A Modified Version of AlQuAnS: An Arabic Language Question Answering System

Ahmed Abdelmegied, Yasmin Ayman, Ahmad Eid, Nagwa El-Makky, Ahmed Fathy, Ghada Khairy, Khaled Nagi, Mohamed Nabil(✉), and Mohammed Yousri

Computer and Systems Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt
{ahmed.abdelmegied, yasmin.ayman, ahmad.eid, nagwamakky, ahmed.fathy, ghada.khairy, khaled.nagi, mohamed.nabil, mohamed.yousri}@alexu.edu.eg

Abstract. The challenges of the Arabic language and the lack of resources have made it difficult to provide Arabic Question Answering (QA) systems with high accuracy. These challenges motivated us to propose AlQuAnS-an Arabic Language Question Answering System that gives promising accuracy results. This paper proposes a modified version of AlQuAnS with a higher accuracy. The proposed system enhances the accuracy of the question classification, semantic interpreter and answer extraction modules. The provided performance evaluation study shows that our modified system outperforms other existing Arabic QA systems, especially with the newly introduced answer extraction module.

Keywords: Arabic question answering systems · Arabic morphological analysis · Question analysis · Question classification · Answer extraction · Semantic analysis · Question expansion

1 Introduction

Question Answering has gained great attention lately after the great progress in the field of Natural Language Processing (NLP), where Question Answering improves the search experience by suggesting an explicit answer for a user's provided question.

The process of question answering is as follows, when a user provides a question, this question is analyzed from a linguistic point of view, in attempt to predict the expected answer. Afterwards, related documents are searched to retrieve a valid answer for the provided question.

Great research efforts are made to provide reliable QA in different languages. unfortunately, Arabic QA doesn't gain great attention in these contributions although on 2016, 26 countries are using Arabic as their main language and 420 million people around the world talk Arabic which makes Arabic the 6th most spoken language.

The main reasons for the few attempts for building Arabic Question Answering systems are scarceness of Arabic datasets and the richness of Arabic morphology. The Arabic morphological richness imposes the need for intelligent morphological analyzer to process Arabic text.

A. Fred et al. (Eds.): IC3K 2017, CCIS 976, pp. 184–199, 2019.
https://doi.org/10.1007/978-3-030-15640-4_10

1.1 Arabic Morphology

Arabic is a rich morphological language which needs intelligent algorithms to analyze its text. Morphological analysis is a process of figuring out the word features; such as:

- root or stem,
- morphological pattern,
- part-of-speech (noun, verb or particle)
- number (singular, dual or plural)
- case or mood (nominative, accusative, genitive or jussive)

of the word. The root-patterned nonlinear morphology of Arabic makes both theoretical and computational processing for Arabic text extremely hard.

Since Arabic has a completely different orthography based on standard Arabic script going from right to left. Also, each letter has *three* different shapes depending on its position within the word, where each letter may have a diacritic sign above or below the letter. Changing the diacritic of one letter may change the meaning of the whole word. Printed and online text come usually *without* diacritics leaving plenty of room for word ambiguity.

Arabic is also a highly *derivational* language. It is a highly *inflectional* language as well.

$$\texttt{Word} = \texttt{prefix(es)} + \texttt{lemma} + \texttt{suffix(es)}.$$

The prefixes can be articles, prepositions or conjunctions; whereas the suffixes are generally objects or personal/possessive anaphora. Both prefixes and suffixes can be combined, and thus a word can have zero or more affixes. Figure 1 shows an example of the composition of an Arabic word.

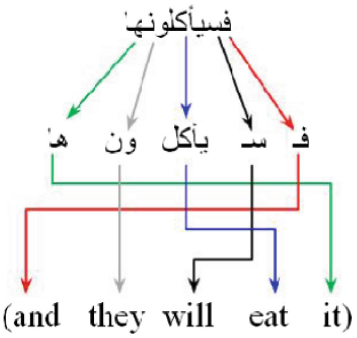

Fig. 1. Example Arabic inflection [1].

The absence of capital letters is another challenge in the Named Entity Recognition (NER) in Arabic [2, 3] Lots of Arabic names are adjectives; such as, “gameel” (handsome), “zaki” (intelligent), or “khaled” (immortal).

