Cognitive Functionality based Question Answering System

Ashish Chandiok
Faculty of Engineering
Dayalbagh Educational Institute
Dayalbagh, Agra, UP, India

D. K. Chaturvedi
Faculty of Engineering
Dayalbagh Educational Institute
Dayalbagh, Agra, UP, India

ABSTRACT

Applying cognitive functionality in the artificial system is the prospect of next computing era and is speedily attracting the business and industry. As the volume of information increasing rapidly in the digital world, it also tends to challenge the current search engines to mine more significant and related information in precise and comprehensible manner. A Question Answering (QA) system outdoes the traditional information retrieval search engines in such situations. The paper presents and implements cognitive functionality in question answering systems to mimic human-like performance. The work proposes an architecture which trails human mental and brain like problem-solving procedures to answer questions. An illustration also deliberated and elucidated using a practical implementation and evaluation along with its fundamental cognitive functionality.

General Terms

Cognitive Technologies, Question Answering System

Keywords

Cognitive Functionality, Question Answering System, Natural Language Processing, Information Retrieval, Information Extraction

1. INTRODUCTION

In the last two decades, a huge amount of focus is on developing a system with the cognitive function to have human-like interaction. This era represents the start of developing a computer-based artificial system which will interact with people in natural language. But then again this leads us to challenges such as how to make computers understand the grammars and meanings of natural languages? It is reasonably clear that with the growth of computers ability to interact and converse with humans in natural language, a question answering system will show a vital role in its progress.

A question answering system explicitly deals with cognitive functions such as understanding the user question, extracting the knowledge info from diverse structured and unstructured data sources, illustrating logical relations among facts, creating candidate answers and choosing the individual best response from the applicable set of candidate answers. Thus, Question Answering System (QAS) provides better and concise answers. On the other hand, search engine system returns a list of documents/links which may or may not comprise the answer. Also, the answer obtained from such systems does not depict a brief answer. So, Question answering systems are a better choice for getting answers for user queries in comparison to traditional search engines. Therefore it motivates to review the past research and developments in Question Answering System (QAS) to show the progress and analyze how this paper will improve the growth of such system.

1.1 Related Works

In the early years of 1960 to 1980, the development of Question Answering Systems mostly focuses on rule-based query understanding and answering. In these systems, the rules are physically recognized and then programmed into computers. Nevertheless, it clearly, leads to certain unwanted limitations on the system performance and outcome. In the later decade of 1980?s, machine learning techniques implemented to find specific patterns in, a question to plan it to particular rules. However, all such systems represent closed domain problems (particular to a specific domain like medical, law, and finance, etc.) and unsuccessful to tackle with open-ended queries. The real world models of such systems are BASEBALL [12] and LUNAR [29]. BASEBALL responded queries on particular leagues of US baseball. On the other hand, LUNAR limited only to the Apollo mission to answer the questions on geological analysis of rocks collection. The high growth of the World Wide Web (WWW) in the 21st century leads to a new type of necessity in creating answers. The requirement of Information Retrieval (IR) and Information extraction (IE) from web-based documents took over and henceforth the progress of web-based question answering systems turns out to be further pertinent. The internet technology makes strong support to advance open-domain question answering systems in contrast to closed domain systems. The pioneer in such open-domain question answering system named as START [15, 16]. The method developed in and around 1992 for answering natural language questions by giving textual extracts and multimedia info mined from the World Wide Web Internet. It attempts to find answer candidates from questions using natural language annotations. Another type of open domain question answering system is LogAnswer [11] that uses 'Theorem Provers' to develop right answers to the queries. The system finds the answer by extracting answers from a logical knowledge representation using precise inference methods. Nowadays, most of QA systems are open domain. The information source is usually Semantic Web, Knowledge Base or World Wide Web rather than a traditional closed database.

