Intelligent Question – Answering Systems: Review of research

J. Tomljanović*, M. Pavlić ** i M. Ašenbrener Katić***
* Polytechnic of Rijeka
Vukovarska 58, Rijeka, Croatia
** *** Department of Informatics, University of Rijeka,
Radmile Matejčić 2, Rijeka, Croatia
* jasminka.tomljanovic@yahoo.com
** mile.pavlic@ris.hr
*** masenbrener@inf.uniri.hr

Abstract - With the development of ICT the need for automated question-answering systems is becoming increasingly important. Question-answering systems are still under development and experimentation. This paper is an overview of the research area that deals with question-answering systems; it explains the concept of question-answering systems and points out the problems that occur during their development. It also refers to a complex assessment techniques that are necessary when designing such systems. The system described is a real system and so are the test results.

Keywords: question-answering systems, artificial intelligence, technology assessment question-answering systems

I. INTRODUCTION

Along with the development of ICT grows a need for systems that allow the user to ask questions using everyday language and getting fast answers with the smallest quantity of context that is necessary for confirming the answer.

The solution to this problem are question - answering systems (QA systems) (Hirschman et al., 2001). QA systems are a method of retrieving information that answer questions asked in spoken language (Tomljanović et al., 2014).

The success in this field was achieved as part of the Text Retrieval Conference (TREC) (Ittycheriah et al., 2000) which is a series of workshops the mission of which is to retrieve as many information as possible from different fields of research and to backup the research in the field of QA systems.

To answer the question, the system has first to analyze the problem using the context. It finds one or more possible answers offered by the Internet source and offers an answer in an appropriate form. In most cases it is the supporting material that makes the answer understandable to the user or provides some further explanations regarding the answer.

In 1999 TREC started with the evaluation of QA systems that answer the factual questions by consulting the documents contained in TREC collections. A large number of systems successfully combine information retrieval and natural language processing techniques.

One of the groups of methods for knowledge representation is a graphical method for representing knowledge. These methods first transform knowledge into the model, then search knowledge by the model, and then are able to answer the questions. The NOK method (Jakupović et al., 2013), (Pavlić et al., 2013a), (Pavlić et al., 2013b), (Pavlić et al., 2013c) belongs to this group.

This paper will cover the methods which use the existing sources of knowledge registered in different texts.

II. COMPLEXITY ASSESSMENT TECHNIQUES QA SYSTEM

There are two critical research questions in development of QA systems: resources and evaluation. These two questions are closely related; developers need tools to build the system and they need assessment methods (exercises and tests) to evaluate the effectiveness of their system. The introduction of a general evaluation, such as TREC, has created associations interested in the area, and speeded up the research process as well. This is the evaluation with the special emphasis on methods for automatic evaluation.

A. Collection of questions and answers

In order to build a system that answers the questions, scientists need a stream of questions and answers. Ideally, such groups would be often occurring questions, and their answers would be contained in a large collection of documents. Kupiec uses a trivial search for questions as a source of question - answer pairs, as well as the Internet encyclopedia as a collection where to seek the answers (Kupiec, 1993).

A much larger collection is required for efficient use of machine learning and statistical techniques. For example, the syntax of questions and statements is different, and a large collection that explains or narrates usually contains very few questions, so without special settings, the rules based on the collection, will not work for the analysis of questions. The proper analysis of questions requires a large collection of questions and related short answers in order to develop premium features for speech tagging, parsing and questions input. For example, Mann used 2 collections of trivial questions with short answers (Mann, 2001). Auxiliary systems on the Internet include access to the most frequently asked questions as sources of question - answer sets, which are particularly useful in the development of QA systems specialized in certain areas. The number of QA websites is increasing. Good examples are sites that provide tests for foreign languages learning or contain quizzes based on actual events (Roever, 2001; Ushida, 2005).

Additional work in finding and collecting such collections on the Internet would accelerate the progress of the QA systems. The most common source of question - answer sets are multiple choice questions. Such questions are used in standardized tests. MCQs are not as natural as short-answer questions because they are designed to make the evaluation easier. Since the designing of tests is rather expensive, it is difficult to get a collection of materials for research or evaluation.

