CG-converted UD Portuguese treebank is originally based on an improved and enriched version of the 7.4 dependency version of the revised Bosque part of the Floresta Sintá(c)tica treebank (cf. Linguateca.pt). The version 7.4 was created in 2006-2008 aligned with a new live run of the PALAVRAS parser to propagate morphological features from unambiguous to ambiguous words, and to add what the Floresta team called "searchables", i.e. tags for features distributed across several tokens, such as NP definiteness and complex tenses. The public treebank only used this for the constituent version, which was the one actively revised by the Floresta team until 2008 (Linguateca.pt version 8.0).

To store all aggregated information, the Jena Triple Store was used. Apache Jena (or Jena in short) is a free and open source Java framework for building Semantic Web and Linked Data applications. The framework is composed of different APIs interacting together to process RDF data. Also, the web page to manage the database and executing of SPARQL query is available.

3.1.2 Knowledge Bases

BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of semantic relations, made up of about 14 million entries, called Babel synsets. Each *BabelSynset* represents a given meaning and contains all the synonyms which express that meaning in a range of different languages.

The major advantage of this KB is aggregation of different semantic sources in one, that allow perform synsets identification that is common among different databases. As the information extraction regarding the focus entity is made by querying DBpedia, the correct synset must to be identified. In this case, was determined that DBpedia encodes concept and named entities in the same way as Wikipedia which is a part of possible BabelNet sources. In this way, extracted synsets from Wikipedia in most cases can be directly used to query DBpedia.

Additionally, the synset identification is made using Multilingual WordNet, because of multilingual input of the system. Using BabelNet allow us querying different semantic databases from the same location. DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data.

3.2 Multilingual Version

After the development and evaluating of the Portuguese version of the system, it was decided to develop an enhanced multilingual version, with the aim of demonstrating that the first version, without significant modifications, can respond to factual natural language questions made in other languages. This is because the Natural Language Processing tools and Knowledge Bases used in the first version are multilingual, making the development of the second version easier.

The purpose of the second version of the system is to respond to factual natural language questions made in 4 distinct languages, English, French, German and Portuguese, and in contrast to the base version, the questions used will be randomly selected from the QALD 7 training set.

For this purpose, the first version of the system has been analyzed to identify the modules that need to be adapted to analyze and answer questions in other languages. First, the adaptation must be done in the NLP module, since it allows to identify the relationships between the words of the question, thus resulting in the dependency tree, which allows to identify the focus entity of the question. As previously mentioned, the NLP module consists of the use of two multilingual tools, Freeling and MaltParser, which must be configured to answer questions in other languages.

The Freeling configuration is much simpler than MaltParser, since the first one consists of a set of files organized in folders, where each folder corresponds to a specific language. In this way, it's enough to load the configuration files from folder of desired language to Freeling perform its analysis. Despite this simplicity of adaptation, Freeling does not provide the same features (e.g. NER) for all languages, thus varying its inclusion or exclusion of their configuration files.

After some analysis, it was identified that the Freeling provides required features only for English and French, whereas for German, it is missing an essential module NER, which makes us use another NLP tool to identify named entities in the question. For this purpose, the OpeNER tool was selected due to its performance and simplicity of configuration.

OpeNER (Open Polarity Enhanced Named Entity Recognition) is a project funded by the European Commission under the 7th Framework Programme and its main goal is to provide a set of open and ready to use tools to perform some NLP tasks in six languages: English, Spanish, Italian, Dutch, German and French (García-Pablos. A. et.al., 2013).

OpeNER processing consists of a pipeline, receiving some input text (users' question in this case) and returning an XML file, with the tokenized text and identified named entities.

The identification of the correct focus entity depends of Freeling and MaltParser, where the second must be trained for the target language.

Despite the existence of pre-trained models for MaltParser to process the English and French languages, they were not used since the MaltParser trained with these models require a set of input data that cannot be produced by Freeling, and so it returns incorrect results. In this way, for the multilingual version, MaltParser was trained in the same way as in the first version of the system, combining the result of Freeling and UD treebank of target language to create the training set, which guarantees the desired results. The table 1 shows what adaptations have been made and in which modules.

