

DEVELOPEMNT AND EVALUATION OF A WEB BASED QUESTION ANSWERING SYSTEM FOR ARABIC LANGUAG

Heba Kurdi, Sara Alkhaider, Nada Alfaifi

Department of Computer Science
Al Imam Muhammad Ibn Saud Islamic University
Riyadh, SA

ABSTRACT

Question Answering (QA) systems are gaining great importance due to the increasing amount of web content and the high demand for digital information that regular information retrieval techniques cannot satisfy. A question answering system enables users to have a natural language dialog with the machine, which is required for virtually all emerging online service systems on the Internet. The need for such systems is higher in the context of the Arabic language. This is because of the scarcity of Arabic QA systems, which can be attributed to the great challenges they present to the research community, including the particularities of Arabic, such as short vowels, absence of capital letters, complex morphology, etc. In this paper, we report the design and implementation of an Arabic web-based question answering system, which we called "JAWEB", the Arabic word for the verb "answer". Unlike all Arabic question-answering systems, JAWEB is a web-based application, so it can be accessed at any time and from anywhere. Evaluating JAWEB showed that it gives the correct answer with 100% recall and 80% precision on average. When compared to ask.com, the well-established web-based QA system, JAWEB provided 15-20% higher recall. These promising results give clear evidence that JAWEB has great potential as a QA platform and is much needed by Arabic-speaking Internet users across the world.

KEYWORDS

Question Answering system, Natural Language Processing , Arabic language tools.

1. INTRODUCTION

Question-answering is a challenging task in general. It involves state-of-the-art techniques in various fields, such as Information Retrieval (IR), Natural Language Processing (NLP), Artificial Intelligence (AI) and software technologies. The main goal of a question-answering system, is to give a precise answer to users' queries in natural language [1] [2]. This has the great advantage of helping users to get desired answers without searching large pools of information. Unlike search engines, such as Google and Yahoo, which allow a user to retrieve webpages or documents that are partially relevant to given keyword(s), while leaving the job of excluding irrelevant hyperlinks and locating desired passages to users, QA systems provide the user with highly relevant information in answer to their questions at the passage or sentence level [3]. QA systems categorize questions into three main groups: first, fact-seeking questions, i.e. factoid questions, asking about (who, when, where, what and how much) which relate to different types of entities: person, location, organization, time and quantity. This type of question is the most common form of questions and therefore is the focus of this work. Second is list questions (e.g. List all the countries of the European Union), asking about multiple pieces of information with a common relation between them. Third is why-type questions (e.g. Why is the sky blue?) which seeks explanations of causes for a certain phenomenon or event. The second and third categories are

less common in QA systems and require more advanced NLP and AI techniques and were therefore excluded from this study.

Based on the knowledge domain, QA systems classify questions as either closed-domain questions, which ask about information related to a specific domain (for example medical or sport, etc.), or open-domain questions which ask about information in any knowledge discipline. They require wider knowledge and advanced techniques to extract the most relevant answers [4]. The latter is the focus of this paper.

A QA system consists of three main components: a question-classification component, an information-retrieval component, and an answer-extraction component. The question-classification component plays a primary role as it categorizes questions based on their keywords: who, when, where, what and how much. The information-retrieval component searches for relevant answers in information sources by looking for matching keywords. Finally, the answer-extraction component selects the most relevant answers and ranks them accordingly. The quality of a QA system heavily depends on the effectiveness of this module as it makes the decision about the final list of answers to be presented to the user [5] [6].

Although the systems have common core components, they differ vastly in the techniques utilized by each component. This can be attributed mainly to the supported language(s) by the system, since each language has special features that affect how it should be analyzed, searched and retrieved.

Arabic is among the most popular languages with nearly 300 million speakers across the globe. It is a Semitic language. These are well known for their non-concatenative morphology, in which roots consist of an isolated set of constants, rather than syllables or words. This feature among many others has rated NLP and IR as grand challenging tasks for the Arabic language, despite some attempts such as [7]-[15]. Consequently, the huge Arabic content on the Internet is still underutilized [16]. Important emerging applications, question-answering systems for online services in particular, are simply not available in Arabic, despite their abundance in many other languages, especially English and other Latin-based languages. The challenging features of the Arabic language, scarcity of Arabic QA systems and lack of web-based Arabic QA systems are the main drivers for this paper. It adds several contributions to this important field, including comprehensively surveying Arabic QA systems and a comparing between them, developing and evaluating of JAWEB, the first Arabic web-based QA system, and extending the Arabic Corpus provided in [30].

The remaining of this paper is organized as follows: In section 2, the challenges of the Arabic language are highlighted. Section 3 reviews related work in QA systems. The system architecture and functional components of JAWEB are introduced in section 4. In section 5, the experimental results are presented and discussed. Finally, section 6 concludes the paper and indicates future work.

2. CHALLENGES OF ARABIC LANGUAGE

Arabic is highly inflectional and derivational. This results in sparseness of terms in text, which leads to inefficiency in many statistical IR and NLP techniques. The absence of diacritics in Modern Standard Arabic also adds a lot of ambiguity to Question Analysis and Answer Extraction [16].

Many aspects are involved in the slow progress of Arabic NLP and IR, compared to the accomplishments in English and other languages. These aspects, as highlighted in [17], include:

1. Arabic has a complex morphology. It is highly derivational and inflectional, which extremely complicates morphology analysis.

- Arabic is a derivational language, to find a word in an Arabic dictionary, we start by extracting the root and then we search for the root in the dictionary. The word derivation is done by adding affixes (prefix, infix, or suffix) to the root, according to several patterns [18] [19], Figure 1 shows an example of that.

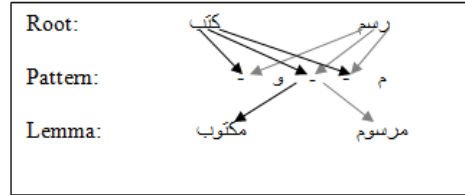


Figure 1 An example of Arabic Language derivation

- Arabic is an inflectional language, which means that the construction of a word involves finding the root and adding affixes (prefix, infix and suffix) to it, as illustrated in Figure 2.

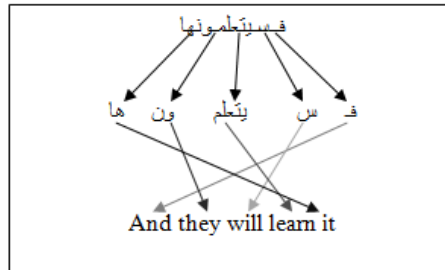


Figure 2an example of Arabic words composition

- The correct meaning of an Arabic word can be ambiguous in the absence of diacritical marks (short vowels);so a written word may hold different meanings as illustrated in Figure 3.
- Arabic does not use capital letters which makes differentiating between named entities and other words difficult, as shown in Figure 4.