Michigan State University implemented AnswerBus. AnswerBus [30] supports multiple languages including English, German, etc. However, the returned result is a URL or a web page which contains the answer. Users have to suffer from discovering answers from resumed URL or web page. QUANTUM [23] executed by Concordia University and it is comparable to AnswerBus. QUANTUM gives in to required data inside a search engine after covering users? necessities. QUANTUM extracts answers by investigating the web pages reverted to the search engine. Watson [10] implemented by IBM is a revolution in question answering system. Watson founded on a parallel framework DeepQA. DeepQA is self-possessed of two key constituents which are IE module and NLP module. NLP component is accountable for question analysis. IE component achieves to discover candidate answers and then assess them. The system returns the best answer. JAVELIN [22]implemented by Carnegie Mellon University. JAVELIN represents a star architecture. Entire subtasks like question analysis, information retrieval, answer extraction and answer computing observed as nodes. A Centre Node recalls such nodes. When the user provides a query, the center node causes suitable scheduling policy to discover the answer by the category of question. QUARK [17] implemented by Toyohashi University of Technology. It is a regular QA system. Question interpreted from natural language into logical language, which is prepared by GEMINI English. SNARK, a prover, is implemented to search answer from Alexandria Digital Library Gazetteer, the CIA Factbook, Semantic Web, etc. With the progress of concept, model and algorithm of QA, certain distinct QA systems such as medical QA system MEANS [1], biologic QA system OntoNLQA [2], traffic QA system Uniponse [19], aviation QA system[18] arise. The AQUALOG [20] permits the user to choose ontology and then ask NL queries concerning the universe of discourse covered by the ontology. ASKMSR [4] implemented an architecture to answer queries on data by retrieving it from a search engine, collect summaries, mine, filter and tile n-grams to get the best answer. For experimental assessments, the authors used the first 500 TREC-9 questions (201-700). AskHERMES [6] is online software for answering clinical questions which allow physicians to enter a question naturally with minimal query formulation and allows physicians to efficiently navigate among all the answer sentences to quickly meet their information needs. In [25], Tapeh et al. create a knowledge-based question answering system for B2C e-commerce to support the user in asking queries for online shopping. PowerAqua [19] supports users in querying and exploring the multiple semantic web resources. QACID [8] is an ontology-based Question Answering system implemented in the CInema Domain. The system permits users to retrieve information from prescribed ontologies by expending as input queries articulated in natural language. The unique characteristic of QACID is the approach used to plug the opening among users? expressiveness in addition to formal knowledge representation. ORAKEL [7] presents a user-centered model for porting natural language interfaces (NLIs) among specific areas efficiently. The model accepts that a domain expert has no background knowledge regarding computational linguistics. So it will accomplish the automatic customization of the NLI to a specific domain. On similar approach, other natural language search engine are Ginseng [3], PANTO [28] and damljanovic2010natural, which can handle guided Input. QAKiS [5] is an open domain QA system following relational patterns to answer queries. In [21, 24, 27, 9, 26, 18, 13], the researcher?s implements ontology-based SPARQL queries to answer factoid questions. The above mention systems still cannot perform human-like inference and information retrieval. The system does not determine the knowledge based on matching confidence values of the query and the stored info. It is essential to implement a cognitive way of answer finding by implementing human-like problem analysis at the starting level, planning information retrieval and reaching the goal of finding the best answer. These computations are done on latest technologies of cognitive computing through interactive human behavior and decision making. The proposed system performs the human-like functionality of reactive, deliberative and reflective action to reach the outcome. The architecture uses natural language processing and semantic web to analyze and classify query. Information retrieval tools based vector space model to get information. Information extraction and cognitive tools used for decision making and finding the best answer. The vital throughput in evolving open-domain question answering system took abode from end to end research by IBM Research in the year 2007. It started its early ground effort based on its comprehensive QA system PIQUANT. This new QA system named IBM Watson based on a utterly modern architecture entitled DeepQA [14] describing several phases of exploration in a handling pipeline. It is capable of creating multiple candidate answers for a query and allocating scores to these solutions based on evidence along various dimensions. The DeepQA also trains its statistical machine learning algorithms on previous question sets and their particular answers in a direction to increase its accuracy. IBM unquestionably is the industry frontrunner in Cognitive Computing, and it quaked the world when IBM Watson won the Jeopardy! Challenge in 2011 overhauling the world champions at that time. IBM Watson today supports a lot of experts of finance, healthcare, cancer research, customer care and various industries, in decision making, information extraction, and pattern recognition using its cognitive skills. Cognitive Computing is the digital model illustration of human thought process or problem-solving. With the help of cognitive computing methods more contextual, comprehensive information extracted, and dissimilar relationships among different facts in the knowledge base result to attain precise answers. Numerous ambiguities and natural language complications also resolved.

1.2 Aim of the Paper

The primary purpose of the paper is to show by following points:

(1) To review the past question answering system, and find limitations of the prior methods.

