ABC Research Alert
Vol 8, Number 1
Published Online: 06 April 2020
https://abc.us.org/ojs/index.php/abcra

Question Answering System over Linked Data: A Detailed Survey

G. M. Rasiqul Islam Rasiq[1*], Abdullah Al Sefat[2], Tanjila Hossain[3], Md. Israt-E-Hasan Munna[4], Jubayeath Jahan Jisha[5], Mohammad Moinul Hoque[6]

[1*]Department of Computer Science and Engineering, Daffodil International University, **Bangladesh**
[2,3,4,5,6]Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, **Bangladesh**

*(rasiq.cse@diu.edu.bd)

Abstract

As the amount of information in the world is growing very quickly, in the case of semantic web this increasing amount of information is becoming more difficult to find and manage the exact answers to our various questions. To overcome these difficulties some systems have been developed that make it work for us. But there exists many challenges in developing these systems that require a lot of improvement. In this tutorial we give a basic understanding of Semantic web, RDF triple, SPARQL query language. Here we will discuss the main obstacles for QA system in processing the questions and a detailed survey of the existing systems. We also provide some advantages and disadvantages of existing QA systems. We also discuss the evaluation campaigns of the existing models based with their precision, recall and F-1 scores on QALD dataset.

Keywords

Natural Language Processing, Question Answering (QA) Systems, Ontological resources

I. INTRODUCTION

The amount of information is growing very rapidly all around the world, Question Answering System become more and more important for retrieving and extracting information from this massive amount of knowledge. Human can easily understand each other by means of communication using language but computer cannot. To make the computer understand what we are asking it needs to process NL (Natural Language) to find the answer from somewhere since the computer does not have a brain. The Semantic Web (SW) [1], has the structured data. The structured data is found in web pages and computers can easily access them. Question answering system which is also known by QA system plays an important role to build a system that takes input questions from the user and will give the

accurate answer as an output. Here the users can ask questions in natural language that is also a significant characteristics of the QA system.

The stored web pages data either structured or unstructured might be called as databases. Some databases also known as knowledge bases like Wikipedia, Freebase, and DBpedia etc are a huge source of information. Among them the information can be classified into structured and unstructured data. Linked Data created by Berners-Lee in 2006 [2]. Linked data can be categorized as structured data. **Resource Description Framework** is a very well-known format that has three parts. [3].Hyper Tex Transfer Protocol as another example of structured data. The Semantic Web at a time includes more new data and also make links to make connection between data of a source to the data of another source so that the end users and the QA systems easily find the required data at ease. When we have some of the data, we can find another data related to the previous part, the Linked data stored as a data model named RDF model. RDF is elaborated as Resource description Framework [15]. Every data is arranged in three parts. Subject, Predicate, Object are the name of the parts. It is called "RDF Triple" as there are three parts in the model.

II. MAJOR CHALLENGING FACTS

When a user want to get some information from linked data cloud, he faces some major challenging facts which will be described here shortly. There should have been a relation between the question asked by the end users and the answer retrieved from the data cloud by the help of the Question Answering System. So we can understand that the task is not to enclosed with only questioning and getting the answers, it also needs some translation between the questions one side and the answers other side. Some sort of translation to a suitable form is needed for the questions asked by the user and also there is a validation among multiple answers to pick up the right one.

A. Bridging natural language to Linked Data

As we said that Question Answering system has to face some problems the main problems we can say that bridging between the questions with the information stored in the linked data cloud. Users when ask questions in their natural language the questions are not always in their suitable format that exists in the data cloud. So analyzing the user asked questions and then translating to the particular form is a necessary step to get connected to the answers.

1) Mapping vocabulary elements to NL: To map the expressions derived from the questions asked in natural language to vocabulary ontology exists for the data are the biggest challenge because in doing so finding lexical and structural mismatches is one of the main challenges. To understand vocabulary element matching with natural language expression clearly let us visualize a simple database table with an attribute named player and we want to retrieve a value from that attribute player. But while querying if we use gamer as the attribute name instead of player, it will be unable to retrieve the value from the database even though player and gamer is a synonym.

2) Meaning variation: The question is needed to be analyzed because of meaning variations such as, I was surprised that Jisha lost, it surprised me that Jisha lost, and That Jisha lost surprised me

B. Data quality and data heterogeneity

Multilingualism is another challenge that is faced by natural language processing. Because users use not only the English language but also their native language. Users naturally can ask questions in their own language. It is not the obvious fact that the users will always ask questions in English. So, here is the problem as the existing Question Answering systems can't process questions other than in English language. As a result, our desired Question Answering Systems should process questions asked in multiple languages and this is the fact of multilingualism. (Shown in a recent study [4]).

C. Performance and scalability

It is of paramount importance that Questions Answering Systems give an answer which is accurate and complete since a wrong answer is less desired than no answer at all. For linked data, a system is necessary that will deal with heterogeneous data. QA systems need to have the ability to determine whether the data retrieved from the query contains the answer or not; since, in linked data the knowledge is incomplete. Duplicate information is often contained in the dataset since various dataset may use different vocabulary for the same meaning. [5]

D. Coping with distributed and Linked Datasets

Yielding performant systems that answer in time is challenging. It is a challenge to work with a dataset which has billions of tuples; since, real time performance is desired by the users and it requires questions to be processed and answered under a second. However, real time performance can be achieved if the system uses proper indexing, search heuristics and uses parallel distributed computing. In [6], for the same dataset, FREyA[7] required 36 seconds whereas Aqua Log needed only 20 seconds, which is substantially less compared to the aforementioned system.

E. Integration of data

A handful of systems only take into account that, the avail-able structured data is distributed among a huge intertwined collection of dataset. Moreover, amalgamation of information of various sources can provide answers to the questions. However, there has not been a lot of research on evaluating questions of distributed linked datasets.

F. Integrating structured and unstructured data

A large amount of information is available only in the form of text. In order to integrate structured and unstructured data, there has to be approaches that deal with the specific characteristics of structured data and also looks up and finds information from multiple sources and processes them and finally, integrates the collected information to form an answer. The main hurdle, however, remains mapping the vocabulary elements.

III. HOW QUESTION ANSWERING SYSTEM WORKS

A. Question answering system domain

Question answering system domain can be either domain specific or open domain [8]. For different purposes, we use these two types of domain. Now, we will discuss these two types of domain.