Last but not least, from a statistical viewpoint, if Arabic texts are compared to texts written in other languages which have a less complex morphology, Arabic texts look much sparser because of the inflectional characteristic of the language that we mentioned above. This specific characteristic of the language is the reason that makes it more difficult to handle each task in Natural Language Processing (NLP).

2 Motivation

Because of the mentioned challenges, we were motivated to propose AlQuAnS [4] – An Arabic Language Question Answering System that gives promising accuracy results. In this paper, we propose a modified version of AlQuAnS with improved accuracy of various modules. The overall performance results show that this modified version gives enhanced performance compared to past related work.

The rest of the paper is organized as follows. In Sect. 3, we give a short survey on the related standard work in the field of QA systems. Then, we focus on Arabic QA systems. Section 4 contains an overview of the system architecture of AlQuAnS and description of each system component. In Sect. 5 we show our modifications to AlQuAnS. The system evaluation is presented in Sect. 6. Section 7 concludes the paper and presents some ideas for our future work in this area.

3 Related Work

A Question Answering system is a system that takes an input question from the user, retrieves the related result sets to the question topic and then extracts an exact answer to the question to be returned to the user. A typical state-of-the-art Question Answering system, divides the Question Answering task into *three* core components:

- Question Analysis (including Question preprocessing and classification),
- Information Retrieval (or document retrieval),
- Answer Extraction.

Question classification plays an essential role in QA systems by classifying the submitted question according to its type. Information retrieval is very important for question answering, because if no correct answers are present in a document, no further processing can be carried out to find the answer. Finally, answer extraction aims at retrieving the correct passage containing the answer within the retrieved document.

3.1 Latin QA Systems

Kngine stands for *Knowledge Engine* (Kngine, n.d.) is a knowledge engine that is designed to give direct answers for questions. Kngine leverages natural language processing, machine learning, and data mining algorithms to build an extraction engine, that learns meaningful concepts, information and relationships from data. The authors claim that Kngine is the world's first multi-language question answering engine. Currently, Kngine supports English and other languages as Arabic, German and Spanish based on translation from the English language.

The authors of [5] propose an offline strategy for QA in which information is extracted automatically from electronic offline texts and stored for quick and easy access. The system extracts semantic relations (e.g., concept-instance relations) between lexical items using multiple Part-Of-Speech (POS) patterns. Then, it filters out the noise from the extracted relations using a machine-learned classifier. At the end, it tests the feasibility of this strategy on one semantic relation and a challenging subset of questions, which can be extended to include other types of questions.

LAMP [6] is a web-based QA system that takes advantage of the snippets in the search results returned by a search engine like Google. Asking a question such as "Who was the first American in space?", LAMP submits the question to Google and grabs its top 100 search results. Then, the system utilizes a Support Vector Machine (SVM) to classify the questions (e.g., this question asks for a person name). The system then extracts all information of the same type from the search result sets as *plausible answers*, using a Hidden Markov Model (HMM)-based named entity recognizer. For each plausible answer, the system constructs a snippet cluster which is composed of all the snippets containing that answer. For each plausible answer, the system constructs a snippet cluster which is composed of all the snippets containing that answer. Finally, the system uses cosine similarities between clusters and the question to get the best cluster that matches the question and returns the named entity that represents the best matching cluster.

The work of [7] addresses the problem where queries and relevant textual content significantly differ in their properties and are difficult to match with traditional information retrieval methods. A novel algorithm is presented that analyses the dependency structures of the known valid answer sentence. These acquired patterns are used to more precisely retrieve relevant text passages from the underlying document collection. The positions of key phrases in the answer sentence relative to the answer itself are analyzed and linked to a certain syntactic question type. The algorithm does not require a candidate sentence to be similar to the question in any respect.

IBM developed a statistical QA system for TREC 9 [8]. It is an application of maximum entropy classification for question and answer *type* prediction and named entity marking dealing only with *fact*-based questions. The system retrieves documents from a local encyclopedia. Then, it performs query expansion and, finally, does passage retrieval from the TREC collection. The paper also describes the answer selection algorithm. It determines the best sentence given the question and the occurrence of the expected answer type by minimizing various distance metrics applied over phrases or windows of text. For TREC 10 [9], IBM adapted the system to de with definition type questions and completed the trainability aspect of their QA system. The authors introduce the following:

- new and refined answer tag categories,
- query expansion lists,
- focus expansion using WordNet,
- dependency relationships using syntactic parsing, and
- a maximum-entropy formulation for answer selection.