Table 1. Modules adaptation

Language	Module	Work done	Resources used
English	Question type Analysis	- Adapted to identify Wh words like Who, Where, etc.	
	Morphosyntactic analysis	- Adapt Freeling to load configuration files for English language.	
	NE and Focus Identification	- Training MaltParser using UD treebank. - Changing synset identification language to English.	https://github.com/UniversalDependencies/UD_English
French	Question type Analysis type Analysis	- Adapted to identify Wh words like, Quel, Qui, etc...	
	Morphosyntactic analysis	- Adapt Freeling to load configuration files for French language.	
	NE and Focus Identification	- Training MaltParser using UD treebank. - Changing synset identification language to French.	https://github.com/UniversalDependencies/UD_French
German	Question type Analysis	- Adapted to identify Wh words like, Was, Welche, etc...	
	Morphosyntactic analysis	- Adapt Freeling to load configuration files for German language.	
	NE and Focus Identification	- Configuring OpeNER and extraction Named Entities. - Training MaltParser using UD treebank. - Changing synset identification language to English.	https://github.com/UniversalDependencies/UD_German

4 EVALUATION ON

In this section are presented the evaluation methods for each version of the system, the obtained result, its analysis and the final discussion.

4.1 Method

The Portuguese version of the system was evaluated using a set of factoid questions as for now it was not possible to find a golden collection for Portuguese Q&A systems based on an existing knowledge base. An effort was made to cover different domains to allow understanding which question is the easiest to answer.

For the evaluation of the multilingual system, questions were selected from the training set of the *QALD2017 CHALLENGE Task 1: Multilingual question answering over DBpedia*. The *Question Answering over Linked Data* (QALD) challenge aims at providing an up-to-date benchmark for assessing and comparing state-of-the-art-systems that mediate between a user, expressing his or her information need in natural language, and RDF data.

Thus, for the evaluation of the multilingual system, 30 factual questions in different languages (English, French and German) were randomly selected using the online list randomizer (http://www.randomlists.com/) from 136 extracted questions from QALD training

set. Given that, the training set of *task 1* does not provide the questions in Portuguese language, they have been translated manually. As such, this process of questions choosing for the evaluation resulted in 4 sets of 30 questions, where each set corresponds to each language.

4.2 Results for Portuguese Version

The system could correctly answer 10 questions (45% of the 22 evaluation questions). Analyzing the questions with not answered by the system was found that answers to 7 questions do not exist in the DBpedia and Named Entities were not identified by Freeling in 4 questions. Taking this into account, in general, the system achieved 67% correct answers (10 of 15 questions with answer in DBpedia), and thus it performs a good analysis of users' question.

4.3 Discussion

The failures that have been presented are in general the lack of information which implies a lack of response. Given that the data source chosen for extracting information, DBpedia, contains a lot of information but it is not persistent in all entities as seen in the example of the *Mediterranean Sea* and *Atlantic Ocean*, where first entity had *depth* property and other not.

Another type of failure corresponds to the meaning of properties that the DBpedia uses. It appears, for example. In the question "*Quantos golos marcou o Cristiano Ronaldo?*" (How many goals did Ronaldo score?). In this case, the answer consists of a set of properties which in general have no precise meaning. The answer is: *Goals* – 84; *Nationalgoals* – 7; *Nationalgoals* – 1; *Nationalgoals* – 5; *Goals* – 248. In this case is only possible to conclude that the sum of all goals of Cristiano Ronaldo is 248. Probably the property *nationalgoals* indicates the total goals when playing for Portugal, but in this case, we have 3 different objects to the same property, which makes difficult to understand what is the meaning of each.

The identification of synsets also presented some flaws. For some questions, the amount of synsets is too small to make possible to obtain the answer. To provide more insight in the capabilities of the system, will be analyzed some of the questions and the respective answers.

