ISSN 0005-1055, Automatic Documentation and Mathematical Linguistics, 2012, Vol. 46, No. 3, pp. 138–145. © Allerton Press, Inc., 2012.
Original Russian Text © V.A. Lapshin, 2012, published in Nauchno-Tekhnicheskaya Informatsiya, Seriya 2, 2012, No. 6, pp. 1–9.

Question-Answering Systems: Development and Prospects

V. A. Lapshin

Department of Natural Language Query Processing, ABBYY Co., Moscow, Russia

e-mail: mefrill@yandex.ru

Received March 16, 2012

Abstract—A number of problems that are involved in creating question-answering systems are discussed. A review is provided of the systems of this type that are most popular. The typical architecture of a question-answering system includes a question classification module. Different methods for creating this module are examined in the paper.

Keywords: question-answering systems, DeepQA, Watson, question classification

DOI: 10.3103/S0005105512030053

1. INTRODUCTION

The last decade saw growing interest in developing so-called *question-answering systems*, or QA-systems). A question-answering system is a software module that enables a person to have a natural-language dialogue with a machine. The user asks questions within a software system and it prints out answers that are formulated as grammatical sentences.

The first question-answering systems emerged in the 1960s. Among the most popular examples are the BASEBALL and LUNAR systems. The BASEBALL system enabled a dialogue with a user who was interested in the results of competitions involving the US Baseball League over the previous year. The LUNAR system answered questions regarding the geological analysis of species of rocks that were flown from the lunar surface by the Apollo expeditions. Both systems were quite efficiently implemented and were examples of question-answering systems that were targeted at a certain domain. For example, the LUNAR system was shown at a conference in 1971 that highlighted the problems of lunar studies; it allowed one to receive answers to about 90% of the questions the system was asked.

Some famous software systems that were developed in the 1960s contained question-answering modules as subsystems. Hence, ELIZA[1] contained a question-answering system that basically provided communication with the user as a software module [1].

In the 1970s and 1980s a great many question-answering systems were created that enabled a dialogue with a user in a certain domain; for example, the Unix Consultant software complex answered questions concerning the UNIX operating system. Unix Consultant was based on a rather complex and advanced *knowledge base* that contained information about the UNIX operating system. The knowledge-base interface was implemented as a question-answering system.

All of the above question-answering systems allowed one to answer questions concerning a certain domain, i.e., they were *narrow and specialized* question-answering systems. In the late 1990s at the dawn of the Internet and Web there was an acute awareness of the need for question-answering systems that are not associated with any domain. These are so-called *open-source* question-answering systems. They provided a dialogue on all knowledge areas, for example, one based on partially structured knowledge the Web contains.

At the moment, many question-answering systems exist. The START question-answering system is noteworthy [2]. A comprehensive review of this system in Russian was provided in [3].

Another interesting system that claims to be open-source is SUS [4]. The development of this system goes back 30 years. Various multi-purpose ontologies have been developed in this time. Originally a large database, the SUS system is also an English-language interface and therefore a question-answering system. SUS was discussed in a more detail in [5, 6].

The best-known open-source question-answering systems are OpenEphyra [7] and PIQUANT [8] (the next generation of this system, which is known as Watson, will be discussed in a more detail in Section 3).

Below, questions concerning the creation of open-source question-answering systems will be discussed.

[1] ELIZA simulated a conversation with a doctor based on the methods of so-called *Rogerian therapy* where a doctor refrains from providing guidelines to the patient and instead focuses on creating an atmosphere of mutual understanding and trust.

2. A STUDY PROGRAM ON OPEN-SOURCE QUESTION-ANSWERING SYSTEMS

At the start of the third millennium a group of scientists who were studying text searching developed a study program with basic stages for studying particular methods of creating question-answering systems [9]. We will enumerate its main points.

