Using a Dialogue Manager to Improve Search in the Semantic Web

Dora Melo[1], Irene Pimenta Rodrigues[2], and Vitor Beires Nogueira[2]

[1] Instituto Politécnico de Coimbra and CENTRIA, Portugal
dmelo@iscac.pt,
[2] Universidade de Évora and CENTRIA, Portugal
{ipr,vbn}@di.uevora.pt

Abstract. Question-Answering systems that resort to the Semantic Web as a knowledge base can go well beyond the usual matching words in documents and, preferably, find a precise answer, without requiring user help to interpret the documents returned. In this paper, we introduce a Dialogue Manager that by analysing the question and the type of expected answer, provides accurate answers to questions posed in Natural Language. The Dialogue Manager not only represents the semantics of the questions, but also the structure of the discourse including the user intentions and the questions context, adding the ability to deal with multiple answers and providing justified answers. Our system performance is evaluated by comparing with similar question answering systems. Although the test suite has slight dimensions, the results obtained are very promising.

1 Introduction

Question-Answering (QA) systems provide a concise answer to questions posed by the user in Natural Language, in his own terminology [8]. In order to enhance communication and cooperation between system and users, the answers provided by the system, when appropriate, should also be informative, complete and justified.

Consistent with the role of ontologies in structuring and organizing semantic information on the web, QA systems based on ontologies allow exploring the expressive power of ontologies and enriching queries interpretation. Ontologies and the Semantic Web (SW) [9] have became formalisms able to represent the conceptual domains of knowledge and promote the capabilities of Question Answering systems based on semantics [7].

In this paper, we introduce a Dialogue Manager that by analysing the question and the type of expected answer, provides accurate answers to the questions posed by the user in Natural Language (NL) (currently, only the English language). The Dialogue Manager not only represents the semantics of the questions, but also the structure of the discourse that includes the intentions of the user and the questions context, allowing this way to deal with multiple answers and to justifying those answers. The Dialogue Manager resorts to ontologies, OWL2 descriptions and other web resources such as DBpedia [1] and WordNet [5].

The proposed Dialogue Manager is a component of a Cooperative QA system for Ontologies OWL2 presented in [20,19], that receives questions expressed in Natural Language and returns a collaborative answer, also expressed in Natural Language. When the Dialogue Manager (DM) has multiple choices for the answer, it starts a clarifying dialogue with the user. Our goal is to provide a tool that is independent of prior knowledge of semantic resources by the user and answer directly and accurately to questions posed in NL.

The remaining sections of the paper are organized as follows. First, in Section 2, we present a brief overview on Ontologies and the Semantic Web. In Section 3, we present related work, highlighting the similarities and differences with our proposal. Afterwards, in Section 4, we introduce the proposed Dialogue Manager, highlighting its capabilities. Hereafter, in Section 5, we present a preliminary evaluation which boils down to a first experimental set of tests done to the system. Finally, in Section 6, we present our conclusions and elaborate about future work.

2 Ontologies and the Semantic Web

The definition *an ontology is an explicit specification of a conceptualization* was originally introduced by Tom Gruber [6] in the domain of Artificial Intelligence. The term is borrowed from philosophy, where ontology is defined as a systematic account of existence. For Artificial Intelligence systems, what "exists" is what can be represented. When the knowledge domain is defined in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects and the relations that can be established between them are expressed in the representational vocabulary, methodology used by knowledge based programs and that allows representing knowledge.

Thus, in the context of Artificial Intelligence, we can describe the ontology of a problem by defining a set of representational terms. In such ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory.

Ontologies promote and facilitate interoperability, intelligent processing, sharing and reuse of knowledge among information systems. More recently, ontologies have been recognized as an important component for building the Semantic Web [9]. This new field introduces changes in the way ontologies are built and used. The Semantic Web is not a separate web but an extension of the current one, in which information is given well-defined meaning. The Semantic Web is used to reduce the ambiguity of Natural Language, enabling the implementation of intelligent agents. The components of the Semantic Web are the expression of meanings, knowledge representation, Ontology (contextualization), Agents (reacts to environment) and the evolution of knowledge.

In the present work, ontologies are used to define, structure and fit the semantic information of the question and its terms, according to search domain, permitting to associate and contextualize terms, improving the interpretation of the question.

3 Related Work

START[3] [11] is a Natural Language Question Answering system that provides users with multimedia information access through the use of Natural Language annotations. The annotations are sentences and phrases parsers that describe the content of various information segments. The user query is compared against the annotations stored in the knowledge base and when a match is found, the corresponding segment is returned as answer.