The question "*Qual é a altura do Michael Phelps?*" *(How tall is Michael Phelps?)* has a unique answer, *1.83* according to DBpedia. This type of question is simple to analyze and information about the focus entity *Michael Phelps* is available. For the property *altura* (height) there were two synsets, *pitch* and *height*. The process of identifying of possible predicate determinates the ontology property *height* which corresponds to the answer.

The answer to "*Qual a profundidade do Mar Mediterrâneo?"* (How deep is the Mediterranean Sea?) is like the answer about Cristiano Ronaldo, where a list was returned. But in this case the properties have a compressive meaning for user which not only correctly answers the question returning the maximum depth of 5267 meters as also return additional information of average depth.

For question "*Qual a profundidade do Oceano Atlântico?*" (How deep is the Atlantic Ocean?) the correct entity and its properties were successful identified, but the property that contains *depth* does not exist in the list of predicates for this property. To find the correct answer, the DBpedia page of *Atlantic_Ocean* was analyzed. Every subject on DBpedia contains the property *sameAs* that corresponds to this entity in our sources. In the list of *sameAs* result, the page of *Atlantic_Ocean* on Wikidata was found. There the property *deepest_point* was identified and possibly is the correct answer. The problem there is the use

of expansion methods to find out the answer, using the *sameAs* property. But even using this expansion method, the *deepest_point* property would not be identified using the *depth* property. Thus, in the answer to this question there was a main problem, no existence of the same properties for different entities, that is, whereas in the previous question *depth* property was identified, for *Atlantic Ocean* no, being that the system correctly identified all the necessary information to get the answer.

In question "*Qual é a velocidade da luz?*" (What is the speed of light?) there are two nouns *velocidade* (velocity) and *luz* (light). What is important here is to identify the true focus entity, which is only possible by analyzing the dependency tree. The focus entity *light* was correctly identified, while the synsets for the *velocity*, being a noun were obtained from Wikipedia. This identification of synsets from Wikipedia only returned a *velocity* synset, which is considered a concept. This identification is considered good since Wikipedia identification is only carried out on Named Entities or Concepts. In this case, *velocity* could also be obtained from WordNet, which in addition to v*elocity* identified another synset s*peed*. The greater problem in answering this question is the absence of such information in the data source, present in most cases.

The problem identified in question "*Quem foi a mulher de Napoleão Bonaparte?*" (Who was the wife of Napoleon Bonaparte?) was the inconsistency of the semantic representation of the entity *Napoleon Bonaparte*. Seen to be a noun, the synsets were obtained from Wikipedia by getting synset *Napoleão_Bonaparte* being a Portuguese representation. As mentioned before, DBpedia uses in most cases the same representation of Named Entities and Concepts as Wikipedia, which is considered a true statement. Thus, the problem consisted of misidentification of the synset. The synset for *Napoleão Bonaparte* corresponds to the Portuguese representation *Napoleão_Bonaparde*, while the English representation in DBpedia is *Napoleon*.

4.4 Results for Multilingual Version

The information on the correct answers obtained for each language is summarized in Table 2. Besides the number of correct answers, 2 percentages are given: first considering all questions, second considering the number of questions with answer in DBpedia (only 20).

Table 2. Results for Multilingual version

Language	Number of correct answers	Percentage correct answers (in 30 questions)	Percentage correct answers (in the XX questions with answer
English	11	37%	55%
French	7	23%	35%
German	7	23%	35%
Portuguese	6	20%	30%

From the table 2 is clear that exist some difference in the performance obtained for the several languages, being the best results obtained for English (55% correct answers when information exists in DBpedia) and the worst for Portuguese (only 30%), almost half of the correct answers obtained for English.

Investigating from where in the process comes the failures was the subject of the second analysis, presented in a graphical form in Figures 2 and 3. Figure 2 presents, in a radar-like graph, the number of correct results for the several steps of the processing (Answer type determination, determination of Focus, determination of synset, and identification of Entities).