B. Expansion model

Any collection that contains question - answer sets may be useful. Some types of questions are much easier to be dealt with than the others. Previous researches have focused mainly on simpler questions. At the beginning of TREC assessment, questions were limited to simple factual questions that had the answer in the associated collection of documents. The result showed that the best assessment strategy was a ranked list of proposed correct answers.

The following assessment has increased the complexity of the questions in two dimensions: it allows questions that have no answers and questions with a list of answers (Ittycheriah et al., 2001). This further complexity requires changes in the assessment of the answer and in the construction of the system. To deal with questions that have no answers in the separate collection, systems themselves must measure their reliability regarding the answer.

Questions that require an answer sheet (e.g. a list of countries bordering Croatia) will need to choose an answer from more sentences comprised in one document, or even merge the answer from more documents. Both the former and the latter require the extension of a simple model to simple finding of sentences or parts of the document that give the best answer.

C. Evaluating response problems

The first problem that arises in the assessment is the choice of criteria to be used to assess the answer. *How to evaluate your question answering system every day . . . and still get real work done* (Breck, et al., 2000) calls for six following criteria:

- *relevance:* the answer should answer a given question
- *correctness:* the answer should be factually correct
- *conciseness:* the answer should not contain extraneous or irrelevant information
- *completeness:* the answer should be complete, i.e. partial answer should not get full credit
- *coherence:* the answer should be coherent, so that the questioner can read it easy
- *justification:* the answer should be supplied with sufficient context to allow a reader to determine why this was chosen as an answer to the question

Previous evaluations focused mainly on relevance, although today TREC - QA assessment requires that the answer must be justified by the context and that the size of the answer must be limited (in bytes). This is the first step towards conciseness. Optimization of a single criterion affects the quality of the other. For example, the justification of answers can reduce conciseness. This is exactly why the assessment criteria must depend on the purpose of use, the type of the user who searches for an answer, as well as on the interface.

After the criteria, the assessment procedures occupy the second important position. Experiments carried out during The Eighth TREC Conference introduced human evaluators to read and evaluate each answer. The results obtained showed that the human evaluator was consistent enough to preserve the relative ranking system, which implies that it is actually possible to have only one evaluator.

This has significantly reduced costs, but since there is a need for a human hand, this method does not support the systematic repetition of tests for machine learning. It is always useful to have a human subject dealing with evaluation, and therefore, tests that assess reading comprehension are ideal. In case of using multiple-choice tests, a human evaluator is not required.

D. Automated assessment techniques

Automated grading methods are under development and experimentation. For evaluation of short answers, it is possible to automate the comparison of answers, regardless of whether the answers are provided by the system or by a person, e.g. a student, with answers from the key provided by an expert. Such an assessment is described by Hirschman, Breck, Light,

Burger and Ferro in their book "Automated grading of short answer tests" (Hirschman et al., 2000). Such comparisons, though not as precise as those by human evaluators, nevertheless provide reasonably good results, 93-95 % according to human evaluators, which was good enough for machine learning.

Automated answers and evaluation are of great importance for testing the educational community. Despite the fact that the tests with short answers and essays are better, the multiple choice tests continue to be widely used. Tests where the person himself / herself writes the answers (either short answers or essays) are considered to be too subjective to be used for standardized testing. Recent research in this area (Kukich, 2000) have demonstrated the feasibility of automated methods of assessment, sometimes in combination with only one human evaluator. These results indicate that the automated grading systems are approaching human evaluators. We can already encounter automated assessments and e–learning systems throughout the Internet.

III. EXAMPLES QA SYSTEMS

1. *IBM's statistic system*

Operation of the system that answers the questions is explained using the example of IBM's statistic system that answers the questions (Ittycheriah et al., 2000). The architecture of the system is built around 4 major components, as shown in Figure 1:

- QA Type Classification
- Query expansion/Information Retrieval
- Named Entity Marking
- Answer Selection

The question is input that is classified, as asking for an answer belonging to one of the named entity classes. In addition, the question is presented to the IR engine for query expansion and document retrieval. This engine, given the query, looks at the data base of documents and outputs the best documents or passages annotated with the named entities. The final stage is to select the particular answer, given the answer class and the top scoring passages.