For TREC 11, IBM collected a 4,000 question-answer corpus based on trivial questions for training and developed answer patterns for the TREC collection of documents [10]. The authors added three more features including:

- the occurrence of the answer candidate on the web,
- the re-ranking of answer candidate window using a statistical MT dictionary, and
- developed lexical patterns from supervised training pairs.

3.2 Arabic QA Systems

QARAB [11] is a QA system to support the Arabic language. The system is based on the *three*-module generic architecture:

- question analysis,
- passage retrieval, and
- answer extraction.

It extracts the answer from a collection of Arabic newspaper text. For that, it uses a keyword matching strategy along with matching simple structures extracted from both the question and the candidate documents selected by the information retrieval module using an existing tagger to identify proper names and other crucial lexical items. The system builds lexical entries for them on the fly. For system validation, four native Arabic speakers with university education presented 113 questions to the system and judged whether the answers of the system are correct or not.

The Arabic language was introduced for the first time in 2012 in the QA4MRE lab at CLEF [12]. The intension of the research is to ask questions which require a deep knowledge of individual short texts and in which systems are required to choose one answer from multiple answer choices. The work uses shallow information retrieval methods. Unfortunately, the overall accuracy of the system is 0.19 and the questions proposed by CLEF are suitable only for modern Arabic language.

ALQASIM [13] is an Arabic QA selection and validation system that answers multiple choice questions of QA4MRE @ CLEF 2013 test-set. It can be used as a part of the answer validation module of any ordinary Arabic QA system. It comes up with a new approach like the one used by human beings in reading tests. A person would normally read and understand a document thoroughly, and then begins to tackle the questions. So, the suggested approach divides the QA4MRE process into *three* phases:

- document analysis,
- locating questions and answers,
- answer selection.

ArabiQA [1] is a QA system that is fully oriented to the modern Arabic language. ArabiQA is obeying to the general norms reported at the CLEF conference. However, the system is not complete yet. The following points is a part of the researchers' investigation, as listed in their work:

- The adaptation of the JIRS passage retrieval system to retrieve passages from Arabic text.
- The development of the annotated ANERcorp to train the Named Entity Recognition (NER) system.

- The development of the ANERsys Named Entity Recognition system for modern Arabic text based on the maximum entropy approach.
- The development of an Answer Extraction module for Arabic text for *factoid* questions (Who, where and when questions).

DefArabicQA [14] presents a definitional QA system for the Arabic language. The system outperforms the use of web searching by two criteria. It permits the user to ask an ordinary question (e.g., "*What is X?*") instead of typing in a keyword-based query. It then attempts to return an accurate answer instead of mining the web results for the expected information. The question topic is identified by using *two* lexical question patterns and the answer type expected is deduced from the *interrogative pronoun* of the question. Definition ranking is performed according to three scores: a pattern weight criterion, a snippet position criterion, and a word frequency criterion.

The IDRAAQ [15] system is another Arabic QA system based on *query expansion* and *passage retrieval*. It aims at enhancing the quality of retrieved passages with respect to a given question. In this system, a question analysis and classification module to extract the keywords, identify the structure of the expected answer and form the query to be passed to the Passage Retrieval (PR) module. The PR extracts a list of passages from an Information Retrieval process. Thereafter, this module performs a ranking process to improve the relevance of the candidate passages. Finally, the Answer Validation (AV) module validates an answer from a list of candidate answers.

Al-Bayan [16] is a *domain specific* Arabic QA system for the Holy Quran. It takes an Arabic question as input and retrieves semantically relevant verses as candidate passages. Then, an answer extraction module extracts the answer from verses obtained accompanied by their Tafseer (standard explanations of Quran). The system has *four* functionalities:

- It merges *two* Quranic ontologies and uses *two* Tafseer books.
- It applies a semantic search technique for information retrieval.
- It applies a state-of-the-art technique (SVM) for question classification.
- It builds Quranic-based training data sets for classification and Named Entities Recognition (NER).