1) Open Domain: The most current Question Answering (QA) system use the open domain system. Open domain system means that all information in question answering system are from open and free domain like Wikipedia, WordNet etc. Suppose, if we ask a question about catch like “What does he catch?”. If a QA system uses open domain then here the word “catch” is vague that is unclear because “catch” has few different meaning in different domain about which we will talk later.

2) Specific Domain: The main characteristic of question-answering in open domain is that it is developed for a specific purpose. Suppose if we consider the previous example then we can say that if we consider that the specific domain is sports then the “catch” word means catch ball which can be cricket ball, volleyball or basketball or any other ball related to sports.

B. Tasks involved in question Answering system

All of the approaches to Question Answering face the same problem discussed in the section 2 Main challenge. Almost all of the systems are based on same components but their implementations are different from each other. Although these systems implementations are different they follow some specific

tasks. In this section we present some dimensions of Question Answering and their components. Here below figure 1 shows the flow chart of Question Answering system

C. Dimensions of question answering

To understand the different challenges involved in Question heuristics and uses parallel distributed computing. In [6], for the same dataset, FREyA [7] required 36 seconds whereas answering it is important to identify different dimensions.

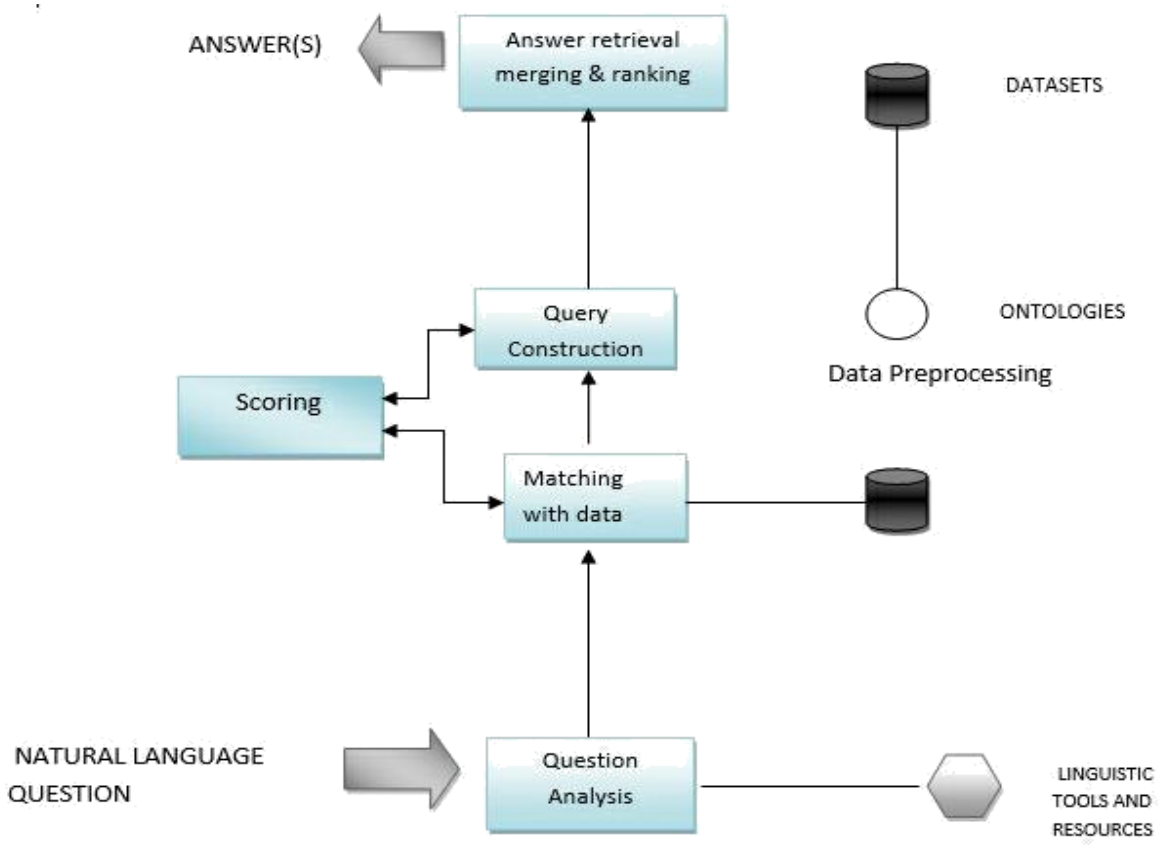

Figure 1: Flow chart of QA system

D. Question and answer type

What does it mean by question and answer type? To reply the question we need to know that there are different types of sentences. Some questions are asked to know a name or numeric values or location name etc; again some questions are asked to know a list of something, some are asked to know someones opinion. So based on what we want to know we ask different questions and answers are also different. These different type of questions [9] have different name.

1) *Factoid questions/ Predictive questions:* e.g.
   Who in-natural language format is important to make the common end vented radio?
   What is the highest mountain in earth?
   When did Humayun Ahmed die?
   Where is Great Wall of China?
   How far is the moon from earth?
   Give me all cities in china.

2) *Definition questions:* e.g. Who was Humayun Ahmed?
3) *Evaluative or comparative questions:* e.g. What is the difference between import and export?

4) *Association questions:* e.g. What is the connection between Barack Obama and China?
5) *Opinion question:* e.g. What do Americans think of gun control?
6) *Process questions:* e.g. How do I make a coffee?

Most of the time Question Answering system focus on the factoid and definition type question because this type of information is available in most of the data source.

E. Data source:

Question Answering systems are different from each other with respect to different data sources. 3 types of data sources are,

1) Structured data, relational database and linked data (DBpedia, Freebase)

2) Semi-structured data. E.g.XML document

3) Unstructured data, e.g. text document.

F. Components of question answering system

1) Data Preprocessing: Data preprocessing means transformation of raw data into an understandable format for computer. To match natural language expressions with labels of vocabulary elements index of the data set is needed.

2) Question Analysis: Question analysis includes linguistic analysis (syntactic and semantic analysis) of the question, also detection and extraction of question features. Linguistic analysis also rely on the tools. Some of the tools are part-of-speech taggers and parsers. Here the figure?? Shows the difference between the structured and unstructured

3) Data Matching: Sometimes there are differences between the question vocabulary and dataset elements.

4) Query construction: After data preprocessing and question analyzing structured query is constructed.