Question Types

For different question types varying strategies for searching are needed. For example, for the question *"Where was Alexander Pushkin born"?* we need to search a knowledge base for the date of birth of the great Russian poet.[2] For the question *"Why did Vanya rollerskate to McDonalds today?"* another search strategy must be used. One might have to make logical inferences from facts such as *"On June 23 Vanya and Masha arranged to meet outside McDonalds to go rollerskating"* or *"This Sunday Vanya stayed overnight at his friend's in the Kuntsevskaya Metro area so that he had a chance to use his rollerskates."*

All possible questions can be divided into classes according to the search strategy. Designing the systems of these classes is a rather challenging task. We will elaborate on the problems that are posed by question classification in a section below.

Question Processing

The same information can be conveyed in a variety of ways. For example, we can ask *"Who was Alexander Pushkin?"* or *"What is the name of the person who wrote a poem about Eugene Onegin?"* These semantically similar questions need to be considered in the same manner. For this, efficient ways of understanding and processing question semantics need to be in place. It is important that the program identifies questions that are identical in meaning regardless of the words, style, syntactic relationships, and idioms used. We would like to have the question-answering system split complex questions into simple ones and perform a correct analysis of context-dependent phrases and probably ask the user for clarifications in the course of a dialogue.

Context Questions

The meaning of a question is defined not only by its content but also the context it is asked in. The context is information that is remembered in the course of a dialogue with a system user; it may help to eliminate ambiguities. For example, the frequently used expression *"These types of steel are available in the rolling plant"* can be made clearer based on the context of a dialogue (e.g., if different types of steel were discussed in the dialogue).

[2] In this context, "a knowledge base" can mean anything, for example, a non-structured text in Russian indexed into an inverted word list in a search engine.

Knowledge Sources for a Question-Answering System

In order to answer a question, one needs to have access to some knowledge base that contains information about it. Otherwise it is hard (if possible at all) to find a correct answer. An open-source question-answering system usually operates with several knowledge sources where answers are sought regardless of the question asked.

Searches

The correct search for the answer to some question depends on the complexity of a question, its type, context, quality of available information sources, search method, etc. Therefore, particular attention is to be given to studying the methods for distinguishing correct answers based on the knowledge bases that are available. Determining necessary sources in order to distinguish the answer also depends on whether a question is correctly classified.

Formulating Answers

The answer must have a natural appearance, ideally in such a way that the user does not guess it is machine-generated. In some cases, a mere database search will suffice to achieve this. For example, we need to find the name of a person, device, or disease; a numerical value (money exchange rate, length, or size); or a date (*"When was Alexander Pushkin born?"*). But sometimes we have to deal with complex queries; thus, we need particular search algorithms that use different sources followed by the combining of these answers into one. A final answer is to be formulated so that the result looks syntactically natural and is exactly what the user was searching for.

Answering Questions in Real Time

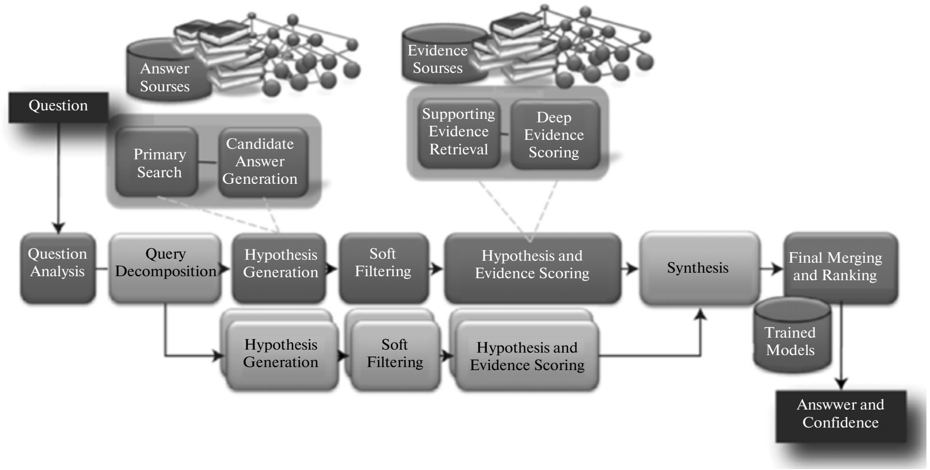
A question-answering system must provide answers in real time, i.e., within several seconds. This condition should be met regardless of the length and complexity of a question, as well as the information source according to which the search is performed.