4 System Architecture

4.1 Overview

Our proposed system is a modification of AlQuAnS – An Arabic Language Question Answering System [4]. We share the same architecture of AlQuAns except for some enhanced modules. Figure 2 shows the system architecture with modified modules highlighted.

AlQuAns is a QA system that works on an open domain and supports the Arabic language. The system consists of 4 main modules:

- Arabic pre-processor module: which preprocess input questions to make the data retrieval more accurate.
- Question analysis module: which includes 2 submodules; query expansion and question classification.

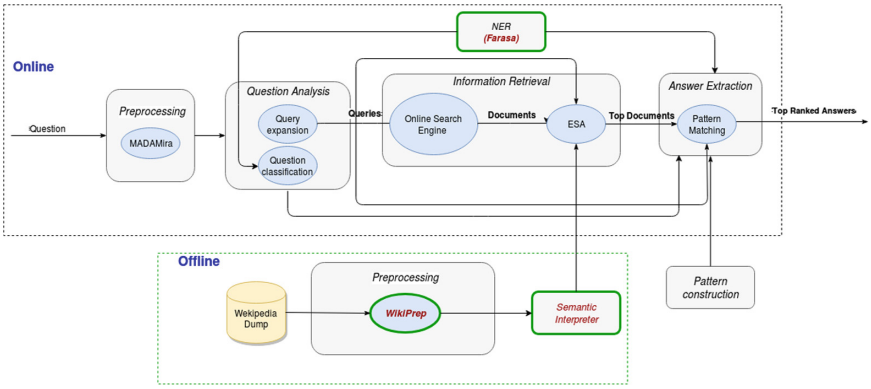

Fig. 2. The overall system architecture (adapted from [4]).

- Semantic information retrieval module: which can retrieve the semantically related documents.
- Answer extraction module: which extracts the ranked answers to the input questions from the retrieved documents with high accuracy.

Arabic Preprocessor. Text Preprocessing is done by applying morphological analysis software to identify the structure of the text. Typical operations include:

- normalization,
- stemming,
- Part-Of-Speech (POS) tagging, and
- stop words removal.

Morphologically, the Arabic language is one of the most complex and rich languages. Thus, morphological analysis of the Arabic language is one of the complex tasks that has been popular in recent research. The AlQuAnS relies on MADAMIRA [17]. MADAMIRA combines the best aspects of two previously commonly used systems for Arabic processing, MADA found in [18] and AMIRA found [19].

MADA is a system for Morphological Analysis and Disambiguation for Arabic. The primary purpose of MADA is to, given raw Arabic text, derive as much linguistic information as possible about each word in the text, thereby reducing or eliminating any ambiguity surrounding the word. MADAMIRA also includes TOKAN, a general tokenizer for MADA-disambiguated text [18]. TOKAN uses the information generated by the MADA component to tokenize each word according to a highly-customizable scheme. AMIRA is a system for tokenization, part-of-speech tagging, Base Phrase Chunking (BPC) and Named Entity Recognition (NER).

Question Analysis. The AlQuAnS divides this module to Query Expansion (QE) and Question Classification. For query expansion, the content and the semantic relations of the Arabic WordNet (AWN) ontology [20] are used. The AWN ontology is a free resource for modern standard Arabic. It is based on the design and the content of Princeton WordNet (PWN) [21]. It has a structure similar to wordnets and exists for

approximately 40 languages. It is also connected to the Super Upper Merged Ontology (SUMO) [22]. SUMO is an upper level ontology which provides definitions for general purpose terms and acts as a foundation for more specific domain ontologies.

Our semantic QE approach uses *four* semantic relations among those existing between AWN synsets (items), words and forms. Therefore, the approach defines four sub-processes for the query expansion:

- QE by synonyms,
- QE by definitions,
- QE by subtypes, and
- QE by supertypes.

For question classification, The AlQuAnS uses the Support Vector Machines (SVM) classifier since it has shown to produce the best results during our experiments. Question classification needs a taxonomy to classify question types. AlQuAnS uses the work of Li and Roth [23]. which provides a hierarchical classifier, taxonomy and data to be used in English question classification. Since 2002, Li and Roth work has been used by all researchers who are interested in building QA systems. They propose a two-layered question taxonomy which contains six coarse grained categories and 50 fine grained categories. The coarse-grained categories are listed below.