(2) To propose a novel architecture for cognitive functionality based question answering using natural language processing (NLP) for question understanding and classification, Information Retrieval (IR) for document/concept retrieval and processing, Information Extraction (IE) for answer extraction, filtering and ordering. Human-like confidence weight based answer selection. Cognitive application of rank based best answer choice.

(3) To implement and evaluate the proposed system for performance.

The paper consists of the following sections. Section 2 explains general question answering system. Also, Section 2 proposes the cognitive functionality based system for answering the question as we humans do. Section 3 layouts the method for developing such systems. Section 4 offers the results by evaluating the system performance and also discusses the salient features of such systems. Section 5 is showing the conclusion of the work.

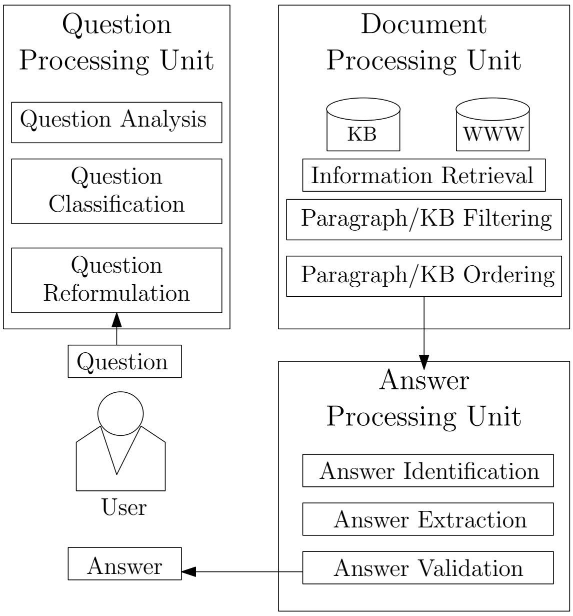

Fig. 1. General Question Answering System

2. BACKGROUND

The Figure 1 illustrates, a unique Q/A system comprising of three components. All of these parts have a core constituent besides other supplemental mechanisms. The three parts are Question Processing Unit, Document Processing Unit, and the Answer Processing Unit, with Question Classification and understanding, Information Retrieval, and Answer Extraction as their fundamental workings respectively. Question processing unit recognizes the critical focus of the question and categorizes the type of query using natural language processing tools. It also catches the answer type anticipated and then develops various semantically corresponding tokens from the query. Document processing unit performs recovery of facts using information retrieval (IR) tools to select a document from the web, so that system filters and correct orders paragraphs to generate precise answers. The last component in a Q/A system is the Answer Processing unit. It uses information extraction tools to extract answers from the selected paragraphs and create the conventional sense of question outcome. Typically, the following situation(s) occur in a Q/A system (also defined explicitly in Figure 1):

(1) At first instance, the user inputs a question to the Q/A system input unit.

(2) Next, in the *Question Processing Module*, the inner component, Question Analysis determines the prime focus of the input question, which enhances the accuracy of Q/A system.

(3) *Question Classification* performs a significant part in the Q/A system by classifying the Question Type (QT) and the Expected Answer Type (EAT).

(4) In *Question Reformulation* stage, the query is reshaped by mounting the query and transferring it to the Document Processing Unit.

(5) The *Information Retrieval* element implements retrieval of the relevant documents based on the essential keywords that occur inside the question.

(6) *Paragraph Filtering* element recovers the relevant documents, filters and condenses them into summarized paragraphs expected to encompass the answer.

(7) In the next part, *Paragraph Ordering* achieved on these filtered paragraphs in a sequence according to the importance and passed to the Answer processing unit.

(8) Based on the answer type and further information recognizing techniques, the system accomplishes *Answer Identification*.

(9) Achieve *Answer Extraction* and *Validation*, a fixed group of heuristics may be defined so that only the appropriate word or phrase (answer) gets extracted.

3. PROPOSED COGNITIVE FUNCTIONALITY BASED ARCHITECTURE FOR QUESTION ANSWERING SYSTEM

Cognitive functionality as shown in Figure 2 support that question answering is a practical human-like skill which can acquire by imitation and repetition of gaining knowledge and responding to queries. The question answering ability follows four crucial phase of cognitive functionality.

(1) Understand and classify the question using Natural Language Processing tools (NLP) and Semantic Web (SW) tools.

(2) Acquire Recall, and Identify information to recognize knowledge about the domain for answering the question using Information Retrieval (IR) tools.