Multi-Language Queries

This means that one should develop systems to operate and search in different languages (as well as automated translation).

Interactivity

Information retrieved by a question-answering system as an answer more often than not is not exhaus-



The architecture of the open-source question-answering system of the Watson system.

tive. For example, a system may misclassify a question or get it wrong. Therefore, allowances should be made for the system to include clues (corrections). Clues should obviously be implemented as a dialogue (a question—answer sequence).

Reasoning Mechanism (Logical Induction)

Users often ask questions to which the answers do not exist in the available knowledge bases. In order to search for the answers to such questions, a question-answering system should be fitted with a system of logical inference that is based on the facts that are obtained from the available information sources.

QA-System User Profile

User data (interests, style of formulating questions, and other specific information) could contribute to improving the performance of these systems.

In order to encourage research on this issue, a program for running contests at the Text Retrieval Conference (TREC) was developed [10] with corresponding tasks that were formulated at the TREC conferences of 2000—2005 (TREC-9—14). For more detail, see [9]. In fact, contests concerning the evaluation of question-answering system were held at the TREC conference from 1999 to 2007. More detailed information can be found at the TREC conference webpage, which describes the question-answering track [11].

3. THE DeepQA PROJECT

In the middle of the last decade the development of question-answering systems was largely sponsored by the famous IBM Company [12]. In 2005 it launched a project called DeepQA. This is a project on an open-source question-answering system with an extended architecture that allows one to adapt a system to operate in a range of domains.

The major challenge faced by the Deep QA project was to create a software system to play a TV quiz show named Jeopardy!, which is popular in the USA (on Russian television it airs under the name Our Own Game). Three contestants participate; each is given a board that is divided into 30 grids (six lines with 5 grids each). The grids contain the texts of questions that are chosen by the contestants. Initially, the texts of the questions are hidden; as a contestant makes a choice, the grid of a question chosen flashes and the contestants have 5 seconds to give a correct answer. A correct response earns the monetary value of the clue. Each line on the board has a name, i.e., all questions in the line belong to the same category. Categories are designed from scratch for each episode of the TV quiz show, i.e., the set of categories is not fixed. The game itself includes three rounds and a final where the host asks only one question that each of the contestants must give an answer to. They have 30 seconds to do so.

The software program that enables one to play Jeopardy! is called Watson after the founder of IBM, Thomas Watson. In early 2011 the Watson system was used in several sessions of Jeopardy with the best contestants in the entire history of the game participating, i.e., those who ended up winning all of their games. The developers of the program published their experience in the design of the Watson system (and the DeepQA project on the whole) in [13]. Below, we will discuss the architecture of this system as the most successful project ever in the history of the creation of open-source question-answering systems.

***

The architecture of the Watson system is shown in the figure below. The original copy was borrowed from [13]. The major principle behind the organization of the system is its openness to adding new components at all stages of operation. In fact, at each level of question processing, the system uses referees who choose the best variant out of the results of question processing using a variety of algorithms.

Let us discuss the components of the Watson system in slightly more detail.

1. Question Analysis

Question analysis has several constituents:

• **Question classification**. Question classification involves determining its category. There is always a specified set of categories. A hierarchical relationship may exist between a set of categories (more on this in Section 4). The Watson system uses several algorithms for classification, most of which have their own sets of categories.

• **Defining a question's focus and the lexical type of the question**:

A question focus is a part of a question which, replaced by the answer, makes the original question into a sensible sentence. For example, the focus of the question *"When bombarded by electrons, phosphorus radiates electromagnetic power in the same form"* would be the word combination *"in the same form"*. Inserting the answer *"light"* instead, we get the grammatical sentence *"When bombarded by electrons, phosphorus radiates electromagnetic power in the form of light."* Determining a question's focus is important for its further processing and answer generation as well.