- Abbreviation
- Description
- Entity
- Human
- Location
- Numeric value

In AlQuAnS we limit the taxonomy to LocationCity, LocationCountry, Human, Individual, NumericDate). With these *four* sub-categories, we focus more on a QA system that can answer questions that ask for cities, countries, humans individuals and different kind of dates (birthdays, event dates, etc.).

Information Retrieval. The Information Retrieval module consists of two submodules: The *Online Search Engine* and the *Passage Retrieval* submodules. Our system is designed to interface with common search engine modules. However, in our implementation, we choose the Yahoo API to be comparable to previous systems using the same API, e.g. [15]. For the *Passage Retrieval* submodule, AlQuAnS we construct a general Semantic Interpreter (SI) that can represent text meaning. The Semantic Interpreter depends on the Explicit Semantic Analysis (ESA) approach proposed in [24]. More details on this module are given in Sect. 5.1.

Answer Extraction. The purpose of the Answer Extraction (AE) module is to search for candidate answers within the relevant passages and extract the most likely answers. Using certain patterns for each type of question is the main approach. In general, patterns can be written by people or learnt from a training dataset. The type of the expected answers is always taken into consideration. The AlQuAns used a Named Entity Recognition system with the patterns extracted for each question type by adapting the approach proposed in [4].

The Answer Extraction module of AlQuAns is composed of *three* phases. The first and second phases are based on the approach in [4]. The *first* phase is to use the web documents retrieved by the Passage Retrieval module to construct a table of patterns for each question type. The *second* phase is to rank these patterns by calculating their corresponding precision. The *third* phase is to find the answer using the extracted answer patterns then filter the answers using the NER of MADAMIRA.

5 Proposed Modifications

In this paper, we enhanced the results of "AlQuAns" by modifying 2 modules, namely the Semantic Interpreter module and the Named Entity Recognition (NER) module. The NER module affects both the Answer Extraction module and the Question Classification module. The following subsections describe the modifications done in detail.

5.1 Semantic Interpreter

In AlQuAnS, we used the Explicit Semantic Approach (ESA) proposed in [24] with 11,000 Arabic Wikipedia documents to build the semantic interpreter. This leads to a short concept vector and hence less accuracy. In this paper, we use the whole Arabic Wikipedia dump of January 2018 which includes more than 1 million documents to build the semantic interpreter. This leads to a larger concept vector that enhances the accuracy of the semantic interpreter and the overall system performance. The following subsections describes in detail the Explicit Semantic Analysis approach and the way used in our system to compute the semantic relatedness between the question and the candidate answers.

Explicit Semantic Analysis. Given a set of concepts, C_1, …, C_n, and a set of associated documents, d_1, …, d_n, we build a sparse table T where each of the n columns corresponds to a concept, and each of the rows corresponds to a word that occurs in $\cup_{i=1\ldots n} d_i$. An entry T [i, j] in the table corresponds to the term frequency–inverse document frequency (tf-idf) value of term t_i in document d_j.

$$T[i,j] = tf\left(t_i, d_j\right).log\frac{n}{df_i} \tag{1}$$

Where term frequency is defined as:

$$tf\left(t_i, d_j\right) = \{1 + log\ count\left(t_i, d_j\right)\ if\ count\left(t_i, d_j\right) > 0\ 0\ otherwise \tag{2}$$

and $df_i = |\{d_k: t_i \in d_k\}|$ is the number of documents in the collection that contains the term t_i (*document frequency*). Finally, cosine normalization is applied to each row to discard differences in document length:

$$T[i,j] \leftarrow \frac{T[i,j]}{\sqrt{\sum_{l=1}^{r} T[i,j]}} \tag{3}$$

where r is the number of terms.

The semantic interpretation of a word t_i is obtained as a row i of table T. In other words, the meaning of a word is given by a vector of concepts paired with their tf-idf scores, which reflects the relevance of each concept with respect to the word. The semantic interpretation of a text fragment, $\langle t_1, \ldots, t_k \rangle$, is the centroid of the vectors representing the individual words. This definition allows us to partially perform word sense disambiguation. Consider, for example, the interpretation vector for the term "mouse". It has two sets of strong components, which correspond to two possible meanings: "mouse (rodent)" and "mouse (computing)". Similarly, the interpretation vector of the word "screen" has strong components associated with "window screen" and "computer screen". In a text fragment such as "I purchased a mouse and a screen", summing the two interpretation vectors will boost the computer-related components, effectively disambiguating both words. Table T can also be viewed as an inverted index, which maps each word to a list of concepts where it appears. Inverted index provides a very efficient computation of distance between interpretation vectors.