PANTO [27] is a portable Natural Language interface to ontologies, that accepts Natural Language as input and outputs SPARQL[4]. It is based on a triple model that constructs a parse tree for the data model using the Stanford parser[5] [13]. The parse tree forms the intermediate representation as query triples form. Then PANTO maps query triples to ontology triples which are represented with entities in the ontology. Finally, together with targets and modifiers extracted from the parse trees, ontology triples are interpreted as SPARQL. PANTO was evaluated with the Mooney geography dataset of 877 questions and they reported precision and recall of 88.05% and 85.86%, respectively.

FREyA [4] is a Feedback Refinement and Extended Vocabulary Aggregation system that combines syntactic parsing with ontological knowledge for reducing customization effort. Mapping user queries with ontology concepts is implemented in two ways: automatically and with users help. The answer type of this system is in graph form and Precision and Recall value for the tested data has reached high as 92.4%. For measuring the system performance the Mean Reciprocal Rank (MRR) algorithm was implemented. It is a statistic for evaluating the process to a query. FREya evolved from the previous work QuestIO [24], a Question-based Interface to Ontologies, which translates a Natural Language or a keyword-based question into SPARQL, and returns the answer by executing the formal query against an ontology.

PowerAqua [16,18,15] is a multi-ontology based Question Answering system that takes as input queries expressed in Natural Language and is able to return answers drawn from relevant distributed resources on the Semantic Web. PowerAqua allows the user to choose an ontology and then ask Natural Language queries related to the domain covered by the ontology. The system architecture

[3] `http://start.csail.mit.edu/`

[4] `http://www.w3.org/TR/rdf-sparql-query/`

[5] The Stanford parser is a Natural Language parser which works out the grammatical structure of sentences, supplied by the Stanford Natural Language Processing Group at `http://nlp.stanford.edu/software/lex-parser.shtml`.

and the reasoning methods are completely domain-independent, relying on the semantics of the ontology and the use of generic lexical resources, such as WordNet. The system is capable of learning the user's jargon in order to improve his experience over time. Their learning mechanism uses ontology reasoning to learn more generic patterns, which could then be reused for the questions with similar context. PowerAqua evolved from the earlier AquaLog system [17], a portable ontology-based semantic Question Answering system for intranets. The Performance of the AquaLog was based on Precision, Recall and also failure types are referred separately. An average of 63.5% successful answers are retrieved from ontology with closed domain environment.

Querix [12] is an ontology-based Question Answering system that relies on clarification dialogues in case of ambiguities. This system is composed by an user interface, an ontology manager, a query analyzer, a matching center, a query generator, a dialogue component and an ontology access layer. Natural Language queries are converted into SPARQL query form and Wordnet is used to identify synonyms. Stanford parser is also used in this system to provide a syntax tree for the Natural Language query. Querix does not exploit logic based semantic techniques.

In recent years, mechanisms that classify automatically questions have become essential components for question answering systems, helping and clarifying the questions' types and the expected answers. For instance, in [23] the authors proposed an automatic method for question categorization in a user-interactive question answering system, which includes feature space construction, topic-wise words identification and weighting, semantic mapping, and similarity calculation. Manfred Krifka [14] classified three types of questions, according to the type of the lacking information: constituent questions, polarity questions and alternative questions. Constituent questions create an open proposition by leaving parts of the description of the proposition unspecified. Languages apply interrogative proforms for this purpose. In English, these pro-forms have an initial wh-(pronoun or question). The polarity questions are also called "yes/no-questions". Finally, the alternative questions are semantically related to constituent questions, differ from these as they mention the possible completions explicitly. Hirschman and Gaizauskas [8] presented a possible way to distinguish questions by answer type: factual answers, opinion answers and summary (description) answers. Also, they regarded that is possible to distinguish different kinds of questions: "yes/no" questions, "wh" questions, indirect requests and commands.

Our proposal is a friendly, simple and cooperative Question Answering system. At this stage, we not claim that our proposal to be a complete intelligent system by interpreting and understanding the input questions. It exploit a reduce set of Natural Language Processing techniques (like Stanford parser, Discourse Representation Structures, WordNet), inference rules and open a controlled dialogue with the user when cannot continue the process of achieving the answer. The main difference is the cooperative way that it reaches the answers to the Natural Language questions posed by the user. We interact with the user in order to disambiguate and/or to guide the path to obtain the correct answer to the query posted, whenever this is possible to do by the reasoner. We also use cooperation to provide more informed answers. The answers are presented in Natural Language and have to clarify what the system can infer about the question, from the knowledge domain. Therefore, the cooperative answer provided by our system has to explain the failure of a query to produce results and/or suggest follow-up queries. In the case where a query does produce results, the cooperative answer will provide additional information not explicitly requested by the user. The points where our proposal is similar with the systems presented above are: the use of semantic interpretation techniques and syntactic parser to interpret and represent in some way the questions posed by the users; the use of reasoning and inference techniques to extract and filter the information needed from the knowledge bases.