The lexical type of answer in the Watson system is a word or a phrase in a question that characterizes the answer type, with this word or phrase being examined with no semantic interpretation whatsoever. For example, in the Chess category the answer to the question *"In the 16th century this move was invented to speed up the game that allowed a player to make several moves without changing the color of the moved pieces"* would be *"castling"* and the lexical answer type would be the word "*move*." Lexical answer types are used in the Watson system as predetermined categories. If, for example, the lexical type of the answer to this question is *"the country"* we can refer to a geographic knowledge base to search in a dictionary of countries.

Establishing relationships. The Watson system also seeks to identify relationships between the lexical elements of a question text. For example, the question *"The Volga River flows through these regions and national republics of the Russian Federation"* specifies relationships such as the Volga River, flows, ?x. We can query the corresponding ontology to obtain the answer to this question. In the process of designing the Watson system, the developers estimated the number of comparable questions from which relationships were identified among the questions of the quiz. It turned out that about 2% of all questions were of this type.

2. Question Decomposition

The Watson system decomposes a question into simpler parts. Given that questions asked in Jeopardy! usually have a complex structure, decomposition is crucial to question processing in the system. In order to identify the parts of a question, the Watson system uses rule-based algorithms and statistical machine-learning algorithms. Each of the identified parts is processed individually. The algorithms are set to operate in parallel.

3. Generating Answer Hypotheses

In order to generate possible answers, knowledge sources are searched. These can be unstructured knowledge, e.g., ordinary web pages, poorly structured knowledge (e.g., Wikipedia articles), as well as structured knowledge (e.g., RDF storage) [14]. Texts in the Watson system are indexed in an ordinary inverted word index. Additional indexing of syntactic and semantic structures that are distinguished in the text analysis is used. There is also a database that is related to the lexical types of words that are defined at the question-analysis stage.

Hypothesis generation comprises two stages: an initial search and the generation of a hypothesis.

During the initial search, different knowledge bases are referred to. As hypotheses are generated from the results of the initial search, these results are transformed into the answer format. The algorithm of this transformation is specific for a knowledge base, for example, for results that are obtained by a search using a "title-oriented" index with the name of a document printed out as the result. With significant results in the storage of RDF-trinities, an expression is transformed into a natural language, etc.

If appropriate hypotheses were not obtained at this stage of question processing, there is no point in going further with this process and the system reports "No results found."

4. Soft Filtering

After a set of hypotheses for a question is designed, so-called "soft filtering" of these hypotheses takes place. Soft filtering entails the use of specialized algorithms that do not require significant resources. All these algorithms can be carried out in parallel, which allows one to decrease the time for hypothesis processing dramatically. The task at this stage of answer processing is to make a shortlist of candidates with answers so that more resource-intensive processing algorithms can be applied to the rest of the hypothesis.

As an example of an algorithm that is launched at the stage of soft filtering, we can mention one that determines the likelihood of a hypothetical answer that corresponds to the lexical type of the answer that was identified for this question at the previous stage of processing.

5. Evidence Scoring

Hypothetical answers that were sifted by soft filtering are further processed by evidence-scoring algorithms. At this rather resource-intensive stage, different knowledge sources are referred to in order to prove that this hypothesis is really the answer to the question that was referred for processing.

This stage is made up of two phases: the phase of obtaining the evidence and the phase of scoring this evidence.

At the stage of obtaining evidence, different knowledge sources are referred to. For example, the text of a hypothetical question is added to the initial question and a knowledge source is then sought using a resulting query. In other words, the answer is once again sought in the context of the question that was asked. Other sources can be referred to such RDF storage. At this stage, answer hypotheses are scored from different sources.