Computing the Semantic Relatedness. ESA represents text as interpretation/concept vectors in the high-dimensional space of concepts. With this representation, computing semantic relatedness of text simply amounts to compare their vectors.

The user's question is passed to a search engine, e.g., Yahoo or Google, and the retrieved snippets are ranked using ESA. To determine the semantic relatedness between the question and the retrieved snippets, we compute the concept vectors of the question and the snippets using the Explicit Semantic Analysis module. Then, we compute the cosine similarity between the question concept vector and the concept vector of each snippet i. The result scores are used to select the top-scoring snippets that are relevant to the question. The more similar the snippets vector to the query vector is, the more likely it is related to the question as illustrated in Fig. 3.

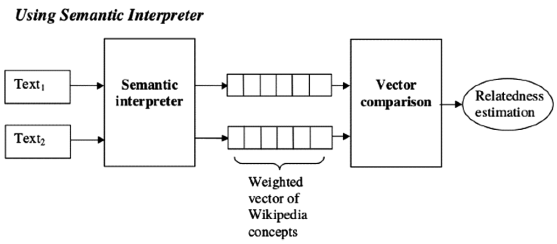

Fig. 3. The semantic relatedness between a question and a snippet (adapted from [25]).

5.2 Named Entity Recognition

The Answer Extraction module finds the matching patterns regardless of the answer word type, so it can extract irrelevant words as an answer like prepositions. Pattern <Name> <Answer> may give answer "In Pyramids" and considers the proposition "In" to be an answer. That is why we use a NER to filter the answer. Using the NER of MADAMIRA, we check the answers because we restrict ourselves to four types of questions: LocationCountry, LocationCity, HumanIndividual and NumericDate. NER of MADAMIRA can find the named entities of words in the three major

categories: LOC, PERS and ORG. Other words that do not belong to these categories will not be added to the dictionary made for the word types. The answers of LocationCountry and LocationCity questions are expected to be a location. For the NumericDate question type, we check its validity by checking if the word is a number or a month name. So, the system checks the words in the dictionary to make sure that these words are recognized to be entities. Our best approach is to check if these words are entities or not to be accepted as an answer.

The NER of Farasa [26] vs the NER of MADAMIRA. In AlQuAns we use the NER of MADAMIRA, that gives good results in general. But we found in our experiments that, for the Organization type, this NER fails in detecting English words that are translated to Arabic by just changing letters, (i.e., transliterated), e.g., UNICEF = . Farasa [26], which is a new alternative to MADAMIRA, claims to detect this type of words with a better accuracy due to the relatively small MADAMIRA's English/Arabic dictionary.

In addition, Farasa's performance regarding other categories of entities is also better than that of MADAMIRA. Table 1 shows the results of an experimental study that we conducted on the NER of Farasa and MADAMIRA, using Trec and Clef datasets [15].

Table 1. Accuracy comparison between the NER of MADAMIRA and Farasa.

	Precision		Recall		F-measure	
	MadaMira	Farasa	MadaMira	Farasa	MadaMira	Farasa
PERS	0.77	0.87	0.63	0.87	0.7	0.86
LOC	0.82	0.97	0.72	0.62	0.77	0.76
ORG	0.63	0.76	0.53	0.56	0.57	0.65
Total	0.79	0.89	0.66	0.68	0.71	0.78

5.3 Question Classification

The performance of the question classification module is also improved because of using a more accurate NER. This module specifies the question type to be used later in the answer extraction module, so its importance increases due to the direct dependency of the answer extraction on it. This module is composed of 3 components:

1. MADAMIRA as a stemmer and as a POS tagger
2. Farasa as an NER
3. SVM classifier, which takes a bag of words, the question term and the recognized named entities as features.

6 System Evaluation

We measure the performance improvements made to the new version of AlQuAnS using the standard metrics for QA evaluation, namely, *Accuracy*, *Mean Reciprocal Rank* (MRR) and *Answered* Questions (AQ). Also, we include previously established Arabic question answering systems result, such as [1, 15].