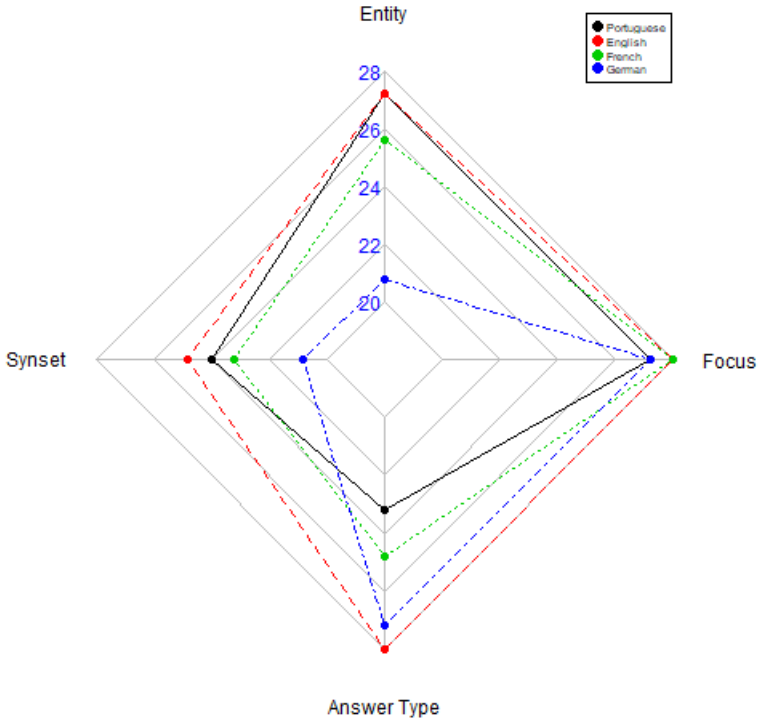

Figure 2. Number of correct outputs for several of the steps of the system, as function of the language

Figure 2 show that: (1) there are steps with much more noticeable language differences than others (ex: Focus much more language independent than Answer Type); (2) there is more dispersion of results for Entity identification (from 21 to 27 correct results); (3) Answer type determination was a big problem for languages such as Portuguese, while having almost no errors for English; and (4) synset identification is similar for English, French and Portuguese.

Figure 3 presents the same information of the previous figure, but now expressing the errors as percentage of the total errors considering all languages. It shows that there are steps that affect particularly certain languages (ex: Focus is problematic for Portuguese and German).

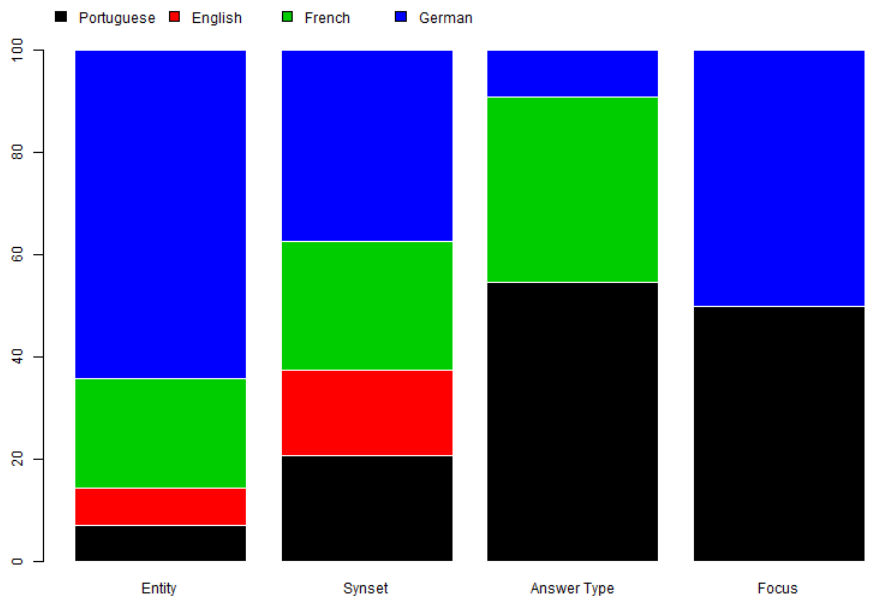

Figure 3. Information on the distribution of errors by language in several steps of the system. The errors of each language were normalized by the total number of errors at each step, considering all languages

4.5 Discussion

As seen, the multilingual version of the system can respond to natural language questions in different languages. The system has shown better results by analyzing the questions in English, because most of them have common named entities for English and natural language processing tools are more optimized for this language. Figure 2 shows that the evaluation results using questions in Portuguese were very similar to the results of English in some aspects (Synset, Entity and Focus), but often failed to identify the correct answer type, thus not obtaining a final response demonstrating poor results in general.