5) Answer retrieval: To extract the answer structured query needs to be executed over the chosen database.

6) Assessment: Even though the answer is extracted there are some possibilities that the whether the answer is the expected one or not. The reason behind this kind of confusion is because of querying over different data sets. To ensure assessment is important. So assessment is an important component to fulfill the users reliability.

7) Answer representation: Even though answer is retrieved, the answer may not be in an understandable format. It may be in an URI or triple format. So representing the answer in user able to understand the answer.

IV. LITERATURE REVIEW

A. Approaches based on controlled natural language

This approach is based on controlled input that means when user asking any answer in natural language, the system continues suggesting the next words until the question is completed. If the user's input contains any word which was not suggested then the question can't be parsed or queried. So in this approach users' input is controlled by the system. According to this kind of approach, system unambiguously interprets words or a restricted subset of natural language.

1) *GiNSENG [10]:* It is A Guided Input Natural Language Search Engine is based on controlled natural language question answering. The system does not use any predefined vocabulary. It does not interpret syntactical query. As vocabulary is controlled the user can give the input from the fixed vocabulary elements that are suggested by the system to get the answer. If the users question contains any ontology that does not belong to the systems vocabulary then the system cannot answer the question.

Dataset: Mooney Dataset

Working procedure: Basing on a grammar, the incremental parser of the system offers the possible completions of a user's entry by presenting the user with choice pop-up boxes (as shown in figure 2). These pop-up menus offer the users suggestions on how to complete a current word or what the next word might be. The possible choices get decreased as the user continues typing. The fact is entries that are not in the pop-up list are ungrammatical and not accepted by the system. In this way, Ginseng guides the user through the set of possible questions preventing those unacceptable by the grammar. Once the query is completed, Ginseng translates the entry to SPARQL statements, executes them against the ontology model using Jena, and displays the SPARQL query as well as the answer to the user. Here figure 2 shows the working procedure with its pop-up window.

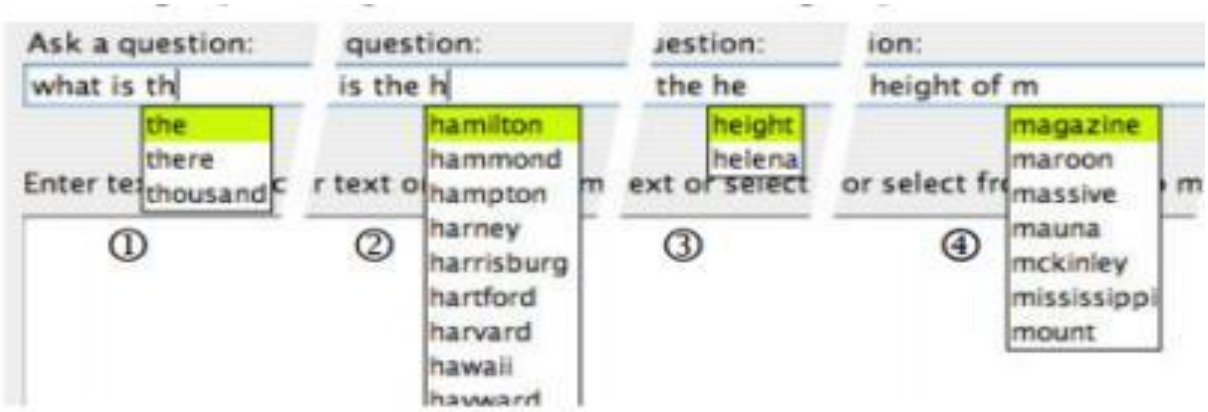

Figure 2: Ginseng query-completion choice window

Result: In the article, "How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users?" by E.Kaufmann, A.Bernstein, in proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, 2007,[10] in one evaluation with 20 users, it is showed that Ginseng is very simple to use without any training (as opposed to any logic-based querying approach) resulting in very good query performance (precision = 92.8%, recall = 98.4%). Furthermore, it has also been found that even with its simple grammar/approach Ginseng could process over 40% of questions from a query corpus without modification.

Advantages: This system ensures that the input query generates correct result. This system is very easy to implement and it supports the users very much.

Limitations: Ginseng cannot process all NL queries as it has controlled vocabulary elements. That means if any user gives an input which does not include systems fixed element then it cannot retrieve the answer. It has a very few sentence structure possibilities.

B. Approaches based on formal grammars

This approach is based on linguistic grammar. Linguistic grammar involves syntactic & semantic representation to lexical units that means words by combining the meaning of the parts in the grammar. This idea uses the principle of composition's semantics or meaning to compute an overall semantic representation of a question. This approach works well complex questions if the question is parsed correctly, otherwise it fails.

1) *ORAKEL [11]:* ORAKEL is a system based on formal grammars and it can easily adapt to a given domain.

Flow chart: Here in the figure 3 below the flow chart of ORAKEL system is shown:

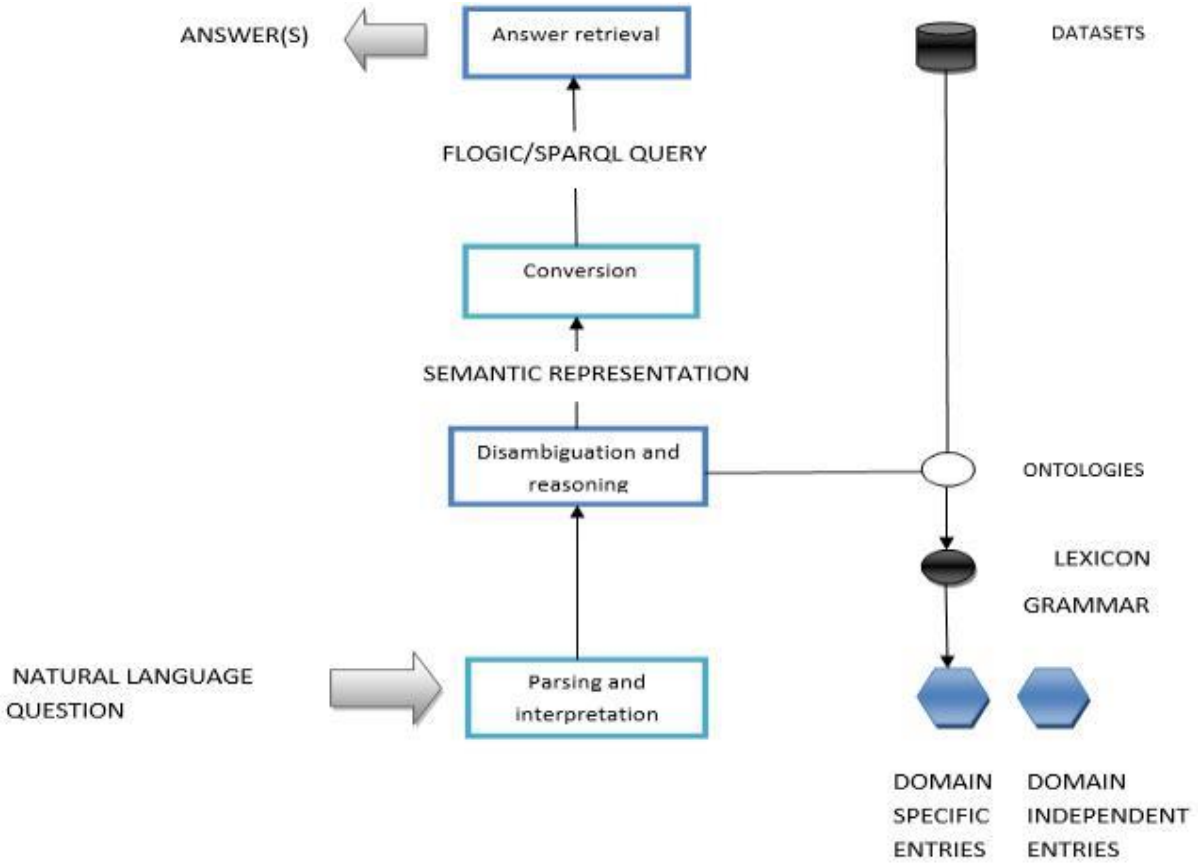

Figure 3: Flow chart of ORAKEL system.

Working procedure: The ORAKEL natural language interface addresses all the above challenges, focusing particularly on reducing the effort of adapting the system to a given domain. ORAKEL is an ontology-based natural language system in two-senses. First, the ontology for a certain knowledge base is used to guide the lexicon construction process. Alternatively, parts of the lexicon are automatically generated from the underlying ontology. Most importantly, the ontology is at the core of whole lexicon acquisition process in ORAKEL. Second, ORAKEL is ontology-based in the sense that it is a natural language interface which depends on removal to answer a user's query. As ORAKEL depends on a well-defined removal process to answer a query, an important requirement is that the user's question is translated into logical form.

Result: In Table I we can see the results of different lexica in ORAKEL system. They will refer to the author as A and the other two participants constructing a lexicon as B and C. While A was very familiar with the lexicon acquisition tool, B and C were not and received 10 minutes of training on the tool. Whereas A constructed a lexicon in one turn, B and C constructed their lexicon in two rounds of each 30 minutes. In the first round, they were asked to model their lexicon, while in the second round they were presented those questions which the system had failed to answer after the first round. They were asked to complete the lexicon on the basis of the failed questions. The 24 persons playing the role of the end users also received instruction for the experiment.

Advantages: The answer generation component evaluates the query regarding the knowledge base and presents the answer to the user. Their studies show that ORAKEL can be successfully adapted to different domains in a reasonable

Table I: Results for the different lexica in ORAKEL system amount of time, typically a few hours.

Lexicon	Users	Rec (avg)(%)	prec. (avg)(%)
A	8	53.67	84.23
B(1st lexicon)	4	44.39	74.53
B(2nd lexicon)	4	45.15	80.95
C(1st lexicon)	4	35.41	82.25
c(2nd lexicon)	4	47.66	80.60

Disadvantages: ORAKEL has also very few limitations. Currently, ORAKEL cannot handle ungrammatical input and cannot deal with unknown words. ORAKEL makes a full parse of the input sentence and expects the sentence to be grammatical. If the question is not grammatical it will fail and tell the user that it did not understand the question, without giving any feedback. If a word is unknown, the system will at least inform the user that the word is unknown.

2) *PYTHIA [5]:* As the Semantic Web gives a large amount of ontology-based semantic markup, so Question Answering systems can utilize in order to explain and answer natural language questions. This means that user translated with respect to a particular ontology which gives natural language expressions with a well-defined meaning.
Here pythia is ontology-based Question Answering system. It is based on the two main ideas. Firstly, it uses proper linguistic representations in order to construct general meaning representations that can be translated into formal queries. Secondly, it depends on a identification of the lexical-ontology interface that clarifies linguistic realizations of ontology concepts.
Working procedure: Both parts of the grammar use respectable linguistic representations. If we describe more specifically, we assume grammar entries to be pairs of syntactic and semantic representation. In syntactic representation, we take trees from Lexicalized Tree Adjoining Grammar (LTAG). LTAG is very applicable for ontology-based grammar generation because of its allowance for flexible basic units. In generating grammar from a given ontology, we have to do firstly is to enrich the ontology with information about its verbalization. We use the LexInfo framework for this, which gives us a general frame for creating a declarative identification of the lexicon-ontology interface by connecting concepts of the ontology to information about their linguistic recognition, i.e. word forms, morphology, sub-categorization frames and syntactic and semantic arguments communicate with each other. The lexical entries defined by LexInfo are then used as input to a general mechanism to generate grammar entries.
For parsing and interpreting natural language questions we then used these linguistic representations. The process of mapping natural language input to formal queries can be described as follows: The process has three main steps. At first, the input is handed to parser. LTAG derivation tree is constructed which considers only the syntactic part of the grammar entries. Next, syntactic and semantic composition rules is applied to build a derived tree together. The syntactic rules are LTAG operations and semantic composition rules are parallel operations on DUDE's. When all arguments are filled, the DUDE communicates with UDRS. Then these are translated into a formal query. We used FLogic for query formulation.
Results: Using Pythia on the mentioned user questions and comparing the results of the constructed queries by the gold standard queries, we reached a recall of 67% and precision of 82% and an F-measure of 73.7%.
Advantages: The main benefits of Pythia is that, it can handle linguistically complex queries which involves quantification, superlatives and comparisons, aggregation functions where most other systems cannot deal with and so difficult to go on.