At the evidence-scoring stage, a complex evaluation of evidence obtained for this hypothetical answer at the stage of obtaining evidence takes place. The scoring algorithms determine the consistency of this evidence. The architecture of the DeepQA project makes it easy to add different scoring algorithms for different sets of evidence types. The DeepQA provides a specialized interface for evidence-scoring algorithms and a unified algorithm for operating this evidence. The article [13] gives an example of the processing the question *"He was presidentially pardoned on September 8, 1974,"* which illustrates the system that operates on different scoring algorithms.

6. Final Combination and Ranking

An ordinary document search returns a set of documents that are presumably more relevant to the query. Jeopardy requires that the answer is given as a sensible sentence. The task at this stage of processing is to combine similar hypotheses into sensible answers and then to rank the results in the order of their decreasing evaluation that was obtained at the previous stage of processing.

At the stage of combining answer hypothesis similar hypotheses are combined into one. The same answer can be expressed in a text in different ways. The resulting hypotheses are normalized so that the results of the operation of the system are grammatically correct.

At the stage of ranking the confidence of the answers to the question asked is finally evaluated. The Watson system uses machine-learning algorithms for ranking: the estimates from different sources are processed based on the model that is obtained as the result of learning based on some learning selection of questions and answers.

The Watson system uses distributed processing of UIMA (Unstructured Information Management Architecture) documents [15]. This is a system with an open initial code, which enables one to save the results of document processing in a search index with their further identification with syntactic and semantic relationships. The components of the system can be located on different computers. Therefore, if needed, the Watson system can be easily extended using additional resources.

In order to perform distributed indexing of a preliminary processed-data source corpus, Watson uses the Hadoop system [16]. Algorithms for adding annotations that are inserted into the indexed data by UIMA are easily applicable to the model of parallel computing map-reduction that is provided by the Hadoop system.

The first versions of the Watson system were performed on an ordinary desktop computer with a dual core processor. It took as long as 2 hours to process one question. The modern system is performed on a cluster with 2500 cores. This allows many processing algorithms to operate in parallel. Query processing takes from 3 to 5 seconds.

As discussed above, the DeepQA architecture is designed to be easily adaptable to operate with different domains. Therefore, in 2009 the Watson system was tested on a question answering track from the TREC conference [11]. Experiments were held involving the newly adapted Watson system, as well as the system as it was prepared for Jeopardy!. In its unprepared state the system scored 35% on the accuracy scale and the newly adapted system scored as high as 60%, which was the best result in the history of TREC conferences.

In early 2012, the Watson system was used to solve a range of practical tasks. The press reported at least one such usage, which is to design a question-answering system for the Citigroup financial holding company. The Watson system will have a dialogue with the staff, request additional necessary information and give answers. IBM cooperation makes the Watson system accessible as an Internet service as well, i.e., as a cloud-computing service which is accessed via the Internet. The new service was given the name WAAS (Watson as service).

4. QUESTION CLASSIFICATION

As discussed in the previous section, one of the most important modules of a question-answering system is the question classification module. The task of the module is to compare a question with one of the predetermined classes (categories). This comparison enables one to determine a source of information that needs to be searched for the answer to the question asked, as well to choose an algorithm to perform this search with.

A set of categories that is used to classify questions can be called an *ontology* [17]. Question ontology is usually a hierarchical structure, i.e., a taxonomy whose upper-level concepts appear to be rather peculiar based on the function a question is set to perform.

The problem of arranging queries into a classification was first discussed in [18], which was published in

Question categories by W. Lehnert

Question category	Definition
Causal Antecedent	Why did Vasiliy come to Moscow?
Goal Orientation	Why did Ivan take this book?
Enablement	What does Oleg have to do in order to leave?
Verification	Has Ivan left?
Disjunctive	Were Osya or Kisa here?
Instrumental/Procedural	How did Vasiliy get to Moscow?
Concept Completion	What did Ivan eat?
Expectation	Why didn't Ivan come to Moscow?
Judgmental	What does Vasiliy have to do so that Masha will not leave?
Quantification	How many people gather at this stadium?
Feature Specification	What is the color of Ivan's eyes?
Request	Will you please pass me the salt?

the late 1970s. Its author, Wendy Lehnert, performed a detailed study where the model of presenting semantics of text expressions was developed based on the so-called conceptual dependency theory.