After the evaluation and detailed analysis of the system, two main problems were identified, lack of information on DBpedia and complexity of questions, being that the first problem already persisted in the first version of the system, while the second one appeared due to the questions with more than one Named Entity, which requires to identify the relationships between them, to what the system was not originally intended. In addition, there are some issues in the identification of Named Entities, more frequent in the German language, for what an OpeNER tool was used, since Freeling provides a few features for German.

5 CONCLUSION AND FUTURE WORK

In this study were presented two versions of a system that aims to answer the questions made in Portuguese natural language, in a first version, and English, French, German and Portuguese in an enhanced version. A study was carried out identifying possible modules and forms of information processing. Most systems of this type are intended for English, and the tools used in these systems are not always multilingual. Considering this, the part of natural language analysis with the purpose of identifying the users' intention was made using Freeling and MaltParser.

The use of BabelNet was also not a common choice for synset identification, but it was a good choice since this dictionary aggregates information from different data sources. Thus, it was not necessary to carry out the communication independently with WordNet, Wikipedia and others. The step about determining the focus entity using the DBpedia data source is common among systems of this type, also having a SPARQL endpoint making this process simple and fast. One of the greatest difficulties was the identification of the true intention of the user as well as the identification of the meaning of words, that is, the identification of synsets. There were cases where synsets could not always be obtained in English for a word in Portuguese language, being an easier task in systems of the English genre.

One advantage of this system is to be adaptable to other languages without significant changes, since the NLP tools and BabelNet are multilingual and DBpedia provides information in 125 different languages. However, at this moment the system has some limitations related with the difficulty of identifying some Named Entities in questions made in German language, as well as the lack of information on DBpedia, which in some cases can be a problem, depending on the question's focus entity.

The system presented good results in answering the questions which had an answer in the source of information (DBpedia in our case), returning concrete results or a list of results if it was present. These lists, when found, were much smaller than the lists of results returned using the common keyword search.

5.1 Future Work

Results show that it is necessary to improve the identification of named entities. Some possible future additions include, after the question input, the system can interact with users for verifying if Named Entities were correctly identified, and use this information for module training. Likewise, users feedback can be collected after the answer reply, indicating if it was a correct answer or not. In literature is possible to find studies that store the answers to the questions when they were right. So, when the question is asked, the search is done first in the already given answers and only later in the database of the system. This solution could be useful for the developed system, thus reducing waiting time for complete processing of question.

Some problems exist in the identification of the correct synsets for Portuguese words, which in some cases could only be obtain in Portuguese. One of the solutions could be to translate beforehand Portuguese to English and then perform synset identification. Finally, in addition to the brief answers to questions, the system could become more expressive providing some additional information or a more extensive response, which involves the use of information extraction techniques that was not used in this study.

ACKNOWLEDGEMENT

This Research was partially funded by National Funds through the FCT - Foundation for Science and Technology funding of IEETA research unit, in the context of the project UID/CEC/00127/2013.

REFERENCES

Berant, J., Chou, A., Frostig, R., & Liang, P., 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In EMNLP (Vol. 2, No. 5, p. 6).

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S., 2009. DBpedia-A crystallization point for the Web of Data. Web Semantics: science, services and agents on the world wide web, 7(3), 154-165.

Branco, A., Rodrigues, L., Silva, J., & Silveira, S., 2008. Xisquê: An online qa service for portuguese. In PROPOR (Vol. 8, pp. 232-235).