Disconnected	فصل
Is disconnected	فصل
So connect	فصل
Disconnection, chapter, semester	فصل
Detail (verb)	فصل
Disconnected parts	فصل

Figure 3: An example of the effect of Arabic diacritics in the meaning of a word



Figure 4. An example of absence of capital letters in Arabic

- Numerals are written form left to right, while alphabets are written from right to left. This makes editing Arabic text difficult when both numbers and letters are presented on the same line.
- The most used encodings for Arabic text, UTF-8 and Unicode, present many problems when processing Arabic texts.
- Lack of Arabic corpora, lexicon and electronic dictionaries.

3.RELATED WORK

Question-Answering systems present a good solution for textual information retrieval and knowledge sharing and discovery. This is why a large number of QA systems have been developed in various languages. Some languages, such as English, are better served than others, such as Arabic. This might be related to language features and the maturity of research in the countries speaking it. This section surveys QA systems for English and for Arabic and compares between them.

A. For Latin languages

Due to the popularity, importance and features of the English language, tens of QA systems are available in English. Extensive surveys are available in [3] [20]. Here we highlight some of the most prominent work on this area.

- QALC [21] is the first QA system developed for English in the Text REtrieval Conference (TREC) evolution campaign in 1999. It provides answers to English factoid questions based on syntactic and semantic analysis.
- QRISTAL [22] is a multilingual QA system (English, French, Portuguese, Italian and Polish). It retrieves answers from a local database or from the web, based on named entities' recognition and conceptual and thematic analysis.
- WebQA [23] is a web based QA system, which uses the template-mapping technique to define the question type. It retrieves passages from Google search engine and employs a clustering technique to extract multiple answer blocks from different web pages.
- Ask.com [24], originally known as Ask Jeeves, is a famous web-based question-answering system that supports English, Arabic and many other languages. It allows users to get answers to questions expressed in everyday, natural language, as well as by traditional keyword searching. It also supports math, dictionary, and conversion questions.

B. For Arabic languages

The situation is less bright for the Arabic language, despite of its wide spread. Although research in the field of Arabic QA systems has already started [7]-[15], it is slow progressing and has limited results. Computerized tools and resources in general are lacking in Arabic [16], which has reflected negatively on the number of Arabic QA systems. The list includes:

- AQSA (1993) [25] seems to be the first QA system in the language. It is a knowledge-based QA system that extracts answers from structured data only. It uses the frames technique to present the knowledge from the radiation domain. However, no published evaluation is available for the system.
- QARAB (2004) [26] is a stand-alone (non-web-based) QA system that uses IR and NLP techniques to extract answers from a collection of Arabic newspaper texts. It provides answers to factoid questions but does not support other types of questions, such as how or why.
- ArabiQA (2007) [27] is a stand-alone Arabic QA system that deals with factoid questions, using Named Entity Recognition techniques and Java Information Retrieval System (JIRS) for Arabic text. It is designed specifically for Arabic factoid questions. However, system implementation has not been completed yet.
- QASAL (2009) [28] is a stand-alone Arabic QA system for factoid questions which uses the NooJ platform [29] as a linguistic development environment. The system takes advantage of some linguistic techniques from IR and NLP to process Arabic text documents containing answers to factoid questions. Published work based on this system does not include experimental results or performance metrics. The overall functionality of the system is limited to the amount of available tools developed for the Arabic language by the NooJ Arabic

modules.

- ArQA (2011) [17] is a stand-alone QA system that provides answers to factoid questions expressed in Arabic. This system has a pipeline architecture with four modules: question processing, passage retrieval, answer extraction and answer validation modules. Each module is the result of combining several IR and NLP techniques and tools to improve validity of returned answers.
- AQuASys (2011) [30]: a stand-alone Arabic QAsystem for factoid questions. It extensivelyutilizes NLP techniques to analyses questions and retrieves answers from an Arabic corpus that has been developed by the authors. Retrieved answers are scored and presented based on their relevance.

Table1 illustrates the differences between some state-of-art QA systems in English and Arabic languages. Based on the table it is clear that web-based Arabic QA systems are lacking.

Table 1: comparison between QA Systems

Main Features	QALC	QRISTAL	Ask.com	QARAB	ArabiQA	ArQA	QASAL	AQuASys	JAWEB
Web-based system	×	×	√	×	×	×	×	×	√
Retrieves answers from a corpus	√	×	×	√	√	√	√	√	√
Retrieves answers from the web	×	√	√	×	×	×	×	×	×
Natural language processing tools	√	√	√	√	×	√	√	√	√
Named entity recognition	×	√	√	×	√	√	√	×	×
Answers factoid questions	√	×	√	×	√	√	√	√	√
Answers Open domain questions	√	√	√	×	√	×	√	√	√
Supports multiple languages	×	√	√	×	×	×	×	×	×
Supports Arabic language	×	×	√	√	√	√	√	√	√
Provides Short answers block	√	√	×	√	√	√	√	√	√
Measures answer precision	-	√	×	-	√	√	-	√	√
Measures answer recall	-	√	√	-	√	√	√	√	√

3.JAWEB ARCHITECTURE

JAWEBis an Arabic web-based QA system that focuses on factoid questions. It accepts questions related to any named entity,including person, location, organization, time etc.. Afteranalyzing the question, important information is extracted to retrieve the most relevant answers from an Arabic corpus. The system is composed of fourcomponents: user interface, question analyzer, passage retrieval and answer extractor. The user interface runs at the client side while all other modules run at the server side.As illustrated in Figure 5, each component is structured of several modules with a distinct task carried by each.