W. Lehnert set forth 13 categories that all questions that can possibly be posed by a user fall into. These categories are identified in the table below, along with some examples of questions that belong to them.

The system of classes that was introduced by W. Lehnert was extended by Arthur Hessser in [19]. Five more categories were added to 13 existing ones: Definition, Example, Interpretation, Assertion, and Comparison. Later, 16 categories were defined in [16].

The above classification is fairly general and is suitable only for initial question processing, for example, to choose an algorithm for query processing. In order to choose data sources in which answers will be sought, it is necessary to define more precise categories. Hence, a more advanced concept system needs to be in place. The specifications of these systems are called ontologies [17]. In other words, an ontology of the classification of questions should be posed at the input of a question-answering system. Classification systems usually use particular ontologies that have only one relationship between classes (a hierarchy). Ontologies with classes that are arranged into a hierarchical structure are called *taxonomies*. Below, we will assume that classes that are used to classify questions in question-answering systems are organized into taxonomies.

The work [21] defined the taxonomy of questions that were used in the TREC-10 track. This taxonomy (see *Appendix*) determines six upper-level categories (so-called "raw" categories) and 50 "fine" classes according to which the classification is performed.

Rule-based and statistical classifiers are usually used as algorithms for question classification. Rule-based classifiers use very simple classification rules: for example, if a questions starts with "What is the rotational velocity?" this question is likely to belong to the "velocity" category. In the above-mentioned paper [21] a statistical classifier was used. In fact, a whole range of works on question classification are available (see [22] for a comprehensive review).

5. CONCLUSIONS

This review examined the current trends in research into question-answering systems. Since the last decade a great effort has been made to study this subject. In the early 21st century this work was mainly encouraged by the program that was formulated in [9], according to which, from 2000 to 2005 competition tasks were proposed at TREC conferences.

Research into question-answering systems continued under the leadership of IBM after program [9] ended. The creation of the Watson system [13] and its subsequent victory in the TV quiz show Jeopardy! at the beginning of 2011 appears to be a great achievement in creating open-source question-answering systems. The present state of question-answering systems enables them to be used in full-blown expert systems that are capable of communicating with users in a natural language.

Rapid development of narrow and specialized question-answering systems is occurring as well. For example, the recently launched WolframAlpha project [23] is an extensive knowledge base that is a natural language interface. From the standpoint of this paper, the WolframAlpha system is a specialized question-answering system.

In our view, the further development of open-source question-answering systems will obviously be based on creating advanced analysis tools of natural languages in order to extract facts from texts written in them.[3]

The structure of these facts should be described by an ontology; a question classification module should correlate a question with a certain ontology. From this point of view, an open-source question-answering system will be a synthesis of a narrow and specialized question-answering system and a system of extracting knowledge from poorly structured uncategorized texts.

[3] From this point of view, a series of articles in [24, chapter 6: Perspectives of developing question-answering systems] might be of interest. The greatest focus in all of them was on developing natural-language analysis tools.

APPENDIX

Taxonomy of questions TREC-10

Question category	Definition
ABBREVIATION	ABBREVIATIONS
abb	abbreviations
exp	complicated abbreviations (expressions)
ENTITY	ENTITY
animal	animal
body	organs of body
color	color
creative	discoveries, books, etc.
currency	currency names
dis.med.	disease and medicine
event	events
food	food
instrument	musical instruments
lang	language
letter	letters (such as a-z)
other	other entity
plant	plant
product	products
religion	religion
sport	sport
substance	elements of the substance and content
symbol	symbols and signs
technique	approaches and methods
term	equivalent terms
vehicle	cars
word	word of speech with special properties
DESCRIPTION	DESCRIPTION AND ABSTRACT CONCEPTS
definition	definition
description	description
manner	the means of action
reason	causes
HUMAN	People
group	groups and organizations
ind	individuals
title	the names of persons
description	description of the persons
LOCATION	LOCATION
city	cities
country	countries
mountain	mountains
other	other locations
state	states
NUMERIC	NUMERAL VALUES
code	postal and other codes
count	the number of
date	dates
distance	the distance measure
money	prices
order	rank order
other	other numbers
period	time period
percent	percent interest
speed	speed
temp	temperature
size	dimensions, areas and volumes
weight	weight