C. Approaches based on graph exploration

Graph exploration approaches such as Top-k exploration [12], process a question query by selecting a basis entities and processing through the knowledge base graph using matching with the natural

language terms. Like any other graph exploration algorithm, this approach is limited to the depth of the graph extension over large amount of linked data. Implemented heuristic allows to efficiently explore graph even without the knowledge of the schema. Typical examples of this approach are Treo Question Answering system, Top-kexploration approach and the approach by Ngonga et al[13]. Quantified question is another challenge for the graph exploration approach as long as semantic connection doesn't contain trivial evidence of filtering or quantifying.

1) ***Treo:*** Treo, Top-k exploration and the approach by Ngonga et al, interpret a natural language question by mapping elements of the question to entities from the knowledge base, and then proceeding from these pivot elements to navigate the graph, seeking to connect the entities to yield a connected query. Another system is gAnswer [6] which is a graph-driven Question Answering system that processes questions in two stages. First, based on the dependency parse of the question, a graph is build that represents the semantic structure of the question. Second, this graph is matched with subgraphs in the RDF dataset. Disambiguation takes place when evaluating subgraph matches. The system achieves real-time performance, requiring an average of 972 milliseconds to answer a question

D. Approaches based on ontology based

This approach is based an ontology and takes queries expressed in Natural Language (NL) as input. After executing SPARQL query over one or more Knowledge Bases like DBpedia, Wikipedia etc it returns answer. This approach saves users from the complex structures of the ontology. The system which uses ontology based approach, matches linguistic structures to semantic triples and tries to map between elements of the query and resources or predicates. Main challenging fact of this approach is the mapping query element and with the DBpedia ontology.

1) ***PowerAqua[14]:*** Power Aqua is a system which focuses on querying in multiple dataset on the semantic web. This grammar entries. Next, syntactic and semantic composition rules is applied to build a derived tree together. The syntactic rules are LTAG operations and semantic composition rules are parallel operations on DUDE's. When all arguments are filled, the DUDE communicates with UDRS. Then these are system is one of the first Question Answering system which targets on semantic web data. Power Aqua is an ontology based Question Answering system. Power Aqua is for using multiple ontologies and knowledge bases to answer the queries in multi domain environment.

Dataset: DBpedia

Working procedure: This is the first step. This linguistic analysis is done using a tool named GATE to detect the question type. This tool is also used to translate the natural language question into a triple based representation which is called query triples. These triples are then further processed to match them to ontology compliant triples, or Onto-Triples, from which an answer can be derived by applying merging and ranking techniques. Thus, the data model is triple based, namely it takes the form of a binary relation model and expresses statements (subject, predicate, object) in the same form as the RDF-based knowledge representation formalisms for the SW, such as RDF or OWL. The second step then searches for candidate instantiations of the occurring terms that means detecting ontologies. In this step a set of output tables contains matching semantic elements for each term in the query triples. These mappings used in turning query triples into DBpedia ontology triples.

Architecture: The Architecture is shown through the flow chart in figure 4

Advantages: Power Aqua's main strength is that it locates and integrates information from different, heterogeneous semantic resources, relying on query disambiguation, and ranking and fusion of answers.
Disadvantages: Its main weakness is that due to limitations in GATE this system cannot deal with aggregation, i.e. counting (e.g. How many), comparisons type question (such as larger than or more than), and superlatives (such as the largest and the most).

2) ***AquaLog [15]:*** Aqualog is a famous system which is based on ontology and it uses a gate for parsing question.

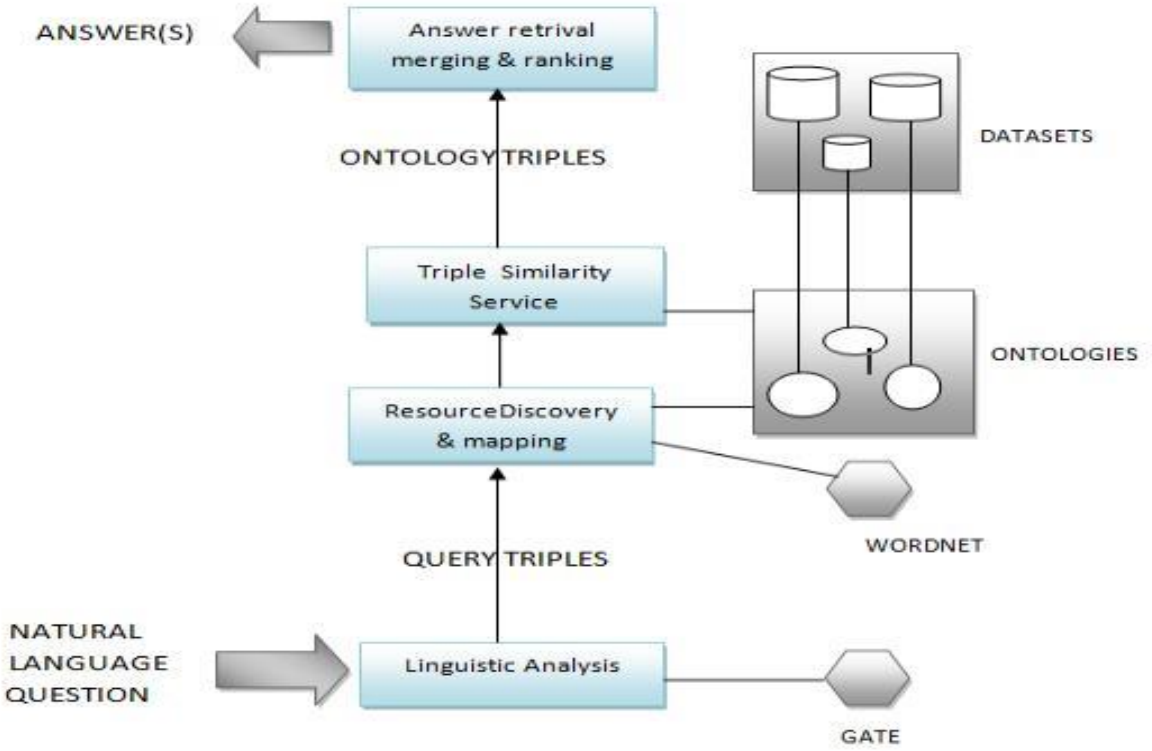

Figure 4: Architecture of Power Aqua system.