(3) Apply extraction, filtering, and organization of answers for a particular question using Information Extraction (IE) tools.

(4) Analyze, synthesize and evaluate answers to reach the best answer for a particular question using Statistical and Decision-making tools.

4. METHOD

4.1 Understand and classify the question using Natural Language Processing tools (NLP) and Semantic Web (SW) tools:

The first stage in cognitive functionality is to understand the problem by using Natural Language Processing and web for obtaining correct question format, question type, and answer type.
Let us take an example considering a simple question to illustrate the method:
Sample Question: "*What is the capital of medieval India?*"

(1) *Tokenization*: What— is— the— capital— of— medieval— India—?

(2) *POS Tagger*: **WP** What **VBZ** is **DT** the **NN** capital **IN** of **JJ** medieval **NNP** India?

Where-**WP**: wh-pronoun, **VBZ**: Verb, 3rd ps. Sing. Present, **DT**: Determiner, **NN**: Singular Noun, **IN**: Proposition, **JJ**: Adjective, **NNP**: Proper Singular Noun.

(3) *Shallow Parsing (Chunking)*:**NP** [What] **VP** [is] **NP** [the capital] **PP** [of] **NP** [medieval India]?

Where- **NP**: Noun Phrase, **VP**: Verb Phrase, **PP**: Prepositional Phrase, **ADVP**: Adverb Phrase

(4) *Name Entity Recognition*: What is the capital of medieval [GPE India]?
Where GPE: Geo-Political Entity

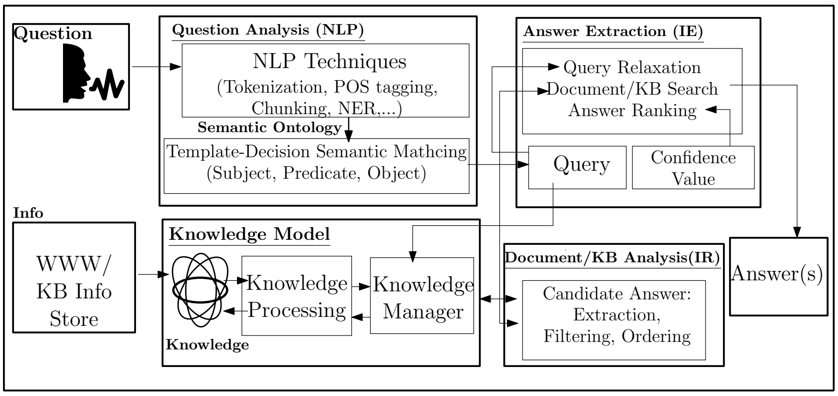

Fig. 2. Proposed Cognitive Functionality Based QA System Architecture

The above done Natural Language Processing (NLP) and applying Heuristic Rules help to recognize the Question (Q), Question Type (QT), and Answer Type (AT).

—Question (Q): What, Question Type (QT): Factoid, Answer Type (AT): Location

Semantic Web helps to determine the Subject (S), Subject Attribute (SA), Predicate (P), and Predicate Attribute (PA). This outcome forms the keywords for finding information from documents in the next stage.

—Subject (S): India, Subject Attribute (SA): Medieval, Predicate (P): Capital.

So, the query vector formed of the attributes having [India, Medieval, Capital]

4.2 Acquire Recall, and Identify information to recognize Knowledge about the domain for answering the question using Information Retrieval (IR) tool:

The information retrieval chooses the documents from the World Wide Web or other local sources. The documents are unstructured textual information having multiple paragraphs. The task of information retrieval tools is to select document which matches the query.
In the document processing unit, firstly the primary natural language processing gets implemented. The documents transformed into documents vector depiction by using tokenization and stemming process. Consequently, evaluation is lead between query vectors to the document vectors. These procedure outcomes a set of candidate answers. If the keyword occurs in the documents, the marking is 1 (one) as a numerical value. On the other hand, it has mark as 0 (zero) value. The indexing outcome turns out to be the tf (document term frequency) value, and idf (inverse document frequency) will then calculated by using the formulas:

$$tf = \frac{n_f}{N_T} \tag{1}$$

Where n_f denotes the number of times term appears in particular document. Also, N_T represents the total number of words in the document.