4 Dialogue Manager

The Dialogue Manager (DM) is the main component of the system proposed in [20,19], a Cooperative Question Answering system for Ontologies. In a brief way, the DM searches for an answer by looking at the semantic interpretation of the question, the type of the expected answer, the structure of the ontology and the information available on the Semantic Web, as well as using string similarity matching and generic lexical resources, aiming to provide a direct and informative answer. Moreover, the DM verifies the question presupposition, chooses the sources of knowledge (Ontologies, WordNet, etc.) to be used; decides when the answer has been achieved or iterates using new sources of knowledge. The decision of when to relax a question in order to justify the answer, when to clarify a question and how to clarify it, is also taken in this module.

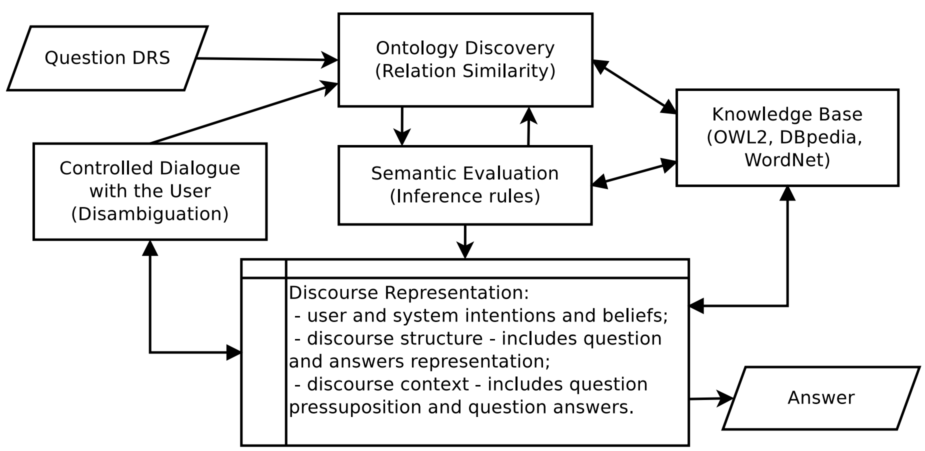

Fig. 1. Dialogue Manager Architecture.

The architecture of the Dialogue Manager is presented in Figure 1 and can described as follows:

Question DRS is the input information for the Dialogue Manager and consists of the Discourse Representation Structure[6] of the Natural Language question posed by the user and is supported by Discourse Representation Theory [10]. The transformation of the Natural Language question into its corresponding DRS is supported by two modules of the main system: the Syntactic Analysis and the Semantic Interpretation.

Ontology Discovery module is invoked when the DM has to look for knowledge base entities that represent the question's concepts. At this stage, the system performs ontology matching, e.g., has to transform the question's DRS predicates into their corresponding ontological representation. So, the Ontology Discovery module maps the question's terms into the ontology's concepts, in order to define and represent the question in terms of knowledge domain ontology. Essentially, this module searches for similarities between labels according to their string-based, taking into account abbreviations, acronyms, domain and lexical knowledge. To maximize recall, the Ontology Discovery searches for classes, properties, instances and/or data values that have labels matching or containing a search term either exactly or partially. If no entity is found, the question concept is extended with its synonyms, hypernyms and hyponyms obtained from WordNet [28]. Afterwards, a set of semantic resources is extracted and may contain the information requested.

Semantic Evaluation module concerns to the pragmatic evaluation[7] step of the system, where the question semantic representation is transformed into a constraint satisfaction problem. The Semantic Evaluation must reinterpret the representation of the sentence, based on the considered ontology, in order to obtain the set of facts that represent the information provided by the question.

This step is executed when the Ontology Discovery module finds a representation of the DRS. Then the system has to find the resources/entities of the knowledge base that are solutions to the ontological representation of the DRS. The solution will be added to the solutions set and the ontology representation of the DRS will be added to the discourse representation associated to the question. The Semantic Evaluation uses of ontologies, SPARQL queries and logic based semantic techniques.

In outline, the DM is invoked after transforming the Natural Language question into its semantic representation and controls all the steps until the end, e.g., until the system can return an answer to the user.