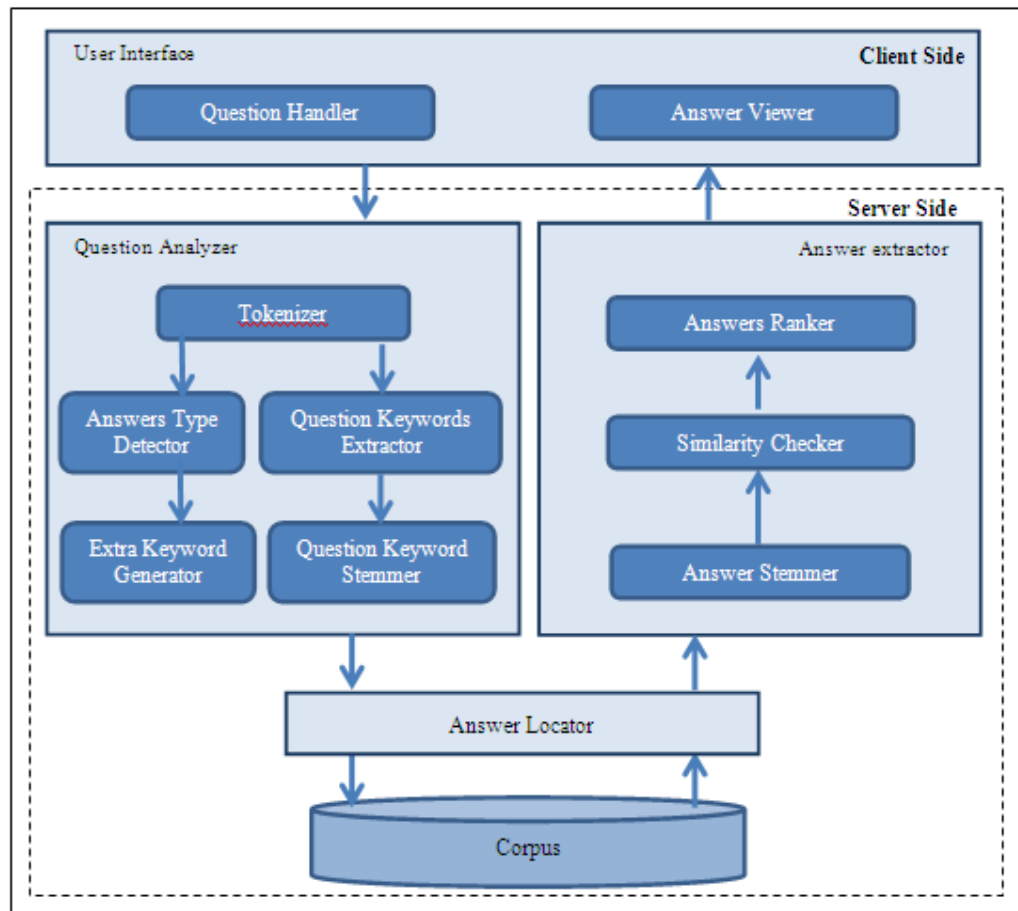


Figure 5: JAWEB system detailed architecture

- User Interface

The user interface is implemented as a simple webpage with an input form that accepts Arabic factoid questions expressed in natural language. The question handler module validates entered questions and passes them to the server side. Later on, when the answers are retrieved from the corpus, the answer-viewer module formats and presents them in descending order of relevance; the desired answer is displayed at the top, followed by a list of candidate answers.

- Question-Analyzer:

This component aims at identifying the type of the question. It has five modules: tokenizer, answer-type detector, question keyword extractor, extra keywords generator and question words stemmer. The user enters a question in a web-based form at the client side. The question is sent to the server side, where the tokenizer breaks the question into separate words. Afterwards, the answer-type detector identifies the category of the answer, which can be time, organization, person, location etc., based on the interrogative nouns (من who, ما what, أين where and متى when), as illustrated in Table 2. The question keywords are the rest of the words in the question, after excluding stop-words and interrogative nouns; they are identified by the question keywords extractor module. The extra keywords generator produces synonyms of the keywords to expand the range of related answers. For example, for a question like: متى استقلت تونس: (When has Tunisia become independent?) the extra keywords are (عام, year, سنة, month etc.).

keywords are stemmed, if necessary, so affixes are removed by the question keywords stemmer module to ease computing similarity with the keywords in the extracted answers later. The high-level algorithm of the Question Analyzer is illustrated in Alg. 1.

Table2: Example of question interrogative nouns and expected answer type

Expected Answer type	interrogative nouns
Time	متى when
Location	أين where
Object	ما what
Person	من who
Quantity	كم how much
Quantity	كم how many

Alg. 1 Question Analyzer

Input: user question in natural language

Output: type of expected answers, stemmed question keywords and extra keywords

```

tokenVector = tokenizer (userQuestion)
interrogativeNoun = tokenVector.interrogativeNoun ();
If (interrogativeNoun == "أين")           // where
    Then{ answerExpectedType = "Location"
If (interrogativeNoun == "من")           // who
    Then{ answerExpectedType = "Person"
If (interrogativeNoun == "متى")          // when
    Then{ answerExpectedType = "Time"
If (interrogativeNoun == "ما")           // when
    Then{ answerExpectedType = "Thing"
If (interrogativeNoun == "كم")           // how much/many
    Then{ answerExpectedType = "Quantity"
questionKeywords = tokenVector.questionKeywords ();
stemmedQuestionKeywords = stemmer (questionKeywords)
extraKeywords = extraQuestionKeyword.Generator (questionKeywords) }
```

- AnswersRetriever:

This component searches for potential answers and retrieves them from the corpus. It locates all sentences that contains a pattern that matches any word from the list of question keywords and extra keywords, as shown in Alg. 2.

Alg. 2 Potential Answer Retrieval

Input: question type, list of stemmed question keywords and extra keywords

Output: list of potential answers

n = 0 // no. of potential answers

While not endOfCorpus

```

    potentialAnswer[n]= locateMatchingSentence (questionKeywords, extraKeywords)
    n++
```

- Answers Extractor

This component is responsible for selecting the most relevant answers from the potential answers list. It is composed of three modules. The first module is the answer keywords stemmer which gets roots of the keywords in the retrieved answer to ease checking similarity between a user question and potential answers, a task carried out by the second module, which is the answer-similarity checker. Similarity is measured by counting the number of matching keywords between the question and each retrieved answer. Based on this, the third module, answers ranker, sorts answers in descending order of their relevance. The most relevant answer is considered as the desired answer while the rest of answers are candidate answers. The main steps carried by this module are illustrated in Alg. 3.

```

Alg. 3 AnswerExtractor (potentialAnswer [],stemmedQuestionKeywords)
Input: list of potential answers,stemmed question keywords
Output: list of extracted answers ranked in descending order of similarity
n = 0    // no. of potential answers
Repeat
    stemmedAnswerKeywords = stemmer (potentialAnswer [n])
    similarity = noOfSimilarWords (stemmedQuestionKeywords,
                                stemmedAnswerkeywords)
    potentialAnswer [n].similarity =similarity
sortDescending (potentialAnswer [])
    
```

4.IMPLEMENTATION AND TESTING

JAWEB has been built based on AQuASys [30] and as a natural extension to it. It provides a web interface to the system, an additional support for Arabic language presentation in web browsers, an extended corpus as well as an extensive evaluation and testing framework. The user interface was developed as a JSP webpage that accepts Arabic factoid questions. It provides a user-friendly interface where a user can type his/her questions. The question should begin with an interrogative noun:

Dreamweaver, an Adobe proprietary web development application, was used to design the interface. The servlet and java classes on the server side were implemented using NetBeans IDE. The server was a GlassFish Server 3.1.2, which is an open-source application server project started by Sun Microsystems for the Java EE platform and now sponsored by Oracle Corporation. This server ran on 2.50 GHz CPU and an internal memory of 4 GB.

An extended version of the Arabic corpus developed by [30], containing 39,660 words with size of 457 KB, was used as the information pool to retrieve answers. The Arabic Khoja's stemmer [31] was adopted for keywords stemming.