$$idf = \log \frac{N}{tf} \tag{2}$$

The outcome denotes a feature vector as a sparse matrix calculated by multiplying tfandidf. After calculation of the $tf \times idf$ score for each term in the document $d = \{d_1, d_2, d_3, \ldots\}$ and the query input, the next step is to create Vector Space Model to find out Cosine Similarity (CS) between d_i and denoting the matching between the query and respective document. From each document and query derive a vector. In a vector space model, the group of documents in a pool then is observed as vectors set holding individual $tf \times idf$ values as $d_i\,[n]\,, q_j\,[n]$ for the terms in a vector space. Each term of either query or the document will have its particular axis. Using the formula given underneath we can discover out the similarity among any two documents.

$$CS\left(d_i, q_j\right) = \frac{d_i \bullet q_j}{\|d_i\| \times \|d_j\|} \tag{3}$$

where $d_i \bullet q_j = d_i\,[0] \times q_j\,[0] + \ldots + d_i\,[n] \times q_j\,[n]$
$\|d_1\| = \sqrt{d_i[0]^2 + d_i[1]^2 + \cdots d_i[n]^2}$
$\|q_j\| = \sqrt{q_j[0]^2 + q_j[1]^2 + \cdots q_j[n]^2}$

4.3 Apply extraction, filtering, and organization of answers for a particular question using Information Extraction (IE) tools.

The selected documents sorted by similarity score values for the input query. Based on the similarity values from the previous module, documents ranked with the highest similarity values. From the particular document, answer candidates extracted as paragraphs or short sentences. Let Documents be $D = \{d_1, d_2, \ldots, d_n\}$ and Query $Q = \{q\}$
So, Similarity value $SV\,[d\,(i)] = sim\,(d_i, q)$
Sorted Rank List for document match is $R_{d(i)} = Sort\,(SV\,[d\,(i)])$
Select Answer candidates: $AC_{d(i)} = (s, p) \in d_i$
where s is short sentences and p is paragraph

4.4 Answer Selection

Top-ranked filtered and organized answer candidates achieved with NER tagging to generate the best K candidate answers based on Confidence Value(R). By having the top candidate list of responses, the most repeatedly happening answer is nominated to be the best answer. Best K candidate answers are: $B\,(k) = \max_k R\left(AC_{d(i)}\right)$

5. RESULT AND DISCUSSION

Each question entered into the system goes through several processes; they are query processing (tokenization, POS tagging, stopword removal, question checking for question type and answer, reformulation). Document processing (tokenization, stemming, matching keyword between the query reformulation (by getting subject, subject attribute, predicate, predicate attribute) and document) using vector space model with tf*idf scores. Answer extraction, filtering and ordering to have a list of candidate outcomes. In the end, answer selection gives the final score which derives from vector space design that is sequenced from the uppermost to the lowermost so that it provides the recall and precision values. The individual answer that has the maximum score is the best answer. The system provides the particular answer to the user.
50 questions related to country India is selected manually as a query dataset comprising factoid questions like what, when, where, why, who, and how. The questions are taken randomly and entered into the system to evaluate the system performance. The dataset is from the Wikipedia site having information of India. The evaluation of the system is based on the parameters like Precision@K (P@K), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).

(1) **Precision@K**: It uses a numerical value K as a rank threshold and determines the % relevant answers from the top k answer candidates while ignoring the rest of answer candidates. Let assume that the value of K=5. Amongst best five answers the instances occur at 1^{st}, 2^{nd}, 3^{rd}, and 4^{th}, then P@5 is 3/5. On the other hand, P@3 is 3/3.

(2) **Mean Average Precision (MAP)**: Let's consider the location of ranks for each applicable candidate answer as K_1, K_2 $\ldots$ K_p. Calculating the Precision@K for each instance and then averaging the instance precision gives us Average Precision (AP). So, for 1st, 2nd and 5th candidate answers are correct then average precision (AP) is 1/3(1/1+2/2 +3/5) representing 0.866, while another case 1st, 3rd, and 5th answer are correct then (AP) is 1/3(1/1+2/3+3/5) representing 0.76. So, Mean Average Precision (MAP) is 1/2 (0.866+0.76) which is equal to 0.813.

(3) **Mean Reciprocal Rank (MRR)**: For computing Mean Reciprocal Rank it takes k rank position of the first correct candidate answer. It calculates the Reciprocal Rank (RR) = 1/K. MRR denotes mean value of all RR by an individual query. For example, MRR = 1/3(1/4 +1/2+1/3) when the three queries have the first correct candidate answer at the location 4th, 1st, and 3rd location with individual Reciprocal rank (RR) as 1/4, 1/2, and 1/3 respectively.