[6] For us a DRS is a set of referents, universally quantified variables and a set of conditions (first-order predicates). The conditions are either atomic (of the type $P(u_1, ..., u_n)$ or $u_1 = u_2$) or complex (negation, implication, disjunction, conjunction or generalized quantifiers).

[7] The pragmatic evaluation is the capacity to judge or calculate the quality, importance, amount or value of problem solutions that are solved in a realistic way which suits the present conditions rather than obeying fixed theories, ideas or rules.

The implementation of the proposed Dialogue Manager is based on Logic Programming, specifically Prolog. Of all the reasons for this choice, the main one is the fact that there is a vast amount of libraries and extensions for handling and questioning OWL2 ontologies [25], as well as incorporating the notions of context in the process of reasoning. In addition, the WordNet also has an export to Prolog.

4.1 Answer Extraction

Answer extraction consists in finding all solutions to the question posed by the user. That is, when the Natural Language question has been transformed into its semantic representation, the Dialogue Manager resorts to the ontology structure and the information available on the Semantic Web, as well as string similarity matching and generic lexical resources, in order to obtain the set of entities that are solutions to the question. The DM must supervise the search (made at Ontology Discovery step) and validation (made by Semantic Interpretation module) of the entities among the knowledge base and when a solution is found, it will be added to the discourse representation associated to the initial question.

The classification of the question is needed to process and reason about the answer. In the present work, questions are classified as one of the types supported by the system, including: basic queries requiring an "yes/no" and for "wh-questions" - who, where, when, what, and which, respectively, "PERSON/ORGANIZATION", "PLACE/LOCATION", "DATE/TIME", "THING/OBJECT", and "CHOICE". The question's classification is made during the transformation of the Natural Language question into its corresponding representation DRS.

Consider the question "*Where is the Taj Mahal?*", presented in [3], and its semantic representation:

```
drs('PLACE/LOCATION',
    [where-X, exist-Y, exist-Z],
    [name(Y,'Taj Mahal'), location(Y,Z), place(X)],
    [is(X,Z)]).
```

where the discourse's referents are `where-X`, `exist-Y` and `exist-Z`, with `X` an entity of the discourse universally quantified and `Y` and `Z` existentially quantified discourse entities, the predicate main question is `is(X, Y)` and the presupposition predicates are `name(Y, 'Taj Mahal')`, `location(Y, Z)` and `place(X)`.

For this query, we can find entities in DBpedia, which are related to the name "Taj Mahal". So, we can get facts about the entities through non-taxonomic relation that verifies the question. For instance, one entity that is related to the term "Taj Mahal" is the resource `http://dbpedia.org/resource/Taj_Mahal_Palace` and to state that this entity has its location in Bhopal and that Bhopal is a place, the DBpedia contains the following statements triples RDF:

```
dbpedia:Taj_Mahal_Palace dbpedia-owl:location dbpedia:Bhopal .
dbpedia:Bhopal rdf:type dbpedia-owl:Place .
```

That is, these statements validate the mappings of the questions terms in the ontology, namely: location is mapped into `http://dbpedia.org/ontology/location` and place is mapped into `http://dbpedia.org/ontology/Place`, and determine a solution to the semantic representation of the question.

```
X = http://dbpedia.org/resource/Bhopal
Y = http://dbpedia.org/resource/Taj_Mahal_Palace
Z = http://dbpedia.org/resource/Bhopal
```

The solution, the RDF triples that generate the solution and the mappings of the question's terms in the ontology, which validate the semantic representation of the question, are added to the knowledge base of the question, the discourse's representation.

4.2 Answer Processing

Answer processing consists in determining the final representation of the answer returned to the user, which is interpreted in the knowledge base with the facts that were extracted. At this stage, the DM analyses the facts obtained (questions solutions) and gives the user an appropriate answer, taking into account the type of the question and the type of the expected answer:

- If the set of solutions is empty, the answer has to inform that fact and the user can re-write the initial question, or make a new one, or simply stop the process.
- If the set of solutions has only one solution, the answer presented to the user, besides direct and objective, also informs about the entities that served as support, allowing a better communication between the system and the user.
- If the set of solutions has multiple solutions, various interpretations can be made. If there isn't enough information to decide which one is the correct, a controlled dialogue with the user is initiated. So, it presents a set of alternatives and the user's answer to those alternatives will clarify or restrict the subject is referring to.