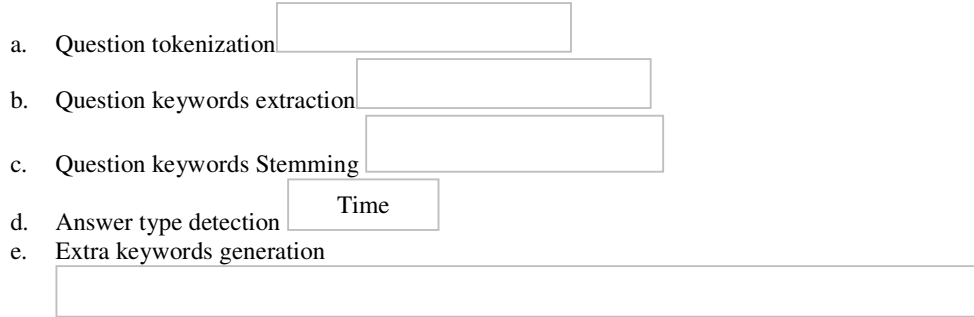
The system was tested thoroughly to insure correctness of obtained results. The following example illustrates the main steps carried out by the system to answer the Arabic factoid question: "when did Ibn Khaldun die?"

1) In JAWEB webpage, illustrated in Figure 6, the user typed the question in the textbox:



Figure 6: Jaweb system input interface

2) The user clicked the " " button, sent the question to the sever side, where the question analyzer received it. The following processes were done:



3) The system located all answers that contain any keyword or extra keyword from the corpus, as shown in Figure 7.



Figure 7: Retrieved answer based on extra keyword

4) In each retrieved answer,

- Keywords were stemmed.
- Similarity was calculated by counting how many keywords are contained in each answer.
- Answers were ranked based on similarity.

5) The system displayed the answers in order, based on similarity. The answer with highest rank was placed at the top, as shown in Figure 8.



Figure 8: View the answers based on the rank

5. PERFORMANCE EVALUATION

6.1 Performance Measures

The end objective of JAWEB is to develop a web-based QA system that answers Arabic factoid questions with high accuracy and speed. To evaluate this system, three performance measures were considered:

- **RECALL** [32]: the ratio of the number of relevant sentences retrieved to the total number of relevant sentences in the corpus. It is usually expressed as a percentage and calculated as:

$$RECALL = \frac{A}{A+B} * 100\% \quad (1)$$

Where

A: number of retrieved relevant answers, B: number of relevant answer not retrieved

- **PRECISION** [32]: the ratio of the number of relevant sentences retrieved to the total number of sentences retrieved. It is usually expressed as a percentage and calculated as:

$$PRECISION = \frac{A}{A+C} * 100\% \quad (2)$$

Where

A: number of retrieved relevant answers, C: number of retrieved irrelevant answers

- **RESPONSE TIME**: the time from when a user hits the search button until a response is displayed on the browser.

6.2 System Performance

We ran tens of experiments with various types of factoid questions. The results of recall, precision and response time are presented in Figure 9, Figure 10 and Figure 11.

As illustrated in Figure 9, The **RECALL** of JAWEB was always an incredible 100%. This is because it was always able to return all relevant answers in the corpus successfully. This means that B, the number of relevant answer not retrieved, was maintained at zero. It is also important to

note that the first retrieved answers of all questions were correct and accurate, which can be attributed to the use of the stemmer (in both question and answer modules), the extra keywords generator and the similarity checker which has resulted in accurate ranking of retrieved answers.

Figure 10 shows that the average *PRECISION* was 80%, a remarkable number, considering the limited size of our corpus. Even the “how much” questions, which had the least precision, had a percentage no lower than 70%. “What” and “how many”, the most straightforward questions, received the most precise answers. It should be noted these results are also highly dependent on the amount of relevant and irrelevant information available in the corpus that relates to each question.

In Figure 11, the average *RESPONSE TIME* is calculated in nanoseconds and presented for each type of questions. The average response time was 108.2 nanoseconds. “How much” questions consumed the longest response time, while questions starting with “who” took the least. This makes sense, as the latter requires less processing, since searching for people’s names is usually straight-forward. However, the *RESPONSE TIME* in general depends on the corpus size and the power of the server machine.

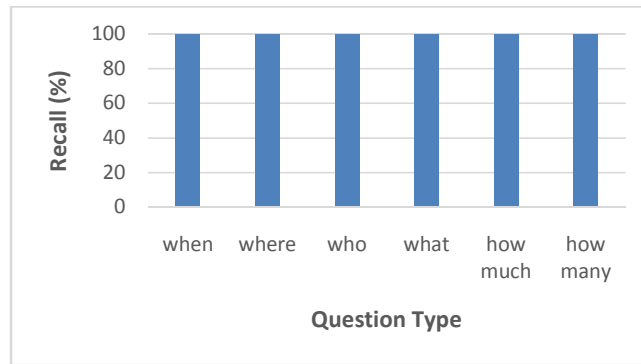


Figure 9: Recall of different types of questions

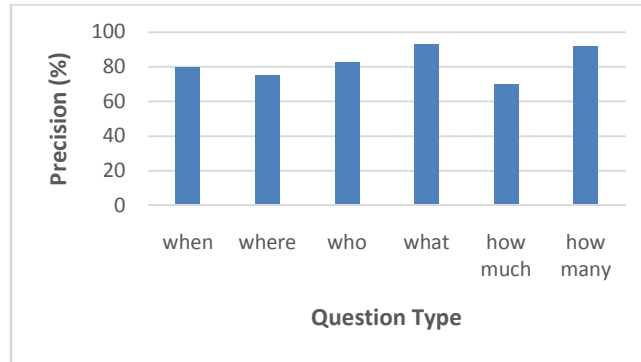


Figure 10: Precision of different types of questions

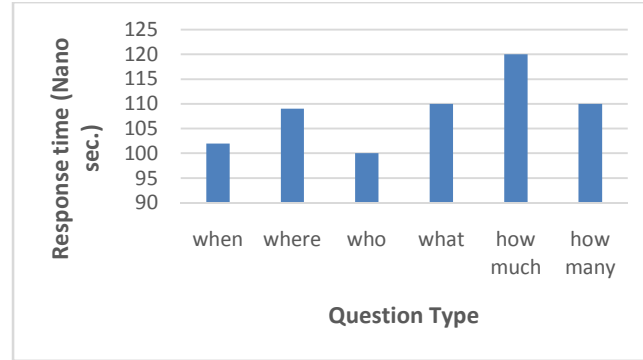


Figure 11: Response time of different types of questions

6.3 Benchmarking

To have clear insight into JAWEB performance, we used a well-established QA system that supports the Arabic Language, Ask.com [10], as a benchmark. Ask.com is a famous web-based question answering system with multilingual support. It accepts colloquially-expressed questions and retrieves hyperlinks to webpages that contain similar keywords to those in the questions.

We ran several experiments with each of the supported type of factoid questions to get answers from Ask.com and JAWEB. Snapshots of answers to some example questions, listed in Table 3, are presented in Figure 12 –Figure 17. All figures illustrate that JAWEB has consistently given the right and most accurate answer as the first responses which can be attributed to the use of the stemmer (in both question and answer modules), the extra keywords generator and the similarity checker which has resulted in accurate ranking of retrieved answers. In contrast, Ask.com provided the correct answer in the second or third hyperlinks.