Table 1 shows the system evaluation of the question answering methodology for obtaining candidate answers from the document sources based on user queries. Evaluation is on 50 questions taken and entered into the system for the topic "India". The responses obtained from the list of the appropriate answer is multiple answers. So, it is essential to analyze the performance based on ranks. Thus, the quantitative metrics selected depends on rank-based judgment. The Precision@5 (P@5) of 20 queries is 1, while 19 queries are 0.8, rest of the questions are having 0.6 and 0.4. Around 75% of user inputs are having P@5 0.8 or 1. The Mean Average Precision of the user queries is coming out to be 0.84 representing a good outcome as the value is high. The MRR of the responses is coming out to be 0.9 showing mostly the first candidate answer coming out to be correct. From the outcomes, it denotes that question answering system comprising cognitive functionality is showing a promising result.

Table 1. Evaluation of Proposed System Considering Precision@K, Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR)

Query	P@5	AP	RR	Query	P@5	AP @	RR
1	0.8	0.76	1	26	0.6	0.8	1
2	0.8	0.89	1	27	0.8	0.89	1
3	0.6	0.80	1	28	0.8	0.89	1
4	1	1	1	29	1	0.95	1
5	0.6	0.8	1	30	1	1	1
6	0.6	0.76	1	31	1	1	1
7	0.4	0.59	0.5	32	1	0.95	1
8	0.6	0.8	1	33	0.6	0.76	0.5
9	0.8	0.95	1	34	0.8	0.89	1
10	0.8	0.95	1	35	1	1	1
11	0.8	0.89	1	36	1	1	1
12	0.6	0.5	1	37	1	0.95	1
13	0.8	0.89	1	38	1	0.89	1
14	1	1	1	39	0.8	0.89	1
15	1	1	1	40	0.8	0.89	1
16	1	1	1	41	0.8	0.89	1
17	0.8	0.95	1	42	0.6	0.76	0.5
18	0.6	0.8	0.5	43	1	1	1
19	1	1	1	44	1	1	1
20	1	0.95	1	45	0.8	0.89	1
21	0.8	0.89	1	46	1	1	1
22	1	1	1	47	0.8	0.89	1
23	0.8	0.89	1	48	1	0.95	1
24	0.8	0.89	1	49	0.6	0.8	1
25	0.8	0.89	1	50	1	0.89	1

This is an evaluation table for 50 questions answered by proposed system.

6. CONCLUSION

Cognitive functionality based Question Answering System is logically used to ask query for any real word situation and advance the knowledge which is a straight tactic for the users to obtain the information to any facts irrespective of structure and design in a straightforward manner. The QA system proposed in this work performs human-like cognitive approach using Vector space model method on the user query and document source and generates resemblance values by the use of document and user query vectors. And lastly, the best answer is expressed through candidate answer ranking and NER tagging. Presently this Question answering system answers for the modest questions and resembling the closed domain. It identifies the answer type and chooses the best answer from the candidate answers. In forthcoming works, the QA system will improve to give a response to composite queries. It will also extend to work in the Open domain by directly connected to World Wide Web (WWW). The work can also continue to ask multiple queries in a single instant and get summarized results for all the questions. s unstructured data, it is required to use some sophisticated techniques and approaches.

7. REFERENCES

[1] Asma Ben Abacha and Pierre Zweigenbaum. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information processing & management*, 51(5):570–594, 2015.

[2] Amir H Asiaee, Todd Minning, Prashant Doshi, and Rick L Tarleton. A framework for ontology-based question answering with application to parasite immunology. *Journal of biomedical semantics*, 6(1):31, 2015.

[3] Abraham Bernstein, Esther Kaufmann, Christian Kaiser, and Christoph Kiefer. Ginseng a guided input natural language search engine for querying ontologies. jena user conference 2006. In *In Jena User Conference*. Citeseer, 2006.

[4] Eric Brill, Susan Dumais, and Michele Banko. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 257–264. Association for Computational Linguistics, 2002.

[5] Elena Cabrio, Julien Cojan, Alessio Palmero Aprosio, Bernardo Magnini, Alberto Lavelli, and Fabien Gandon. Qakis: an open domain qa system based on relational patterns. In *International Semantic Web Conference, ISWC 2012*, 2012.