The definitions of these metrics are given below.

$$Accuracy = \frac{1}{N_s}\sum_{k \in S} V_{k,1} \tag{4}$$

$$MRR = Average_{k \in S}\left(\frac{1}{5}\sum_{j=1}^{5} V_{k,j}\right) \tag{5}$$

$$AQ = \frac{1}{N_s}\sum_{k \in S} max\left(V_{k,j}\right) \tag{6}$$

Where $V_{k,j}$ equals 1 if the answer to question k is found in the passage having the rank j, 0 otherwise.

The following subsections include the evaluation results of the question classification module, the answer extraction module and the overall modified system.

6.1 Question Classification

As stated previously, the NER affects the performance of the question classification module. Using the taxonomy of Li and Roth [23], Table 2 shows the results of the modified system and AlQuAns when applied on Clef and Trec dataset [15].

Table 2. Comparison between the first version of AlQuAnS and its modified version.

	Precision		Recall		F-measure	
	AlQuAns	Modified version	AlQuAns	Modified version	AlQuAns	Modified version
Entity	26.70%	26.70%	22.20%	22.20%	24.20%	24.20%
Human	70.80%	73.90%	86.30%	89.00%	77.80%	80.70%
Location	78.10%	81.30%	71.40%	74.30%	74.60%	77.60%
Number	97.80%	97.80%	84.60%	86.50%	90.70%	91.80%

6.2 Answer Extraction

Using the CLEF and TREC datasets used in evaluating the system in [15], we divide the datasets into training and testing sets. Together, they consist of 2,242 questions that pass through the Answer Extraction module. The results are shown in Table 3 which compares the accuracy of the Answer Extraction modules of the first version of AlQuAnS, its modified version and the corresponding module in Abouenour [15].

Table 3. Comparison between answer extraction accuracies.

	Abouenour	AlQuAns	Modified
Answer extraction	15.3%	50.9%	55.66%

6.3 Explicit Semantic Analysis Evaluation

In this subsection, we are going to compare the overall performance of the modified version of our system before and after enhancing the Explicit Semantic Analysis (ESA) component (by using the whole Arabic Wikipedia dump of January 2018 for building the semantic interpreter). Again, we benchmark the system against the work in [15]. For that, we managed to get the same training and benchmarking datasets. We update these datasets to adapt the answers since search engines deliver different; yet correct; results over time. For example, "How many Syrian refugees live in Jordan?". The answer changes each year.

As in the previous version of AlQuAnS, the ESA component provides a list of documents for each question ranked in decreasing order based on relatedness to the question.

As in AlQuAnS, we give the question and the set of documents to ESA module which process both to produce a list of most five related documents in decreasing order of relatedness to the given question. In AlQuAns, we proposed a method for ESA evaluation - which was used to get comparable results with the system in [15]. In this method, each document in the list is examined to check if there is an exact match with the answer or not. If this is true, then the document is considered a candidate that contains the question answer.

A criticism of this method is that if the answer is "10الاف سنويا" and the document contains "10الاف خلال السنة", this document will not be considered a candidate that contains the answer. To avoid false negative cases, like the one mentioned above, we propose another method, Method 2, where instead of searching for the whole text we verify that all terms of the answer exit in addition to more than 25% of the question terms.

Due to high Arabic inflection, we propose a third method, Method 3, in which we stem documents, questions and answers terms and then apply method 2. Also, we combine some question terms with answer terms which can give better results. For example, consider the question:

"كم تتربح غوغل سنويا؟" and the answer which is just "10 مليون". If we depend only on the answer terms to judge whether a document is a candidate to contain the question's answer or not, this may produce a false positive result since the answer terms don't enforce to whom this 10 million belong.

Tables 4, 5 and 6 present the results of the overall system evaluation of AlQuAnS and its modified version (compared to [15]) for each of the proposed methods.

Table 4. Overall system evaluation using method 1.

	Accuracy	QA	MRR
Abouenour	20.20%	26.74%	9.22
AlQuAns	22.79%	47.67%	8.1
Modified version	23.32%	43.3%	7.8

Table 5. Overall system evaluation using method 2.