Comparisons between Ask.com and JAWEB in terms of *RECALL*, *PRECISION* and *RESPONSE TIME* are depicted in the graphs in Figure 18, Figure 19 and Figure 20, respectively. It is important to note that having no access to the knowledge base of Ask.com and in order to be able to calculate the *RECALL* and *PRECISION*, we have only considered the first five retrieved pages, which may slightly affect the accuracy of the experiments.

Figure 18 shows that *RECALL* of JAWEB was always higher than that of Ask.com and maintained at 100%, as the former has successfully retrieved all relevant answers from the corpus. The *JAWEB PRECISION* was admittedly less than ask.com, as shown in Figure 19, but it should be noted that it performed as well as the famous website, scoring over 90% in precision, despite the large difference in corpus sizes. In addition, even at its worst, for a question type that ask.com had the lowest precision too, JAWEB was not less precise than 70%. Regardless, the precision can easily be improved as the project launches, when feedback from users is attained and the corpus is grown in size and capacity. We believe that this is clear evidence that JAWEB has great potential as a QA platform and is much needed by Arabic-speaking Internet users across the world.

Naturally, ask.com was faster than our system, as illustrated in Figure 20, thanks to the use of high-power server CPUs for ask.com, as opposed to JAWEB 2.50 GHz. Again, if the project launches, the response time is expected to improve dramatically, when servers that are more powerful are provided.

Table 3: Experimental questions

No.	Type	Arabic	English
Q1	Who	من هو محمد طنجة؟	Who is Muhammad Tangier?
Q2	When	متى توحدت المملكة العربية السعودية؟	When was Kingdom of Saudi Arabia united?
Q3	What	ماهي الأهرامات المصرية؟	What are Egyptian pyramids?
Q4	Where	أين تقع المملكة العربية السعودية؟	Where is the Kingdom of Saudi Arabia located?
Q5	How much	كم تبلغ درجة حرارة القشرة الأرضية؟	How much is the temperature of the Earth's crust?
Q6	How many	كم عدد سكان الرياض؟	How many residents are there in Riyadh?