Breck, E., Burger, J. D., Ferro, L., Hirschman, L., House, D., Light, M., & Mani, I., 2000. How to evaluate your question answering system every day and still get real work done. arXiv preprint cs/0004008.

Carvalho, G., De Matos, D. M., & Rocio, V. (2008). Idsay: Question answering for portuguese. In Workshop of the Cross-Language Evaluation Forum for European Languages (pp. 345-352). Springer, Berlin, Heidelberg.

Damljanovic, D., Agatonovic, M., & Cunningham, H., 2011. FREyA: An Interactive Way of Querying Linked Data Using Natural Language. In ESWC Workshops (Vol. 7117, pp. 125-138).

Elbedweihy, K., Wrigley, S. N., Ciravegna, F., & Zhang, Z., 2013. Using BabelNet in bridging the gap between natural language queries and linked data concepts. Proc. 2013 Intern. Conf. on NLP & DBpedia. CEUR-WS. org.

García-Pablos, A., Cuadros, M., Gaines, S., & Rigau, G., 2013. OpeNER demo: Open Polarity Enhanced Named Entity Recognition. In Come Hack with OpeNER! Workshop Programme (Vol. 501, p. 12).

Guha, R., McCool, R., & Miller, E., 2003. Semantic search. In Proceedings of the 12th international conference on World Wide Web (pp. 700-709). ACM.

Hirschman, L. and Gaizauskas, R., 2001. Natural language question answering: the view from here. Natural Language Engineering, 7(4), 275-300.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., & Weikum, G., 2011. Robust disambiguation of named entities in text. Proc. of Conf. Empirical Methods in NLP. ACL.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R., 2016. Ask me anything: Dynamic memory networks for natural language processing. Proc International Conference on Machine Learning (pp. 1378-1387).

Liddy, E.D., 2001. Natural Language Processing in Encyclopedia of Library and Information Science, 2nd Ed, New York: Marcel Decker Inc.

Liu, H., Christiansen, T., Baumgartner, W. A., & Verspoor, K., 2012. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. Journal of biomedical semantics, 3(1), 3.

Lopes, L.S., Teixeira, A.J., Quinderé, M., and Rodrigues M., 2015, From Robust Spoken Language Understanding to Knowledge Acquisition and Management. Proc. of 9th European Conf. on Speech Communication and Technology.

Lopez, V. et al., 2006. Poweraqua: Fishing the semantic web. The Semantic Web: research and applications, 393-410.

Navigli, R., and Ponzetto, S., 2012. Multilingual WSD with just a few lines of code: the BabelNet API. In Proceedings of the ACL 2012 System Demonstrations (pp. 67-72). Association for Computational Linguistics.

Pasca, M., & Harabagiu, S., 2001. High performance question/answering. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 366-374). ACM.

Philips, L., 2000. The double metaphone search algorithm. C/C++ users journal, 18(6), 38-43.

Quaresma, P., Quintano, L., Rodrigues, I., Saias, J., & Salgueiro, P. (2004). University of Évora in QA@ CLEF-2004. In Workshop of the Cross-Lang Eval Forum for European Languages (pp. 534-543). Springer, Berlin, Heidelberg.

Segaran, T., Evans, C., & Taylor, J., 2009. Programming the Semantic Web: Build Flexible Applications with Graph Data. " O'Reilly Media, Inc.".

Song, D., Schilder, F., Smiley, C., Brew, C., Zielund, T., Bretz, H., Martin, R., Dale, C., Duprey, J., Miller, T., & Harrison, J., 2015. TR Discover: A natural language interface for querying and analyzing interlinked datasets. In International Semantic Web Conference (pp. 21-37). Springer, Cham.

Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A. C., Gerber, D., & Cimiano, P., 2012. Template-based question answering over RDF data. In Proc. of the 21st international conference on World Wide Web (pp. 639-648). ACM.

Winkler, W., 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.

Yao, X. and Van Durme., 2014. Information Extraction over Structured Data: Question Answering with Freebase. In ACL (1) (pp. 956-966).