[6] Yonggang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J Cimino, John Ely, and Hong Yu. Askhermes: An online question answering system for complex clinical questions. *Journal of biomedical informatics*, 44(2):277–288, 2011.

[7] Philipp Cimiano, Peter Haase, and Jörg Heizmann. Porting natural language interfaces between domains: an experimental user study with the orakel system. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 180–189. ACM, 2007.

[8] Oscar Ferrández, Rubén Izquierdo, Sergio Ferrández, and José Luis Vicedo. Addressing ontology-based question answering with collections of user queries. *Information Processing & Management*, 45(2):175–188, 2009.

[9] Sébastien Ferré. Squall: a controlled natural language as expressive as sparql 1.1. In *International Conference on Application of Natural Language to Information Systems*, pages 114–125. Springer, 2013.

[10] David A Ferrucci. Introduction to this is watson. *IBM Journal of Research and Development*, 56(3.4):1–1, 2012.

[11] Ulrich Furbach, Ingo Glöckner, Hermann Helbig, and Björn Pelzer. Loganswer - a deduction-based question answering system (system description). In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning*, pages 139–146, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[12] B. F. Green, C. Chomsky, and K. Laughery. Baseball: an automatic question answerer. In *In: Proceedings of the Western Joint Computer Conference*, pages 219–224, New York: Institute of Radio Engineers, NY, USA, 1961.

[13] Shizhu He, Yuanzhe Zhang, Kang Liu, and Jun Zhao. Casia@ v2: A mln-based question answering system over linked data. In *CLEF (Working Notes)*, pages 1249–1259, 2014.

[14] Aditya Kalyanpur, Branimir K Boguraev, Siddharth Patwardhan, J William Murdock, Adam Lally, Chris Welty, John M Prager, Bonaventura Coppola, Achille Fokoue-Nkoutche, Lei Zhang, et al. Structured data and inference in deepqa. *IBM Journal of Research and Development*, 56(3.4):10–1, 2012.

[15] Boris Katz, Gary C. Borchardt, and Sue Felshin. Natural language annotations for question answering. In *FLAIRS Conference*, 2006.

[16] Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Julie Qiaojin Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *NLDB*, 2002.

[17] Keizo Kawata, Hiroyuki Sakai, and Shigeru Masuyama. Quark: A question and answering system using newspaper corpus as a knowledge source. In *NTCIR*, 2002.

[18] Jin-Dong Kim and K Bretonnel Cohen. Natural language query processing for sparql generation: A prototype system for snomed ct. In *Proceedings of biolink*, pages 32–38, 2013.

[19] Vanessa Lopez, Miriam Fernández, Enrico Motta, and Nico Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

[20] Vanessa Lopez, Michele Pasin, and Enrico Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *European Semantic Web Conference*, pages 546–562. Springer, 2005.

[21] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, i don't speak sparql: translating sparql queries into natural language. In *Proceedings of the 22nd international conference on World Wide Web*, pages 977–988. ACM, 2013.

[22] Eric Nyberg, Robert E Frederking, Teruko Mitamura, Matthew W Bilotti, Kerry Hannan, Laurie Hiyakumoto, Jeongwoo Ko, Frank Lin, Lucian Vlad Lita, Vasco Pedro, et al. Javelin i and ii systems at trec 2005. In *TREC*, volume 2, page 1, 2005.

[23] Luc Plamondon, Guy Lapalme, and Leila Kosseim. The quantum question answering system. In *TREC*, 2001.

[24] Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. Swip: a natural language to sparql interface implemented with sparql. In *International Conference on Conceptual Structures*, pages 260–274. Springer, 2014.

[25] Ali Ghobadi Tapeh and Maseud Rahgozar. A knowledge-based question answering system for b2c ecommerce. *Knowledge-Based Systems*, 21(8):946–950, 2008.

[26] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM, 2012.

[27] Christina Unger and Philipp Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *International Conference on Application of Natural Language to Information Systems*, pages 153–160. Springer, 2011.

[28] Chong Wang, Miao Xiong, Qi Zhou, and Yong Yu. Panto: A portable natural language interface to ontologies. In *European Semantic Web Conference*, pages 473–487. Springer, 2007.

[29] William A. Woods. Progress in natural language understanding: an application to lunar geology. In *AFIPS National Computer Conference*, 1973.

[30] Zhiping Zheng. Answerbus question answering system. In *Proceedings of the second international conference on Human Language Technology Research*, pages 399–404. Morgan Kaufmann Publishers Inc., 2002.