REFERENCES

1. ELIZA. http://en.wikipedia.org/wiki/ELIZA. Accessed Mar. 07, 2012.
2. START. Natural Language Question Answering System. http://start.csail.mit.edu. Accessed Mar. 11, 2012.
3. Nikitin, A. and Raikov, P., Question–Answer Systems. http://yury.name/internet/06ia-seminar.ppt. Accessed Mar. 11, 2012.
4. Cycorp. http://www.cyc.com. Accessed Mar. 11, 2012.
5. Lapshin, V., Cyc System and Its Ontology Library. Ch. 1. *Iskusstv. Intellekt Prinyatie Reshenii* 2010, no. 2.
6. Lapshin, V., Cyc System and Its Ontology Library. Ch. 2. *Iskusstv. Intellekt Prinyatie Reshenii* 2010, no. 3.
7. OpenEphyra. http://sourceforge.net/projects/openephyra. Accessed Mar. 14, 2012.
8. Prager, J.M., Chu-Carroll, J., and Czuba, K., *A Multi-Strategy, Multi-Question Approach to Question Answering* Maybury, M., Ed., Menlo Park: AAAI, 2004.
9. Burger, J., Cardie, C., Chaudhri, V., et al., Issues, Tasks and Program Structures to Roadmap Research in Question and Answering (Q and A), 2000. http://www-nlpir.nist.gov/projects/duc/roadmapping.html
10. Text REtrival Confetence (TREC). http://trec.nist.gov. Accessed Mar. 07, 2012.
11. Question Answering Track. http://trec.nist.gov/data/qamain.html. Accessed Mar. 07, 2012.
12. IBM. http://www.ibm.com. Accessed Mar. 13, 2012.
13. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., et al., Building Watson: An Overview of the DeepQA Project, *AI Magazine* 2010, vol. 31, no. 3, pp. 59–79.
14. RDF.http://www.w3.org/RDF. Accessed Mar. 14, 2012.
15. Ferrucci, D.A. and Lally, A., UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment, *Natural Language Eng.* 2004, vol. 3–4, no. 10, pp. 327–348.
16. Hadoop. http://hadoop.apache.org. Accessed Mar. 14, 2012.
17. Lapshin, V.A., *Ontlogii v komp'yuternykh sistemakh* (Ontologies in Computer Systems), Moscow: Nauchnyi Mir, 2010.
18. Lehnert, W., The Process of Question Answering : *PhD Dissertation*, Research Report, No. 88, 1977.
19. Graesser, A. and Person, N.K., Question Asking during Tutoring, *Amer. Educ. Res. J.* 1994, No. 31, pp. 104–137.
20. Graesser, A., Rus, V., and Cai, Z., Question Classification Schemes, *Proc. 1st Workshop on Question Generation* pp. 8–9, 2008. http://141.225.40.110/16-GraesserEtAl-QG08.pdf
21. Li, X. and Roth, D., Learning Question Classifiers, *Proc. 19th Int. Conf. on Computational Linguistics, (COLING'02)* Stroudsburg, Association for Computational Linguistics, 2002, vol. 1, pp. 1–7.
22. Sundblad, H., Question Classification in Question Answering Systems, *Ph. D. Thesis*, Linkopings Univ., 2007.
23. WolframAlpha. http://www.wolframalpha.com. Accessed Mar. 11, 2012.
24. Strzalkowski, T. and Harabagiu, S., *Advances in Open Domain Question Answering (Text, Speech and Language Technology)* New York: Springer Verlag, 2006.