Figure 12.a: JAWEB answers for a who question

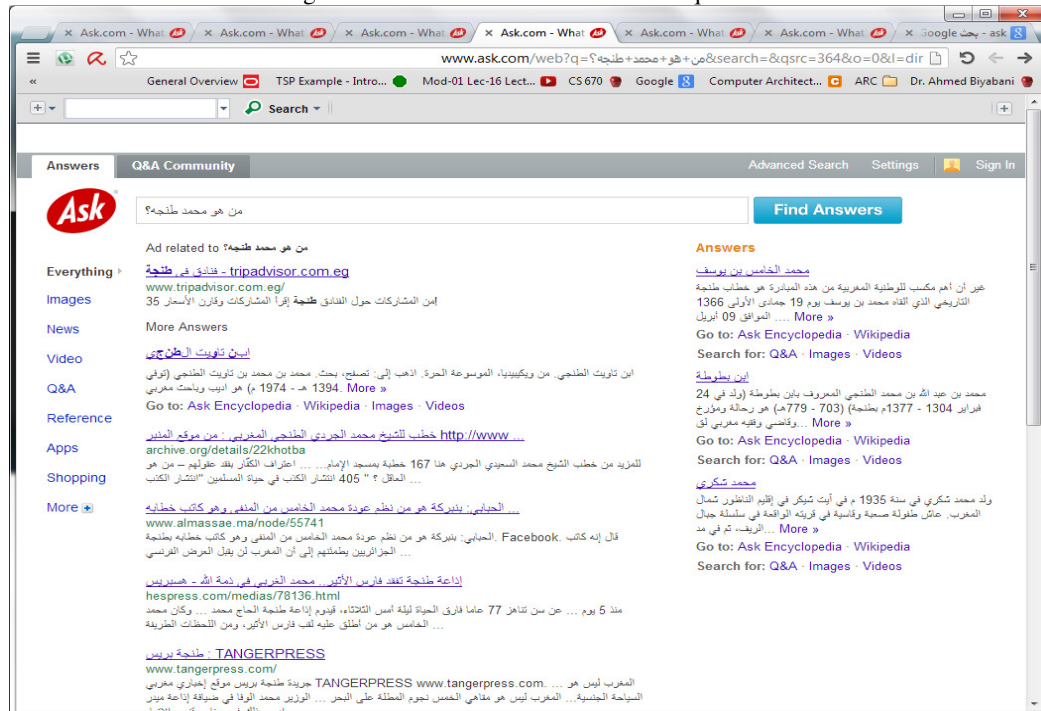


Figure 12.b: Ask answers for a who question



Figure 13.a: JAWEB answers for a when question

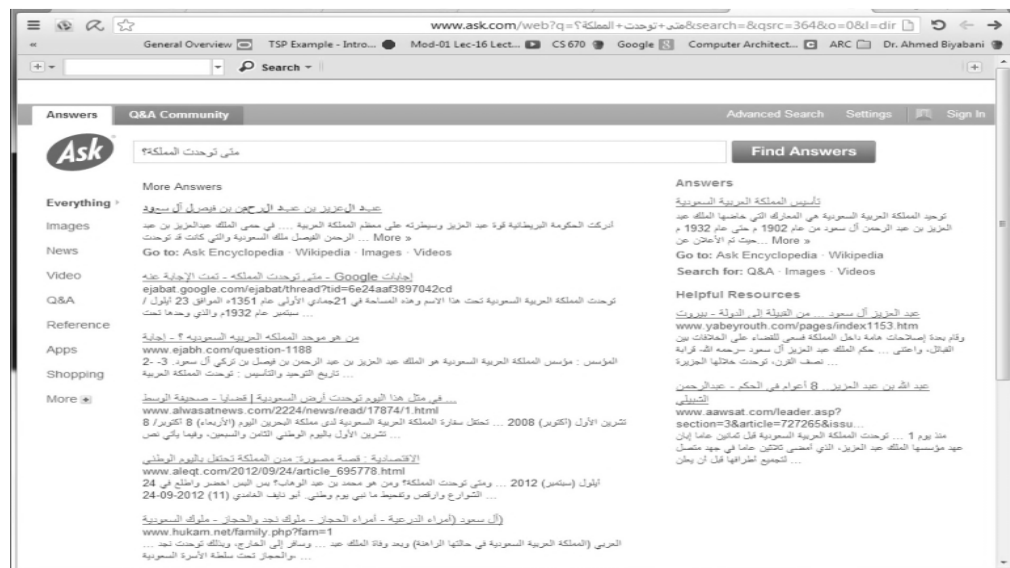


Figure 13.b: Ask.com Answers for when question



Figure 14.a: JAWEB answers for a what question

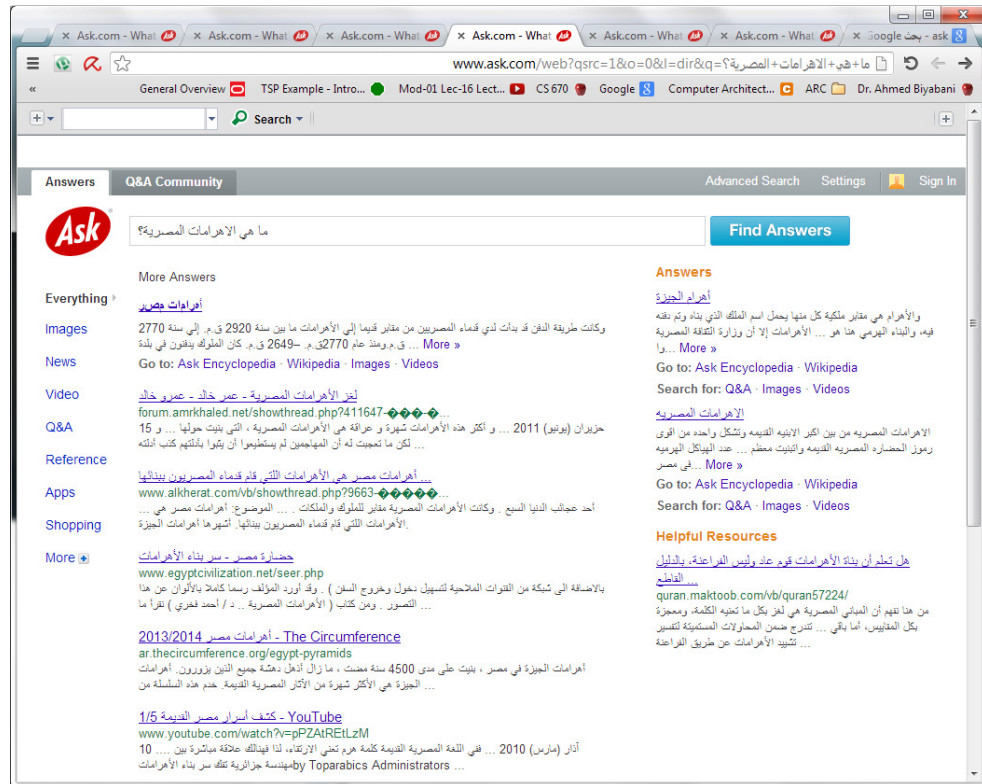


Figure 14.b: Ask answers for a what question



Figure 15.a: JAWEB answers for where question

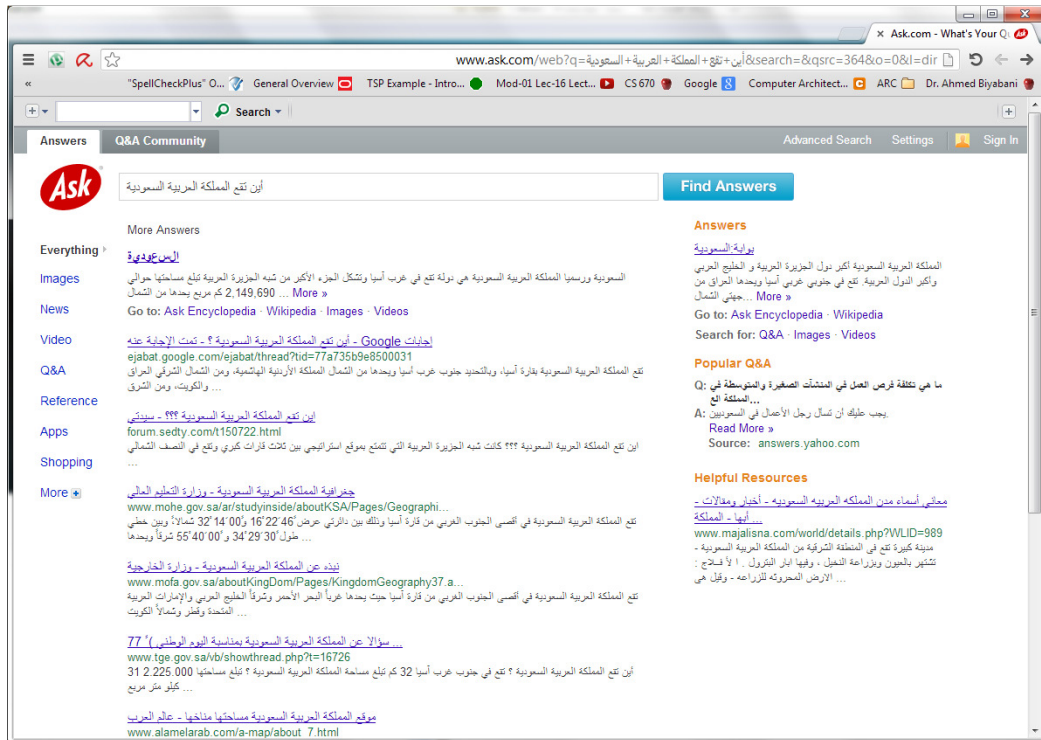


Figure 15.b: Ask answers for a where question



Figure 16.a: JAWEB answers for a how much question

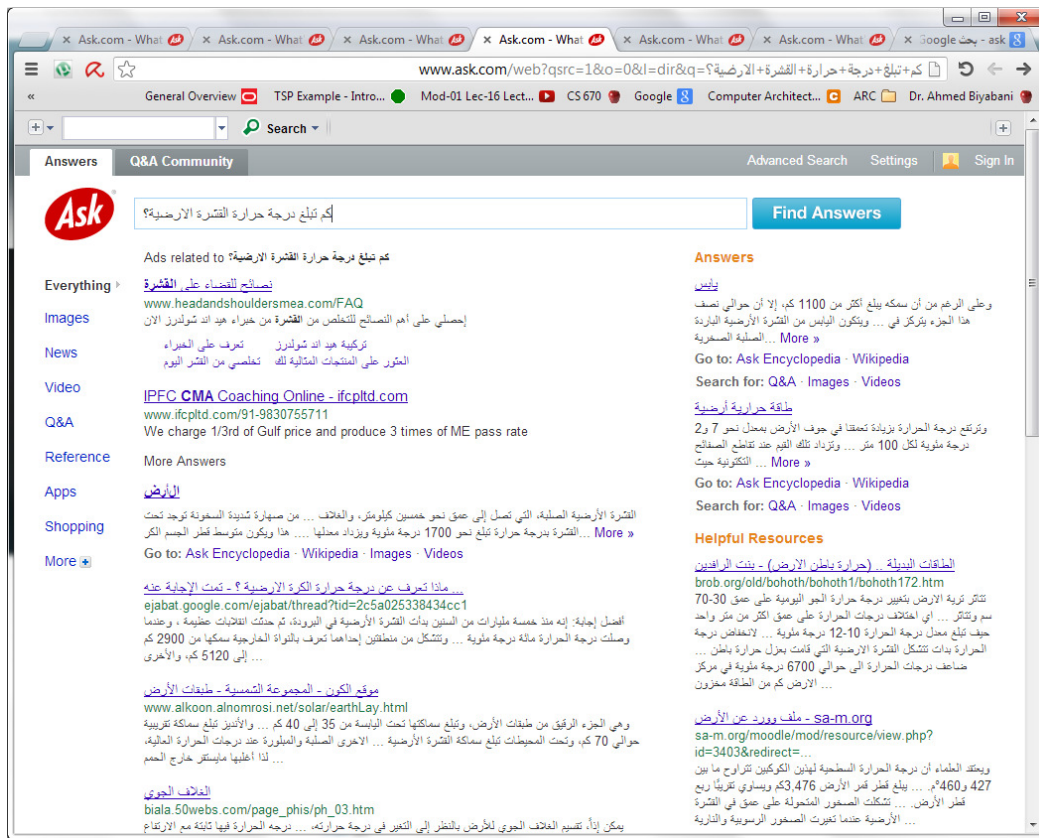


Figure 16.b: Ask answers for a how much question



Figure 17.a: JAWEB answers for a how many question