Working procedure: This system uses GATE NLP platform, string matrices algorithm, WordNet and novel ontology-based similarity services for relations and classes to make sense of user queries in respect of the target knowledge base. It is used to improve the performance of the system over time, in response to the particular community jargon used by the end users. Mapping the NL input query to the Query-Triple is the task of The Linguistic Component. AquaLog uses the GATE infrastructure and resources to parse the question as a part of the Linguistic Component. Standard GATE API is used for communication between AquaLog and GATE. After executing GATE controller a set of syntactic annotations associated with the input query are returned. These annotations include information about sentences, tokens, nouns and verbs. The set of annotations can be extended by identifying terms, relations, question indicators (which/who/when, etc.) and patterns or types of questions.

Advantages: AquaLog can query directly. This system is based on the premise that the semantic web will benefit from the availability of natural language query interfaces, which allow users to query semantic markup viewed as a knowledge base. AquaLog is portable because its architecture is completely independent of specific ontologies and knowledge representation systems.

Result: They collected in total 76 different questions, 37 of which were handled by AquaLog, i.e., 48.68% of the total. This provides a pretty good result though no linguistic restrictions were imposed on the questions.

E. Approaches based on template-based

Using 2 steps this template based approach retrieves answer from the knowledge bases. Their approach specify some SPARQL query template based on which the template based systems works. At first, they analyze questions linguistically and construct pseudo-query template. Secondly, matching the NL as lexical units with query dataset's elements initiates the template. LODQA [16] and TBSL [17] are template based systems.

1) ***LODQA [16]:*** The popularity of Linked Open Data (LOD) is growing rapidly nowadays and so more and more heterogeneous data sets are being integrated, which makes users to use a complex query language e.g. SPARQL. SPARQL helps to search on the data sets. As SPARQL is not an easy task to write even for experienced users, many groups are developing assistive methods, e.g., visual SPARQL editor. In the Linked Open Data Question-Answering (LODQA) system, we focus on natural language as a human-friendly representational means of search queries. It would obviously be very easy if search queries expressed in natural language could be converted to SPARQL queries. LODQA system focuses on developing a natural language query processing system as an open source project.

Dataset: Open dataset (WordNet, Wikipedia etc.)

Flow chart of LODQA System: Here in the figure 5 below the flow chart of LODQA system is shown:
Disadvantages: The templates correspond to the linguistic structure of the question, thus failing if there is any structural mismatch between natural language and dataset.

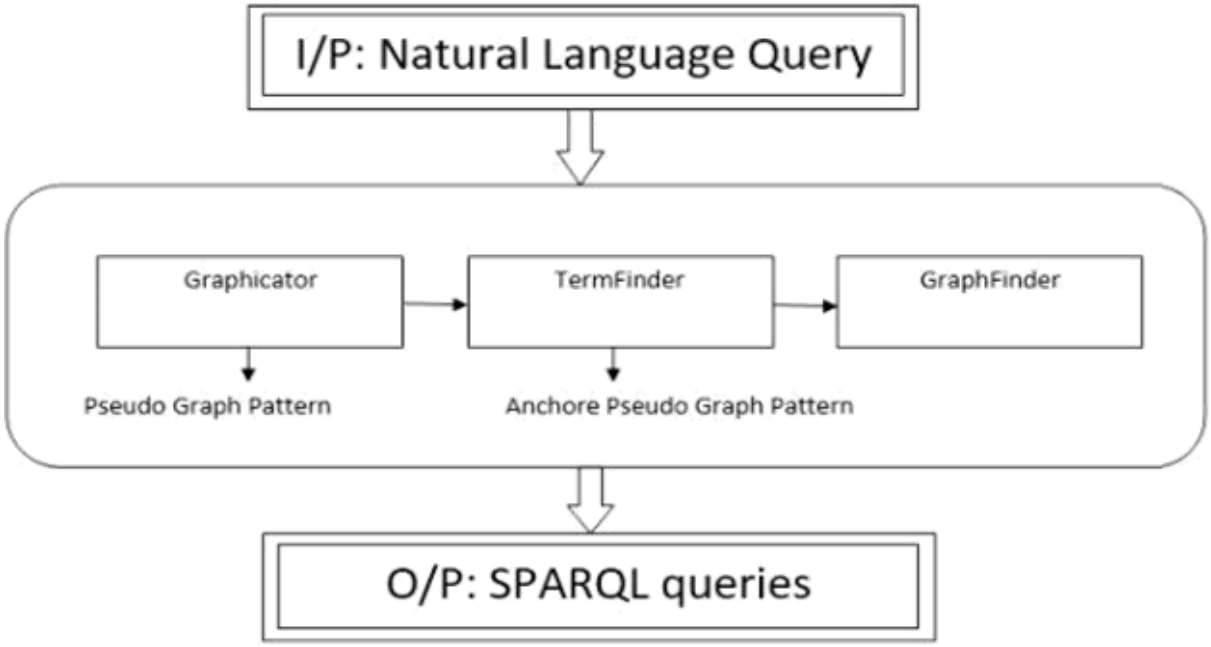

Figure 5: Architecture of LODQA System

F. Other approaches

1) ***CASIA [18]:*** is a Question Answering over Linked Data, which focuses on construct a bridge between the users and the Linked Data. It implements a pipeline consisting of question analysis, resource mapping and SPARQL generation. More specifically, the system first transforms and represents natural language questions as a set of query triples of the form ¡subject, predicate, object¿, based on a shallow and deep linguistic analysis. Second, it instantiates these query triples with corresponding resources from DBpedia, resulting in ontology triples. Third, based on the ontology triples and question type, SPARQL queries are constructed. Finally, the candidate queries are validated and ranked, and the best query is selected.
2) ***Intui2 [19]:*** A Prototype System for Question Answering over Linked Data Intui2 is a prototype system for Question Answering over linked data that can answer natural language questions with respect to a given RDF dataset by analyzing the questions in term of the syntactic constituents (synfragments) they are composed of. Syntactically, a synfragment corresponds to a subtree of the syntactic parse tree of the question, and semantically, it is a minimal span of text that can be interpreted as a concept URI, an RDF triple or a complex RDF query. These synfragments are then compositionally combined to an interpretation of the whole input question.