In situations when there are ambiguities (multiple answers), the system initiates the clarification mechanism presented in Algorithm 1. The reformulation of this algorithm follows the idea presented by the authors of [22] and to help understand how it works, a brief discussion of the main steps follows:

Evaluation of the Properties: The alternatives of clarification to present to the user must fulfil two important aspects: report as possible the user intentions and having information most likely to be known by the user. With respect to the later, for instance, if the clarification concerns the characteristics of a person, most likely, the user knows its country than its date of birth. Another possible parameter can be considered that the user would know better numerical information.

Algorithm 1 Multiple solution's clarification.

Require: $S = \{s | s \text{ is a solution of the question}\}$
Ensure: Set an answer to the question

while $\#S > 1$ **do**
    For each referent collect their properties
    Evaluate the best property to differentiate the referents
    Choose the best property
    A = values of the best property for each solution
    Show the clarification's alternatives based on the set A
    Receive the user's choice
    Restrict the set of solutions S to the user's choice
end while
Show solution S to the user

Regarding user intentions, the choice of alternatives to be presented to the user consists in selecting the property that contains more information and this is based on the model of Decision Trees Learning [21]. More precisely, the ID3 (Inductive Decision Tree) algorithm used as a classification method in the construction of decision trees.

The ID3 algorithm uses Entropy and Information Gain to build the decision tree. However, the classification of properties by maximizing the information gain gives preference to properties with many values. For that reason we also introduced the Information Gain Ratio as an evaluation criteria, promoting properties with small Entropy and therefore promoting properties with fewer values. So, to help in the task of classification and selection of the best property, whose values will be presented as alternatives to the user for clarification, and then we introduce the concepts, originally presented in [21]: Entropy, Information Gain and Information Gain Ratio.

The Entropy of a set can be defined as the purity (certainty, accuracy) of that set. This concept borrowed from Information Theory defines the measure of "lack of information", namely the number of bits needed, on average, to represent the missing information, using optimal coding. If the set is completely uniform, the Entropy value is zero, and if the set is divided equitably, the Entropy value is equal to 1. Formally, the Entropy of a set is defined as follows:

Definition 1. *Given a set T, with instances of the class i, with probability $p_i \neq 0$. The Entropy of the set T is obtained by the following expression*

$$Entropy(T) = -\sum p_i \times log_2(p_i). \tag{1}$$

The Entropy of a set T verifies the property $0 < Entropy(T) < log_2(n)$, where n is the total of classes i.

Back to our example, the question posed by the user refers to the location of "Taj Mahal". When the DM analyses the set of solutions detects the presence of multiple solutions, since there are several entities that represent the term "Taj Mahal" (see Table 1) and therefore Algorithm 1 is performed.

Table 1. Resources of the knowledge base related to "Taj Mahal"

Name	Ontology Resource
Taj Mahal	`http://dbpedia.org/resource/Taj_Mahal`
Taj Mahal Palace & Tower	`http://dbpedia.org/resource/Taj_Mahal_Palace_&_Tower`
Taj Mahal Palace	`http://dbpedia.org/resource/Taj_Mahal_Palace`
Taj Mahal Hotel (Delhi)	`http://dbpedia.org/resource/Taj_Mahal_Hotel_(Delhi)`

Table 2. Entropy value of the set T.

Class i	**#(Class** i**)**	$\mathbf{p_i = \frac{\#Class}{\#T}}$	$\mathbf{-p_i \times log_2(p_i)}$
Taj Mahal	81	0.36	0.5306152278
Taj Mahal Palace & Tower	58	0.2577777778	0.5041618283
Taj Mahal Palace	37	0.1644444444	0.4282672424
Taj Mahal Hotel (Delhi)	49	0.2177777778	0.4789088711

Since DBpedia properties are expressed by triples RDF, forming the properties that are associated to each question referents. In the example, we exclude the referent `X` associated to the question adverb, because it is a referent that is semantically related to what the user wants to know. By analogy, also the referent `Z` is excluded, because the condition `is(X,Y)` makes it semantically equal to the referent `X`. Thus, we are left with the referent `Y`, which is associated with the name "Taj Mahal". Consequently, the set T consists only of properties associated to the referent Y.

Table 1 presents the different values of the solutions found by the system associated to the referent `Y`. To these values, which are the classes that instances of T belong, we have to construct the set of all properties that are related with them. The set T has 225 properties. Thus, according to values presented in Table 2, the Entropy value of the set is:

$$\begin{aligned} Entropy(T) &= \sum(-p_i \times log_2(p_i)) = \\ &= 0,5306152278 + 0,5041618283 + 0,4282672424 + 0,4789088711 = \\ &= 1,9419531697 \end{aligned}$$

and has $0 < Entropia(T) = 1,9419531697 < log_2(n) = log_2(4) = 2$.