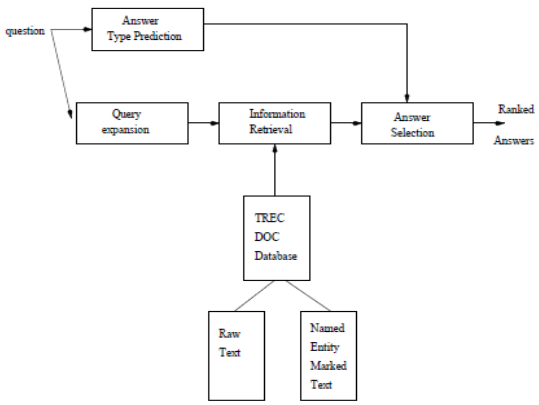

FIGURE 2. QUESTION ANSWERING ARCHITECTURE

A. *QA Type Classification*

The classification refers to the classification of questions and answers. In classifying the type of answer the problem is to label a question with the label of the named entity that the question seeks. These labels are the standard MUC (Chinchor, 1997) categories with the addition of PHRASE (P), which presents all the answers that cannot be put in standard categories. In standard categories we have REASON (R) category, which is tied to *why* questions. Processing of REASON and PHRASE is the same in this IBM system, interpreting it as desiring a clause which has a NOUN PHRASE (NP) imbedded in it.

Ittycheriah, Franz, Zhu, Ratnaparkhi, and Mammone created 1900 questions by presenting a human subject a randomly selected document and having read a portion of the document , a question was phrased, the answer and a document number noted in addition. They also used 1400 questions from a trivia database (Hallmarks, 1999), annotated in a similar way.

Each feature type expands to the property above it. The "Expanded Hierarchy" feature uses WordNet (Miller, 1990) in order to expand words from a question words up to those including noun predicate. The "Mark Question Word" feature identifies the question words and labels them as occurring in the beginning of a question, i.e. bqw, in the middle of a question, i.e. mdw or at end of a question, i.e. eqw.

N- gram language models are used as the method for language modeling. Regarding the length, N-grams can be unigrams (n =1), bigrams (n = 2), trigrams (n = 3)... (Gaspic, 2012).

Table 1 shows the classification of the QA type on the example question *"What year did World War II start?"* First, each word is classified into one of the word classes. WP is *wh* - pronoun such as *who, what, which, how, how many* ... NP refers to a noun phrase, VP verb phrase, NN singular noun, after English noun number, VBD verb in the past tense.

Table 1 shows that *what* is marked as an interrogative pronoun, while *year* is annotated as a noun. Expanding the features of the question pronouns up to those that contain the nominal predicate , i.e. we link *what* to *year*, we can conclude that the *wh* pronoun *what* and a noun predicate *year* require a certain time or a certain time period. Subsequently, we determine

TABLE I. FEATURES USED IN THE ANSWER CLASSIFICATION EXPERIMENTS

Unigrams	What year did World War II start?
Morphed,Part-Of-Speech	what{WP} year{NN} do{VBD} World{NP} War{NP} II{NP} start{NN}
Bigrams	what{wp} what{wp}_year{nn} what{wp}_do{vbd} what{wp}_world{np} ...
Expanded Hierarchy	what{WP} year time_period measure abstraction year{NN} do{VBD} ...
Mark Question Word	what_bqw year time_period measure abstraction year{NN} do{VBD} ...

TABLE II. RETRIEVAL RESULT

	MRR
pass1, TREC	0.4605
pass2, TREC	0.4824
pass2, encyclopedia	0.5031

the position of a question word within a question; in this case it is the beginning of the question, as shown in Table 1.

B. Information Retrieval

The purpose of the IR module is to search the database to select passages of the text, containing information relevant to the query. The IBM IR system uses a two-pass approach. In the first pass, the encyclopedia database is searched. The highest scoring passages are then used to create expanded queries, applied in the second pass. The scoring is based on unigram and bigram features extracted from the text data using tokenization, part-of-speech tagging and a morphological analyzer. (Merialdo, 1990).

In the first pass, Ittycheriah et al. modified the Okapi formula [1] (Robertson, Walker, Jones, Hancock - Beaulieu, Gatford, 1995) in order to score passages extracted from the encyclopedia documents, then converted all the encyclopedia articles into 82 277 overlapping passages, each containing about 100 non-stop words. Based on the first passage ranking, they constructed expanded queries. In the second passage they used extended queries and scored 2 632 807 passages based on the TREC-9 corpus. The passages contained about 200 non-stop words. Table 2 summarizes the Information Retrieval (IR) results on tests (Sparck Jones and Willett, 1997).