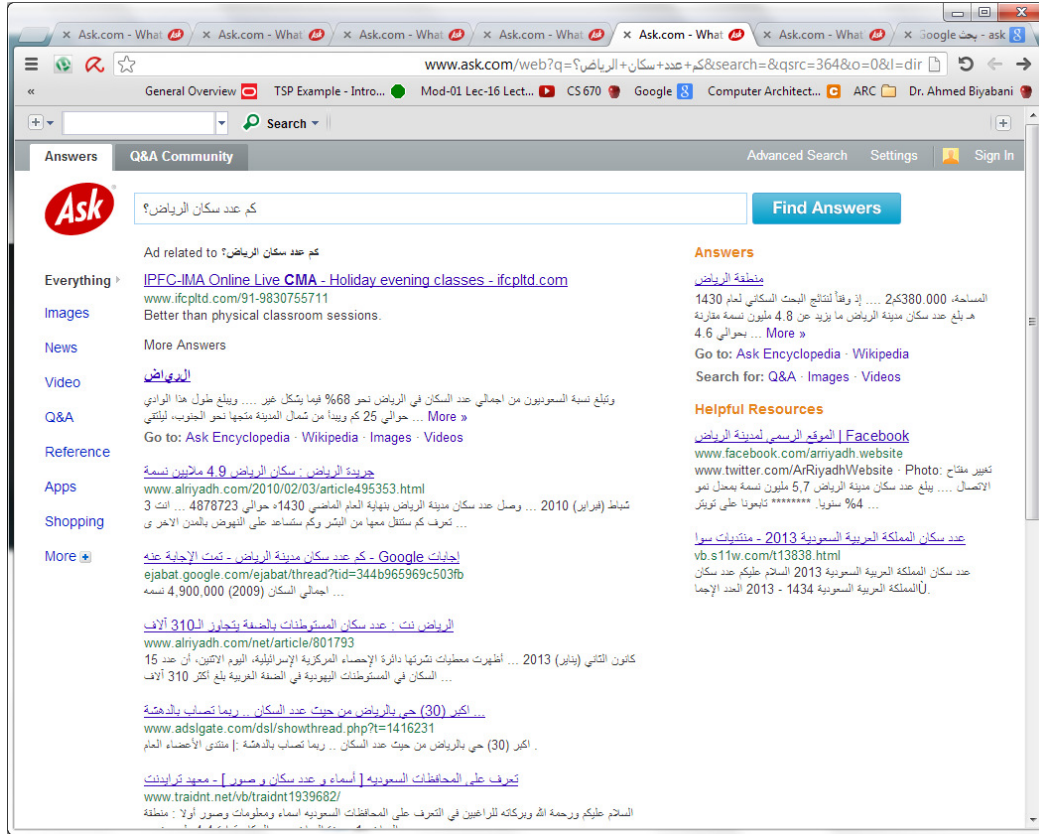


Figure 17.b: Ask answers for a how many question

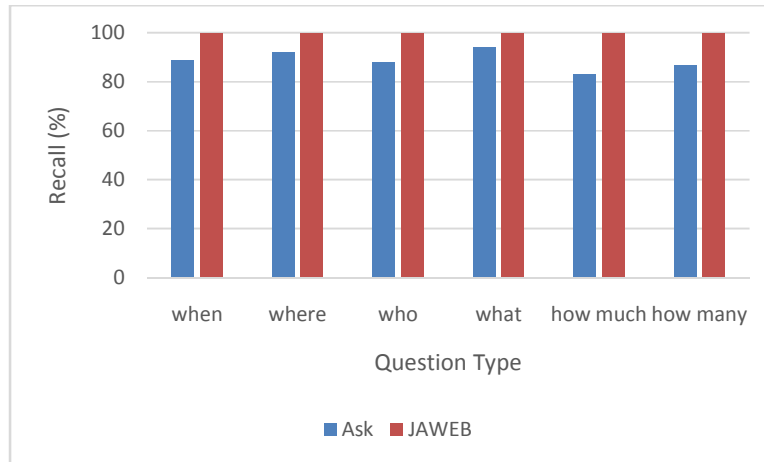


Figure 18: Comparison of the Recall

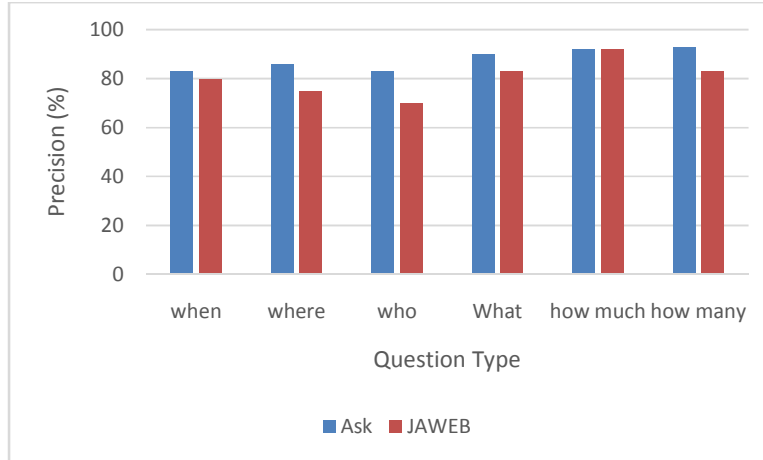


Figure 19: Comparison of the Precision

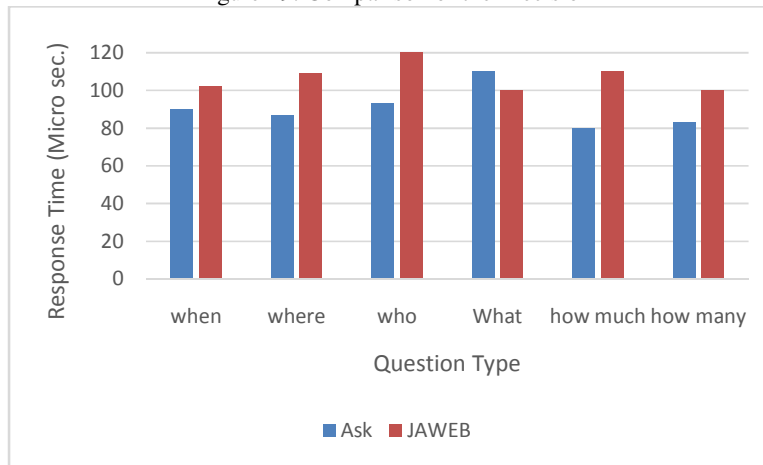


Figure 20: Comparison of the performance-Time

6.CONCLUSION & FUTURE WORK

JAWEB is a web-based Arabic question answering application system. It takes native Arabic questions from the end user and processes them in the server. The processing consists of three modules: question analysis, passage retrieval, and answer extraction. JAWEB finds and extracts accurate answers for the user. It also retrieves the most relevant potential answers and ranks them on the web page.