The construction of a derivation tree is guided by the objective of reducing the Entropy, the difficulty of predicting the variable that defines the classes. The Information Gain defines the decrease in Entropy. Thus,

Table 3. Entropy, Information Gain and the Information Gain Ratio of some of the properties used to clarify "Where is the Taj Mahal?".

Properties	Entropy	Information Gain	Gain Ratio
`http://purl.org/dc/terms/subject`	0,0869565	1,855	21,3325053331
`http://dbpedia.org/property/location`	0,333333	1,60862	4,8258648259
`http://dbpedia.org/ontology/location`	0,333333	1,60862	4.8258648259
`http://dbpedia.org/property/wikiPageUsesTemplate`	0,5	1,44195	2,8839
`http://dbpedia.org/property/latns`	1	0,941953	0,941953
`http://dbpedia.org/property/longd`	1	0,941953	0,941953
`http://dbpedia.org/property/longew`	1	0,941953	0,941953
...	...	...	...

Definition 2. *The Information Gain is the expected reduction in Entropy caused by partitioning the data according to the property testing P. The Information Gain value for the property P is obtained by the expression:*

$$Gain(T,P) = Entropy(T) - \sum_{v \in values(P)} (\frac{|T_v|}{|T|} \times Entropy(T_v)) \tag{2}$$

Back to our example, we have to calculate the Information Gain for each distinct property value and add proportionally to obtain the final Entropy value of the property. The set T has 74 distinct properties.

Definition 3. *Consider the property P of the set T, the Information Gain Ratio value of the property P in the set T is obtain using the following expression:*

$$GainRatio(T,P) = \frac{Gain(T,P)}{Entropy(T,P)} \tag{3}$$

The Information Gain Ratio is not defined when $Entropy(T,P) = 0$.

The Entropy, Information Gain and Information Gain Ratio values of some of the properties used to clarify the multiplicity of solutions are presented in Table 3.

Choose the Best Property: The Information Gain criterion of a property selects the one that maximizes the Information Gain. However, this criterion gives preference to attributes with many possible values. In these cases, it could be chosen one attribute irrelevant, where there is only one alternative for each possible value. Therefore, the number of alternatives would be equal to the number of identifiers and the Entropy value would be minimal because, in each property, all samples (if only one) belong to the same class, which would generate a maximum gain, although totally useless. When this problem occurs, e.g., when the property P to the set T has $Entropy(T,P) = 0$, which corresponds to the Information Gain maximum value, we use the Information Gain Ratio as evaluation criteria to choose the best property.

However, even with these criteria we may have as best property, a property for which information is less known to the user. For example, if the best property represents numeric values (such as birth dates, number of citizen card, etc.), we assume that the user may not know such information. By the presentation of such alternatives to the user will result in unnecessary step. Thus, in these cases we define as priority the properties containing information (value) that is non-numeric.

Definition 4. *Consider the property P of the set T. P is the best property when the Information Gain Ratio value is the highest and its values are non-numeric.*

Back to our example, and according to the Table 3, the property with the highest value of the Information Gain Ratio is the first one (`subject`). Since it has non-numeric values (text), it will the chosen one.

Controlled Dialogue with the User: In the controlled dialogue with the user, besides specifying the alternatives presented by the system, the user also has three possibilities to interact with the system, namely: `?` means "I do not know"; `!` meaning "show all answers"; and the term `quit`, that ends the process. In the first case, the system displays a new set of alternatives according to the current evaluation of properties. In the second case, the system displays all solutions and the process is finished. In the third case, the system simply ends the process.

Continuing our example and according to the evaluation made, the best property is `http://purl.org/dc/terms/subject`. Consider T_1 the set of events of the set T where the best property occurs. The set of alternatives A is formed by the different values of the best property in the set T_1. Afterwards, the system starts a controlled dialogue with the user and presents the alternatives A:

```
USER: "Where is Taj Mahal?"
SYSTEM: "Taj Mahal" refers to a:
1 - Buildings and structures completed in 1654
2 - Buildings and structures in Agra
3 - Mausoleums in India
...
USER: 3
```

The alternative chosen by the user leads the system to one solution.

```
LOCATION/PLACE:
Agra, India
RESOURCE:
http://dbpedia.org/resource/Taj_Mahal
The Taj Mahal is a mausoleum located in Agra, India.
```

If the user chooses one alternative with multiples solutions, the system has to execute the Algorithm 1 to clarify the ambiguity and a similar mechanism occurs with the difference that now we have a richer question context.