3) ***RTV [20]:*** The RTV system implements lexical semantic modeling and statistical inferences within a complex architecture that decomposes the natural language interpretation task into three stages. Firstly, the selection of main information from the question (i.e. predicate, arguments and properties). Secondly, the location of the main information in the ontology through joint disambiguation of all candidates, and finally, the compilation of the final query against RDF triples. This architecture uses a Hidden Markov Model (HMM) for the selection of the proper ontological triples according to the graph nature of RDF. Particularly for each query, an HMM model is produced whose Viterbi solution is the extensive joint disambiguation across the sentence elements.

4) ***Xser [21]:*** Xser is a Question Answering system over Linked Data (DBpedia), converting user's natural language questions into structured queries. There are two challenges involved: recognizing users query intention and mapping the involved semantic items against a given knowledge base (KB), which will be in turn assembled into a structured query. Authors propose an efficient pipeline framework to model a user's query intention as a phrase level dependency DAG which is then instantiated according to a given KB to construct the final structured query. They evaluate the approach on the QALD-4 test dataset and achieve an F-measure score of 0.72, an average precision of 0.72 and an average recall of 0.71 over 50 questions.

5) ***YodaQA[22]:*** YodaQA is a Question Answering system over Linked Data. The QA task is implemented in YodaQA as a pipeline that transforms the question to a set of answers by applying a variety of analysis engines and annotators. It is composed from largely independent modules, allowing easy extension with better algorithms or novel approaches, while as a fundamental principle all modules share a common end-to end pipeline. The YodaQA pipeline is implemented mainly in Java, using the Apache UIMA framework. YodaQA represents each artifact as a separate UIMA CAS, allowing easy parallelization and straightforward leverage of pre-existing NLP UIMA components; as a corollary, authors compartmentalize different tasks to interchangeable UIMA annotators. Extensive support tooling is included within the package [22].

6) ***Watson:*** Watson/DeepQA is a software architecture for deep content analysis and evidence based reasoning. The DeepQA architecture views the problem of Automatic Question Answering as a massively parallel hypothesis generation and evaluation task. As a result DeepQA is not just an answering system rather it can be viewed as a system that performs differential diagnosis: it generates a wide range of possibilities and for each develops a level of confidence by gathering, analyzing and assessing evidence-based on available data. With a question, a topic, a case or a set of related questions, DeepQA finds the important concepts and relations in the input language, builds a representation of the users information need and then through search generates many possible responses. For each possible response it provides independent and competing threads that gather, evaluate and combine different types of evidence from structured and unstructured sources. It can deliver a ranked list of responses each associated with an Evidence Profile describing the supporting evidence and how it was weighted by DeepQAs internal algorithms [23].

V. EVALUATION MEASURES

As we have discussed about different systems so far, we have seen that these systems approach based on modules. So there are different evaluation results systems.

QALD is a series of evaluation campaigns that provide a benchmark for comparing different approaches and systems

1) Get a picture of their strength and shortcomings

2) Gain insight into how we can develop approaches that deal with Semantic Web data as a knowledge source

QALD-1 @ ESWC 2011
QALD-2 @ ESWC 2012
QALD-3 @ CLEF 2013
QALD-4 @ CLEF 2014 QA track
QALD-5 @ CLEF 2015 QA track
QALD-6 @ CLEF 2016
QALD-7 @ CLEF 2017

According to test on different QALD test set we have different results. For example, we will discuss the result on different system. There are four predicted conditions which should be known to us true positive, true negative, false positive and false negative.

1) **True Positive:** True positive measures the proportion of positives that are correctly identified as positive.

2) **True Negative:** True negative measures the proportion of negatives that are correctly identified negative.

3) **False Positive:** False positive measures the proportion of positives that are incorrectly identified as positive.

4) **False Negative:** False negative: False negative measures the proportion of negatives that are incorrectly identified as negative.

$$Recall(q) = \frac{Number\ of\ correct\ system\ answers\ for\ (q)}{Number\ of\ gold\ standard\ answers\ for\ (q)} \quad (1)$$

$$Precision(q) = \frac{Number\ of\ correct\ system\ answers\ for\ (q)}{Number\ of\ system\ answers\ for\ (q)} \quad (2)$$

$$F\ Measure(q) = \frac{2\ Precision(q)\ Recall\ (q)}{Precision(q) + Recall\ (q)} \quad (3)$$

A. QALD-3

Here another example is shown in Table II. The Participating systems are squall2sparql, CASIA, Scalewelis, **RTV**, intui2, **SWIP** and here **QALD-3** data set is used for DBpedia test set.

Table II: Results for DBpedia test set

System	Total	processed	Right	partially	Recall	precision	F-1
Squal2sparql	90	90	80	13	0.88	0.93	0.90
CASIA	99	52	29	8	0.36	0.35	0.36
Scalewelis	99	70	32	1	0.33	0.33	0.33
RTV	99	55	30	4	0.34	0.32	0.33
Intui2	99	99	28	4	0.32	0.32	0.32
SWIP	99	21	15	2	0.16	0.17	0.17

B. QALD-4 with two tasks apart

Here another example is shown in Table III. The Participating systems are Xser, gAnswer, CASIA, intui3, ISOFT, GFMed, RO FII, POMELO and here **QALD-4** data set is used for test.

C. QALD-6

QALD-6 is the sixth in a series of evaluation campaigns on Question Answering over linked data, with a strong emphasis on multilingualism and hybrid approaches using information from both structured and unstructured data. Here is a short analysis of the dataset:

Table III: Results for Task 1: Multilingual Question Answering over DBpedia

	Total	Proc	Right	part.	Recall	precision	F-mearure
Xser	50	40	34	6	0.71	0.72	0.72
gAnswer	50	25	16	4	0.37	0.37	0.37
CASIA	50	26	25	4	.40	0.32	0.36
Intui3	50	33	10	4	0.25	0.23	0.24
ISOFT	50	28	10	3	0.26	0.21	0.23
RO-FII	50	50	6	0	0.12	0.12	0.12

Table IV: Results for Task 2: Biomedical Question Answering over interlinked data

	Total	Proc	Right	part.	Recall	presion	F-mearure
GFMed	25	25	24	1	0.99	1.0	0.99
POMELO	25	25	19	3	0.87	0.82	0.85
ROF-II	25	25	4	0	0.16	0.16	0.6