The performance is measured by the MRR[2] or Mean Reciprocal Rank. The first line of the Table 2 shows the result of the first pass scoring; the second line shows the result of the second pass scoring, and the third line shows the result corresponding to the system applied by the IBM as an example, with queries expanded using the encyclopedia database.

C. Named Entity Marking

Named entity annotation refers to a markup of the text with the class information. As mentioned above, classes correspond to the standard MUC classes.

Windows of + or - words, morphs, part-of-speech tags and flags raised by pattern grammars (DATE, MONEY, CARD, MEASURE, PERCENT, TIME, DURATION). The window for predicting the tag (0) is shown in Table 3.

Each stream has a fixed vocabulary. N-grams from this vocabulary form the features of the maximum entropy model.

[1] In IR is a ranking function used by search engines to rank matching documents according to their relevance to a given search query.

[2] MRR(Q) = $\frac{1}{|Q|} \cdot \sum_{i=1}^{|Q|} \frac{1}{r(d_{i,rel})}$

TABLE III. FEATURES USED IN THE NAMED ENTITY MODEL FOR PREDICTING TAG(0)

Words	w(-) w(-) | w(0) | w(+) w(+)
Morphs	m(-) m(-) | m(0) | m(+) m(+)
Part-of-Speech	p(-2) p(-1) | p(0) | p(1) p(2)
Grammar Flags	f(-2) f(-1) | f(0) | f(1) f(2)
Previous Tags	t(-2).t(-1) t(-1) |

The training data is arranged to indicate a special category for beginning of each named entity, for example BeginPERSON to find the boundaries of the named entity.

The system explores multiple NE hypotheses in parallel and keeps only those only those with high probability. It proceeds with the algorithm to find the most likely part for the whole sentence.

D. Answer Selection

In this phase we have a question, a class of the answer that the question seeks and a ranked set of passages annotated with the MUC classes. We are searching now the optimal sentence that will contain the answer. The TREC length constraints of 250 byte and 50 byte are then applied on the sentence. The algorithm used in this module is listed here:

1) Each passage is split into sentences.
2) A window is formed around each sentence.
3) The following features are computed:
   a) *Matching Words* (the sum of the number of words that matched identically in the morphed space (+)),
   b) *Aligning the Terms* (the sum of the number of words that are synonyms (+))
   c) *Mis-Match Words* (the sum of the number of question content words that did not match in this answer (-))
   d) *Dispersion* (the number of words in the candidate sentence that occurred between matching question words (-))
   e) *Cluster Words* (the number of words in the candidate sentence that occurred adjacently in both the question and answer candidate (+))
4) The location or absence of desired entities is noted in the score.
5) Each of these distances are weighted, the sentences ranked and the top 5 sentences are then output.

Each of the above distances has its own weight and the corresponding sign shown above to attach to it. The score for an answer is the sum of distances and the top 5 sentences are then output.

In order to select the 250 or 50 byte answer from these sentences, the system identifies the longest mismatched pieces between the answer and the question. It then analyses the answer and the question to find where the center of the match is and using a

subject-verb-object assumptions of the sentence, it takes the question as either desiring the subject or the object, whichever has the least matches with the question.

E. Testing

Testing this IBM example, the authors came to the conclusion that the selection of the answer is still the hugest problem in QA systems. In 24.7 % of cases, their 250B system showed an error when selecting answers, while at 50B system the percentage reaches even 35.6 %.

The method used in the IBM QA system is a complex search that is not trying to get to the semantics of the text, but introduces a number of measures to calculate the overlapping of the content and the answer.