5 Experimental Results

For the evaluation of the Dialogue Manager we used the DBpedia ontology OWL2 ([2]), SPARQL *endpoints* to query DBpedia database and the DBpedia *Lookup Service* [8] to look up DBpedia URIs[9] for related keywords.

The evaluation test was performed using a set of 84 questions, presented in TREC 9 (*The Ninth Text REtrieval Conference* [26]). The set under analysis contains only direct "wh" questions, which comprises the following questions:

Table 4. Evaluation Results of 84 questions of TREC 9.

	No answer	Correct Answer	Wrong Answer
Our Proposal	10	68	6
START	8	67	9
PowerAqua-DBpedia	3	66	15

Comparison with Similar Systems The tests were performed manually and the answers were validated one by one. Table 4 presents the performance of our proposal and other two systems. It provides the numbers of no answers, correct answers and wrong answers. It shows that we can consider that our proposal has good results compared to other systems in terms of correct answers. We have detected that when the system does not find any solution, if we apply constraint relaxation techniques to referent predicates, the system can obtain some solutions.

Performance of our Proposal From table 5, we know that our system has not obtained any answer to 10 questions (12% of the questions). That is, the system did not found, in the knowledge base, the resources which identify the questions terms or the entities that are solutions to the questions. Starting a dialogue with the user will allow us to rewrite the question posed, clarify the terms or place a new question. This way, and if the knowledge base has the answer, clearly we will be able to increase the results. Analysing the remaining corpus, reduced to 74 questions, we obtained 68 correct answers (81% of the questions), and we have verified it manually. Within these, 48 questions were multiple answers (57% of the questions) that, with the user clarification, the system returned the expected answer. We found that, for each multiple solution question, the system achieved an average of 3-4 solutions. Clearly, a reduced set of alternatives highlighting the potential of the system in searching for the correct answer. The remaining 6 questions (7% of the questions), the system did not get the correct answer. These failures are identified with some factors that lead to incorrect interpretations,

[8] `http://wiki.dbpedia.org/lookup/`
[9] `http://tools.ietf.org/html/rfc3986/`

namely: semantic representation of the question; incomplete or badly formulated questions; dimension of the knowledge base, or incomplete information and non-uniform ontology resources.

Table 5. Information results of evaluation test applied to our proposal, a Cooperative Questions Answering system for Semantic Web.

	Total	Relevance
No answer	10	12%
Correct Answer	68	81%
Simple	20	24%
Multiple	48	57%
Wrong Answer	6	7%
Simple	2	2%
Multiple	4	5%
Average of multiple answer 3,6166667		

This is still a preliminary evaluation, summarizing just a first set of tests, whose results have produced satisfactorily, allowing us to verify the effectiveness of the proposed question answering system and to identify the weaknesses that will allow us to improve its performance. In the future we intend to present a more complete evaluation, extend the set of questions to others questions' types and include the evaluation of the execution time. However, the results encouraged us to proceed.

6 Conclusions and Future Work

We presented a Dialogue Manager that, throughout the analysis of the question and the type of expected answer, allows us to provide accurate answers to questions in Natural Language. The Dialogue Manager not only represents the question semantics, but also represents the structure of the discourse that includes the user intentions and the questions context, gives us the ability to deal with multiple answers and to provide justified answers.

The experiments on a set of simple and direct "wh" questions showed that our system has promising results. We consider that adding a tool, like Dialogue Manager, to the Question Answering systems improves substantially the system performance. Resorting to a controlled dialogue in order to clarify ambiguities helps the system to interpret the user's intentions. These dialogues increase the results obtained by the system and help generating a more objective answer with the information desired by the user. Therefore, it is our opinion, that our proposal approaches rapidly to one that helps bridging the gap between the Semantic Web and real world users.