QALD-6 QUESTIONS DISTRIBUTION

1) Who? - 21 Questions
2) What? - 22 Questions
3) Which? - 12 Questions
4) How many? - 21 Questions
5) Give me? - 10 Questions
6) When? - 6 Questions
7) Where? - 3 Questions
8) In Which? - 9 Questions

Column descriptions:

1) The first column the language constitutes late submissions (i.e.after the official end of the test phase).
2) Processed states for how many of the questions the system provided an answer.
3) Recall, Precision and F-1 report the macro-measures with respect to the number of processed questions.
4) F-1 Global in addition reports the macro F-1 measure with respect to the total number of questions

Multilingual Question Answering over DBpedia

Table V: Sorted by global F-1 score (over all questions)

		Processed	Recall	Precision	F-1	F-1 Global
CANaLI	(en)	100	0.89	0.89	0.89	0.89
UTQA	(en)	100	.69	.82	.75	.75
KWGAnswer	(en)	100	0.59	0.85	0.70	0.70
UTQA	(es)	100	0.62	0.76	0.68	0.68
UTQA	(fa)	100	0.61	0.70	0.65	0.65
NbFramework	(en)	63	0.85	0.87	0.86	0.54
SemGraphQA	(en)	100	0.25	0.70	0.37	0.37
PersianQA*	(fa)	100	0.19	0.91	0.31	0.31
UIQA	(en)	44	0.63	0.54	0.58	0.25

VI. CONCLUSION

We presented the overall idea of the semantic web, linked data, RDF triple, Question Answering, Natural Language Processing and Question Answering over Linked data. We discussed the motivation and goal behind this article as well as showed our contribution in their search area. As the available dataset is huge and it is really very difficult to identify the data from these huge datasets, systems have to face many challenges to build. These different challenges like mapping with vocabulary element to natural language, multilingualism, data heterogeneity etc. have been discussed.
We elaborately discussed the Question Answering (QA) system, QA system domain, different tasks involved in QA system. After that we discussed about different approaches to QA system over linked data and existing system based on these approaches. The comparison between different systems shows precision, recall and F-1 score tested on dataset or fixed questions with a number of users. We evaluating these results, found that the QA system can yet to be developed with more precision, recall and F-1 score than the existing, thus we initiated our research problem with the research corpus for our future work.

ACKNOWLEDGMENT

The authors would like to thank those who made this literature review work possible. We owe our deepest gratitude to Mr. Mohammad Moinul Hoque, Associate Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, whose encouragement, guidance and support from the initial level to the end enabled us to development an understanding of the subject and helped us a lot to finish our literature review based research work. We are obviously thankful to the Almighty Allah for keeping us in good health which was very much felt for successful completion of this work.

REFERENCES

A. Bernstein, E. Kaufmann, and C. Kaiser, “Querying the semantic web with ginseng: A guided input natural language search engine,” in 15th Workshop on Information Technologies and Systems, Las Vegas, NV, pp. 112–126, Citeseer, 2005.

A. Gomez´-Perez,´ D. Vila-Suero, E. Montiel-Ponsoda, J. Gracia, and G. Aguado-de Cea, “Guidelines for multilingual linked data,” in Pro-ceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, p. 3, ACM, 2013.

C. Dima, “Intui2: A prototype system for question answering over linked data.,” in CLEF (Working Notes), 2013.

C. Giannone, V. Bellomaria, and R. Basili, “A hmm-based approach to question answering against linked data,” in Proceedings of the Question Answering over Linked Data lab (QALD-3) at CLEF2013 Lecture Notes in Computer Science (to appear), Springer, 2013.

C. Unger and P. Cimiano, “Pythia: Compositional meaning construction for ontology-based question answering on the semantic web,” in Inter-national Conference on Application of Natural Language to Information Systems, pp. 153–160, Springer, 2011.

C. Unger, A. Freitas, and P. Cimiano, “An introduction to question answering over linked data,” in Reasoning Web International Summer School, pp. 100–140, Springer, 2014.

C. Unger, L. Buhmann,¨ J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano, “Template-based question answering over rdf data,” in Proceedings of the 21st international conference on World Wide Web, pp. 639–648, ACM, 2012.

D. Damljanovic, M. Agatonovic, and H. Cunningham, “Freya: An interactive way of querying linked data using natural language,” in Extended Semantic Web Conference, pp. 125–138, Springer, 2011.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al., “Building watson: An overview of the deepqa project,” AI magazine, vol. 31, no. 3, pp. 59–79, 2010.

D. Molla´ and J. L. Vicedo, “Question answering in restricted domains: An overview,” Computational Linguistics, vol. 33, no. 1, pp. 41–61, 2007.

D. T. Tran, H. Wang, S. Rudolph, and P. Cimiano, "Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data," in Proceedings of the 25th International Conference on Data Engineering (ICDE09), 2009.

F. Gandon, R. Krummenacher, S.-K. Han, and I. Toma, "The resource description framework and its schema," 2010.

K. B. Cohen and J.-D. Kim, "Evaluation of sparql query generation from natural language questions," in Proceedings of the Joint Workshop on NLP&LOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction, pp. 3–7, 2013.

K. Xu, S. Zhang, Y. Feng, and D. Zhao, "Answering natural language questions via phrasal semantic parsing," in Natural Language Process-ing and Chinese Computing, pp. 333–344, Springer, 2014.

P. Baudisˇ and J. Sedivy,` "Modeling of the question answering task in the yodaqa system," in International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 222–228, Springer, 2015.

P. Cimiano, P. Haase, J. Heizmann, and M. Mantel, "Orakel: A portable natural language interface to knowledge bases," 2007.

R. Mervin, "An overview of question answering system," International Journal of Research in Advanced Technology (IJRATE), vol. 1, 2013.

S. He, S. Liu, Y. Chen, G. Zhou, K. Liu, and J. Zhao, "Casia@ qald-3: A question answering system over linked data.," in CLEF (Working Notes), 2013.

T. Berners-Lee, "Linked data http://www.w3.org/designissues,"LinkedData.html, 2006.

T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific american, vol. 284, no. 5, pp. 34–43, 2001.

V. Lopez, C. Unger, P. Cimiano, and E. Motta, "Evaluating question answering over linked data," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 21, pp. 3–13, 2013.

V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," in European Semantic Web Conference, pp. 546–562, Springer, 2005.

V. Lopez, PowerAqua: open question answering on the semantic web. PhD thesis, Open University, 2011.

--0--