2. DeepQA Project

IBM Research undertook a challenge to build a computer system that could compete at the human champion level in real time on the American TV quiz show, Jeopardy. The Jeopardy Challenge helped to address requirements that led to the design of the DeepQA architecture (Figure 2.) and the implementation of Watson. The goals of IBM Research are to advance computer science by exploring new ways for computer technology to affect science, business, and society. The Jeopardy Challenge requires advancing and incorporating a variety of QA technologies including parsing, question classification, question decomposition, automatic source acquisition and evaluation, entity and relation detection, logical form generation, and knowledge representation and reasoning. Baseline performance is the QA system called Practical Intelligent Question Answering Technology (PIQUANT) (Prager, Chu-Carroll, and Czuba 2004), which had been under development at IBM Research by a four-person team for 6 years prior to taking on theJeopardy Challenge. At the time it was among the top three to five Text Retrieval Conference (TREC) QA systems. PIQUANT was a classic QA pipeline with state-of-the-art techniques aimed largely at the TREC QA evaluation (Voorhees and Dang 2005). PIQUANT performed in the 33 percent accuracy range in TREC evaluations. While the TREC QA evaluation allowed the use of the web, PIQUANT focused on question answering using local resources. A similar baseline experiment was performed in collaboration with Carnegie Mellon University (CMU) using OpenEphyra, an open-source QA framework developed primarily at CMU. The framework is based on the Ephyra system, which was designed for answering TREC questions. In these experiments on TREC 2002 data, OpenEphyra answered 45 percent of the questions correctly using a live web search. Minimal effort was spent in adapting OpenEphyra, but like PIQUANT, its performance on *Jeopardy* clues was below 15 percent accuracy.

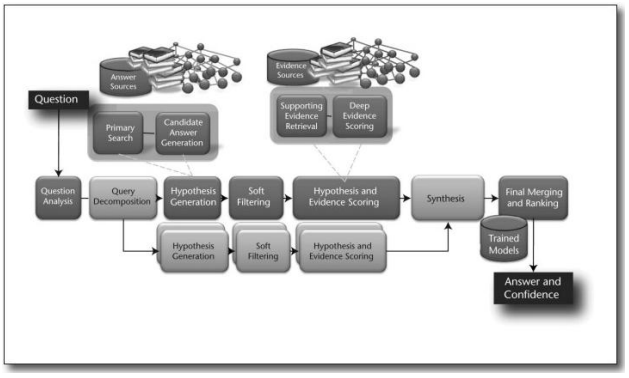

FIGURE 2. DEEPQA ARCHITECTURE

A system called DeepQA, still continuing to develop, is a massively parallel probabilistic evidence-based architecture. DeepQA is an architecture with an accompanying methodology. DeepQA has successfully been applied to both the *Jeopardy* and TREC QA task. It was first adapted to different business applications and additionally to exploratory challenge problems including medicine, enterprise search, and gaming.

The overarching principles in DeepQA are massive parallelism, many experts, pervasive confidence estimation, and integration of shallow and deep knowledge.

Massive parallelism: Exploit massive parallelism in the consideration of multiple interpretations and hypotheses.

Many experts: Facilitate the integration, application, and contextual evaluation of a wide range of loosely coupled probabilistic question and content analytics.

Pervasive confidence estimation: No component commits to an answer; all components produce features and associated confidences, scoring different question and content interpretations. An underlying confidence-processing substrate learns how to stack and combine the scores.

Integrate shallow and deep knowledge: Balance the use of strict semantics and shallow semantics, leveraging many loosely formed ontologies (Ferrucci at al., 2010).

The authors believe that in the future more methods will focus on detecting semantics and propose pretreatment of the texts and their recording in the network of knowledge to be used for asking questions and searching answers. It is planned to do such a research by using the NOK method.

IV. CONCLUSION

The complexity of technical evaluation of QA systems indicates resources and evaluation as two critical issues in their development.

Standardized tests use multiple-choice questions.

The selection of the evaluation criteria presents the problems in assessing the answer. The evaluation criteria are: relevance, correctness, conciseness, completeness, coherence and justification.

Today's QA evaluation of TREC requires the answer to be justified by the context as well as the limitation of the answer (in bytes), which is the first step towards brevity.

Evaluation criteria must depend on the purpose of use, the type of user who searches for an answer and on the interface.

Regarding the short answers, it is possible to automate the comparison of answers. Comparisons that are not accurate, as those by human evaluator, nevertheless provide good results (93-95 %).

Tests where the person himself / herself writes the answers (either short answers or an essay) are considered to be too subjective for standardized testing.

An automated method of evaluation, sometimes in combination with only one evaluator has proven to be worthwhile, which indicates that the automated grading systems are approaching human evaluators.

The IBM, which has played a leading role in the development of information technology, has developed QA systems whose results show an error in 30 % of cases.

They are constantly seeking an algorithm (method, model, procedure) to solve the selection of a short and concise answer.