As future work, we plan to improve the search ontology techniques and the inference rules, aiming to enhance the results of correctness and the execution time. We also plan to increase the number of tests, covering the remaining types of questions (including more complex questions) and to define a more complete quantitative, qualitative and comparative evaluation of the performance of the overall system. In addition, we intend to extend the system to the Portuguese language. For this purpose, it is necessary to enrich the knowledge domain with concepts that can be deducted from the initial domain. We also wish to extend the knowledge domain with other ontologies, enabling us to support the concept of open domain and take advantage of the large amount of heterogeneous semantic information provided by the Semantic Web.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J.: Dbpedia: A nucleus for a web of open data. The Semantic Web 4825(Springer), 722–735 (2007)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009), `http://www.sciencedirect.com/science/article/pii/S1570826809000225`
3. Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., Jacquemin, C., Lin, C.Y., Maiorano, S., Miller, G., Others: Issues, tasks and program structures to roadmap research in question & answering (q&a). Document Understanding Conferences Roadmapping Documents pp. 1–35 (2001), `http://www.inf.ed.ac.uk/teaching/courses/tts/papers/qa_roadmap.pdf`
4. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: combining syntactic analysis and ontology-based lookup through the user interaction. In: Proceedings of the 7th international conference on The Semantic Web: research and Applications - Volume Part I. pp. 106–120. ESWC'10, Springer-Verlag, Berlin, Heidelberg (2010)
5. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, illustrated edition edn. (May 1998), `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/026206197X`
6. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (Jun 1993), `http://dx.doi.org/10.1006/knac.1993.1008`
7. Guo, Q., Zhang, M.: Question answering based on pervasive agent ontology and semantic web. Knowledge-Based Systems 22(6), 443–448 (Aug 2009)
8. Hirschman, L., Gaizauskas, R.: Natural language question answering: The view from here. Natural Language Engineering 7(4), 275–300 (2001)
9. Horrocks, I.: Ontologies and the semantic web. Communications of the ACM 51(12), 58–67 (2008)
10. Kamp, H., Reyle, U.: From Discourse to Logic, Studies in Linguistics and Philosophy, vol. 42. Kluwer (1993)
11. Katz, B., Lin, J.J., Felshin, S.: The start multimedia information system: Current technology and future directions. In: Multimedia Information Systems. pp. 117–123 (2002)

12. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A natural language interface to query ontologies based on clarification dialogs. In: 5th International Semantic Web Conference (ISWC 2006). pp. 980–981. Springer (November 2006)
13. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. pp. 423–430. ACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003)
14. Krifka, M.: Questions. In: Heusinger, Klaus von, C.M.P.P. (ed.) Semantics: An international handbook of Natural Language Meaning, vol. 2, chap. 66, pp. 1742–1785. Berlin, Mouton de Gruyter (2011)
15. Lopez, V., Fernández, M., Motta, E., Stieler, N.: Poweraqua: Supporting users in querying and exploring the semantic web. Semantic Web 3(3), 249–265 (2012)
16. Lopez, V., Motta, E., Uren, V.S.: Poweraqua: Fishing the semantic web. In: ESWC. pp. 393–410 (2006)
17. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. In: ESWC. pp. 546–562 (2005)
18. Lopez, V., Uren, V.S., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. Journal of Web Semantic 5(2), 72–105 (2007)
19. Melo, D., Rodrigues, I.P., Nogueira, V.B.: Puzzle out the semantic web search. International Journal of Computational Linguistics and Applications 3(1), 91–106 (June 2012)
20. Melo, D., Rodrigues, I.P., Nogueira, V.B.: Work out the semantic web search: The cooperative way. Advances in Artificial Intelligence 2012, 3:3–3:3 (January 2012)
21. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (Mar 1986), `http://dx.doi.org/10.1023/A:1022643204877`
22. Quintano, L., Rodrigues, I.: Using a logic programming framework to control database query dialogues in natural language. In: Proceedings of the 22nd international conference on Logic Programming. pp. 406–420. ICLP'06, Springer-Verlag, Berlin, Heidelberg (2006), `http://dx.doi.org/10.1007/11799573_30`
23. Song, W., Wenyin, L., Gu, N., Quan, X., Hao, T.: Automatic categorization of questions for user-interactive question answering. Information Processing & Management 47(2), 147 – 156 (2011), `http://www.sciencedirect.com/science/article/pii/S0306457310000221`
24. Tablan, V., Damljanovic, D., Bontcheva, K.: A natural language query interface to structured information. In: Proceedings of the 5th European semantic web conference on The semantic web: research and applications. pp. 361–375. ESWC'08, Springer-Verlag, Berlin, Heidelberg (2008)
25. Vassiliadis, V., Wielemaker, J., Mungall, C.: Processing owl2 ontologies using thea: An application of logic programming. In: OWLED (2009)
26. Voorhees, E.M.: Overview of the trec-9 question answering track. In: Proceedings of the Ninth Text REtrieval Conference (TREC-9). pp. 71–80 (2001)
27. Wang, C., Xiong, M., Zhou, Q., Yu, Y.: Panto: A portable natural language interface to ontologies. In: Proceedings of the 4th European conference on The Semantic Web: Research and Applications. pp. 473–487. ESWC '07, Springer-Verlag, Berlin, Heidelberg (2007)
28. Witzig, S., Center, A.: Accessing wordnet from prolog. Artificial Intelligence Centre, University of Georgia pp. 1–18 (2003)