The evaluation experiment shows the high recall performance of the system, which is attributable to stemming. In the future, we expect to improve user interaction to incorporate user feedback for more precise results. We also plan on using the Arabic Named Entity Recognition to provide more accurate answers. In addition, we will consider the "why" and "how" questions, which require deep linguistic and semantic analysis.

ACKNOWLEDGMENT

We would like to thank Dr. Sman Bekhti and his colleagues for their valuable support and comments based on their Arabic QA system AQUASYS. We would also like to thank Dr. Shereen Khoja for providing us with the Arabic stemmer package that she has developed.

REFERENCES

1. R. Alshalabi "Experimenting with a Question Answering System for the Arabic Language", *Computers & the Humanities*, Vol. 38, pp. 397-415, Nov 2004.
2. K. Arai and A. Handayani "Question Answering System for an Effective Collaborative Learning", *International Journal of Advanced Computer Science and Applications*, Vol. 3, pp.60-64,
3. A. Allam and M. Haggag, "The Question Answering Systems: A Survey," *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, Vol. 2, No. 3, September 2012.
4. D. Hai, L. Kosseim, "The Problem of Precision in Restricted-Domain Question Answering. Some Proposed Methods of Improvement", In *Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains*, Barcelona, Spain, Publisher of Association for Computational Linguistics, July 2004, PP.8-15.
5. A. Lamjiri, "A Syntactic Candidate Ranking Method for Answering Non-copulative Questions," Ph.D. thesis, Concordia University, 2007.
6. P. Walke, S. Karale, "Implementation approaches for various categories of question answering system," In *Proceedings of IEEE Conference on Information & Communication Technologies (ICT)*, 2013, pp.402,407, 11-12 April 2013.
7. R. Alshalabi, "Pattern-based stemmer for finding Arabic roots," *Information Technology Journal*, Vol. 4, pp. 38-43, 2005.
8. R. Grishman, B. Sundheim, "Message Understanding Conference-6: A Brief History," In *Proceedings of International Conference on Computational Linguistics*, 1996.
9. A. Borthwick, "A Maximum Entropy Approach to Named Entity Recognition". PhD thesis, New York University, 1999.
10. Y. Benajiba, P. Rosso, J. Ruiz, "ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy", In *Proceedings of CICLing*, 2007
11. M. Maamouri, A. Bies, T. Buckwalter, "The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus," In *Proceedings of Arabic Language Resources and Tools*, 2004.
12. S. Mesfar, "Standard Arabic formalization and linguistic platform for its analysis," In *Proceedings of The Challenge of Arabic for NLP/MT Conference*, London, England, pp. 84-95, 2006.
13. A. Kabbaj, K. Bouzoubaa, "Amine Platform Page on SourceForge", <http://amine-platform.sourceforge.net/>, checked Aug.2nd, 2013
14. JM Gómez, D. Buscaldi, P. Rosso, E. Sanchis, "JIRS Language-Independent Passage Retrieval System A Comparative Study," In *Proceedings of 5th International Conference on Natural Language Processing, ICON-2007*, Hyderabad, India, January 4-6, 2007
15. W. Black, S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, C. Fellbaum "Introducing the Arabic WordNet project," In *Proceedings of 3rd International WordNet Conference (GWC-06)*, 2006.
16. A. Ezzeldin, M. Shaheen "A Survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends", In *Proceedings of The International Arab Conference on Information Technology*, 2012, pp. 280-287.
17. O. Badawy, M. Shaheen and A. Hamadene "ARQA High-Performance Arabic Question Answering System", In *Proceedings of Arabic Language Technology International Conference*, 2011, pp. 129-136.
18. P. Brihay, "Technical principles of the Morphological Analysis of AraMorph," <http://www.nongnu.org/aramorph/english/principles.html>, checked Aug.2nd, 2013
19. H. Harmanani, W. Keirouz, S. Raheel, "A Rule-Based Extensible Stemmer for Information Retrieval with Application to Arabic," *International Arab Journal of Information Technology*, Vol.3, No. 3, 2006
20. S. Kalaivani, K. Duraiswamy, "Comparison of Question Answering Systems Based on Ontology and Semantic Web in Different Environment," *Journal of Computer Science*, Vol. 8, No. 9, (2012): 1407.

21. O. Ferret, B. Grau, M. Huraults-Plantet, "Finding an answer based on the recognition of the issue focus," In Proceedings of TREC-10.
22. D. Laurent, P. Séguéla, S. NègreCross, "Lingual Question Answering using QRISTAL for CLEF 2006, Lecture Notes in Computer Science, Vol. 4730, 2007, pp. 339-350.
23. S. Parthasarathy, J. Chen "A Web-based Question Answering System for Effective e-Learning," In Proceedings of IEEE International Conference on Advanced Learning Technologies, 2007, pp. 142-146.
24. Ask.com, checked Aug. 2nd, 2013.
25. F. Mohammed, K. Nasser, H. Harb "A knowledge-based Arabic Question Answering System (AQAS)," In Proceedings of ACM SIGART Bulletin, 1993, pp. 21-33.
26. B. Hammo, H. Abu-Salem, S. Lytinen. "QARAB: A Question Answering System to Support the Arabic Language". In Proceedings of the workshop on computational approaches to Semitic languages, 2002, pp. 55-65, Philadelphia.
27. Y. Benajiba, P. Rosso, A. Lyhyaoui "Implementation of the ArabiQA Question Answering System's components.", In Proceedings of Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS, 2007.
28. W. Brini, M. Ellouze, S. Mesfar, L. Belguith "An Arabic Question-Answering system for factoid questionns," In Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2009.
29. <http://www.nooj4nlp.net/pages/introduction.html>, checked may 6th, 2013.
30. S. Bekhti, A. Rehman, M. Al-Harbi, T. Saba "AQuASys: an Arabic Question Answering System Based on Extensive Question Analysis and Answer Relevance Scoring," International Journal of Academic Research, Vol. 3, pp.45-54, July 2011.
31. S. Khoja, "Stemming Arabic Text,"" <http://zeus.cs.pacificu.edu/shereen/research.htm>, checked may 6th, 2013.
32. K. Arai, A. Handayani "Question Answering System for an Effective Collaborative Learning," International Journal of Advanced Computer Science and Applications, Vol. 3, pp.60-64, 2012.