QA systems are still unreliable, do not always provide good results, require a lot of sources, can't make conclusions, do not solve the context problem, etc.

Further research will go towards the development of QA systems using the NOK.

ACKNOWLEDGMENT

The research has been conducted under the project "Extending the information system development methodology with artificial intelligence methods" (reference number 13.13.1.2.01.) supported by University of Rijeka (Croatia).

REFERENCES

1. Academic Hallmarks (1999). Knowledge master. http://greatauk.com/ (10.02.2014.)
2. Breck, E., Burger, J. D., Ferro, L., Hirschman, L., House, D., Light, M., & Mani, I. (2000). How to evaluate your question answering system every day and still get real work done. *arXiv preprint cs/0004008*.
3. Chinchor, N., & Robinson, P. (1997, September). MUC-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*.
4. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., ... & Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI magazine*, *31*(3), 59-79.
5. Gašpić, K., Statističko strojno prevođenje, (2012) http://www.scribd.com/doc/105502391/Statisti%C4%8Dko-strojno-prevo%C4%91enje (10.02.2014.)
6. Hirschman, L., Breck, E., Light, M., Burger, J. D., & Ferro, L. (2000). Automated grading of short-answer tests.
7. Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering*, *7*(4), 275-300.
8. Ittycheriah, A., Franz, M., Zhu, W. J., Ratnaparkhi, A., & Mammone, R. J. (2000, November). IBM's Statistical Question Answering System. In *TREC*.
9. Ittycheriah, A., Franz, M., & Roukos, S. (2001). IBM's Statistical Question Answering System-TREC-10. In *TREC*.
10. Jakupović, A., Pavlić, M., Dovedan, Z. Han, „Formalisation Method for the Text Expressed Knowledge" neobjavljeno, 2013.
11. Jones, K. S. (Ed.). (1997). *Readings in information retrieval*. Morgan Kaufmann.
12. Kukich, K. (2000). Beyond automated essay scoring. *IEEE intelligent systems*, *15*(5), 22-27.
13. Kupiec, J. (1993, July). MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 181-190). ACM.
14. Mann, G. S. (2001, July). A statistical method for short answer extraction. In *Proceedings of the workshop on Open-domain question answering-Volume 12* (pp. 1-8). Association for Computational Linguistics.
15. Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational linguistics*, *20*(2), 155-171.
16. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, *3*(4), 235-244.
17. Pavlić, M., Development of a method for knowledge modeling 2013., Rijeka: Odjel za informatiku Sveučilišta u Rijeci, 2013a
18. Pavlić, M., Jakupović, A., Meštrović, A.: "Nodes of knowledge method for knowledge representation", Informatol. 46, 2013b, 3, 206-214
19. Pavlić, M., Meštrović, A., Jakupović, A.,"Graph-Based Formalisms for Knowledge Representation", Proceedings of the 17th World Multi-Conference on Systemics Cybernetics and Informatics (WMSCI 2013), Vol 2. 2013c 200-204.
20. Prager, J. M.; Chu-Carroll, J.; and Czuba, K. 2004. A Multi-Strategy, Multi-Question Approach to Question Answering. In *NewDirections in Question-Answering*, ed. M. Maybury. Menlo Park, CA: AAAI Press.
21. Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *NIST SPECIAL PUBLICATION SP*, 109-109.
22. Roever, C. (2001). Web-based language testing. *Language Learning & Technology*, *5*(2), 84-94.
23. Tomljanović, J., Krsnik, M., Pavlić, M. (2014). Inteligentni sustavi pitanja i odgovora. Zbornik Veleučilišta u Rijeci - Journal of the Polytechnic of Rijeka
24. Ushida, E. (2005). The role of students' attitudes and motivation in second language learning in online language courses. *CALICO journal*, *23*(1), 49-78.
25. Voorhees, E. M., and Dang, H. T. 2005. Overview of the TREC 2005 Question Answering Track. In *Proceedings of the Fourteenth Text Retrieval Conference*. Gaithersburg, MD: National Institute of Standards and Technology.
26. http://www.eureka-centar.hr/upisi/on-line-testiranje/engleski-jezik/ (11.02.2014.)
27. http://www.pnas.org/content/100/15/9096.long (11.02.2014.)
28. http://www.surrey.ac.uk/ELI/ltr.html (11.02.2014.)