	Accuracy	QA	MRR
AlQuAns	22.2%	43%	4.4
Modified version	25.9%	49.2%	5.1

Table 6. Overall system evaluation using metric 3.

	Accuracy	QA	MRR
AlQuAns	31.69%	58%	6.3
Modified version	36.7%	58%	7.3

7 Conclusions and Future Work

Question answering is becoming more and more an essential part of our communication with the different devices and not only limited to search engines as it used to be. For example, chat bots are becoming the new way to communicate with computers. The importance of the question answering is increasing, to become a corner stone in the Natural Language Processing (NLP) field in the coming years.

In this paper, we propose a modified version of our Arabic Language Question Answering System (AlQuAnS). The proposed version enhances the accuracy of the question classification, semantic interpreter and answer extraction modules. We tried to push the currently existing components of the system to their limits to show that without introducing new components, just with more efforts with the current architecture and modules, we can reach a higher accuracy. The provided performance evaluation study shows that our modified version outperforms the previous version in addition to other existing Arabic QA systems.

Deep learning became one of the main components in many of the Natural Language Processing tasks, because of the impressive results it provides without the need to feature engineering. We intend to extend our work by applying deep learning techniques which are expected to give more enhancement to the overall performance of our system.

References

1. Benajiba, Y., Rosso, P.: Arabic Question Answering. Diploma of Advanced Studies. Technical University of Valencia, Spain (2007)
2. Benajiba, Y., Rosso, P.: Anersys 2.0: conquering the NER task for the Arabic language by combining the maximum entropy with pos-tag information. In: Proceedings of Workshop on Natural Language-Independent Engineering, IICAI-2007 (2007)
3. Benajiba, Y., Rosso, P., BenedíRuiz, J.M.: ANERsys: an Arabic named entity recognition system based on maximum entropy. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70939-8_13

4. Nabil, M., et al.: AlQuAnS – an Arabic language question answering system. In: Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR, pp. 144-154 (2017). https://doi.org/10.5220/0006602901440154. ISBN 978-989-758-271-4
5. Fleischman, M., Hovy, E., Echihabi, A.: Offline strategies for online question answering: answering questions before they are asked. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, Stroudsburg, PA, USA, pp. 1–7. ACL 2003. Association for Computational Linguistics (2003)
6. Zhang, D., Lee, W.S.: A web-based question answering system. In: Proceedings of the SMA Annual Symposium 2003. Singapore (2003)
7. Kaiser, M.: Answer sentence retrieval by matching dependency paths acquired from question/answer sentence pairs. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2010, Stroudsburg, PA, USA, pp. 88–98. Association for Computational Linguistics (2012)
8. Ittycheriah, A., Franz, M., Zhu, W.-J., Ratnaparkhi, A., Mammone, R.J.: IBM's statistical question answering system. In: Proceedings of the Text Retrieval Conference. TREC-9 (2000)
9. Martin, A.I., Franz, M., Roukos, S.: IBM's statistical question answering system-TREC-10. In: Proceedings of the 10th Text Retrieval Conference. TREC-10 (2001)
10. Ittycheriah, A., Roukos, S.: IBM's statistical question answering system-TREC-11. IBM Thomas J Watson Research Center, Yorktown Heights, NY (2006)
11. Hammo, B., Abu-Salem, H., Lytinen, S.: QARAB: a question answering system to support the Arabic language. In: Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages, SEMITIC 2002, Stroudsburg, PA, USA, pp. 1–11. Association for Computational Linguistics (2002)
12. Trigui, O., Belguith, L.H., Rosso, P., Amor, H.B., Gafsaoui, B.: Arabic QA4MRE at CLEF 2012: Arabic question answering for machine reading evaluation. In: CLEF (Online Working Notes/Labs/Workshop) (2012)
13. Ezzeldin, A.M., Kholief, M.H., El-Sonbaty, Y.: ALQASIM: Arabic language question answer selection in machines. In: Forner, P., Müller, H., Paredes, R., Rosso, P., Stein, B. (eds.) CLEF 2013. LNCS, vol. 8138, pp. 100–103. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40802-1_12
14. Trigui, O., Belguith, H., Rosso, P.: DefArabicQA: Arabic definition question answering system. In: Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC, pp. 40–45. Valletta, Malta (2010)
15. Abouenour, L., Bouzouba, K., Rosso, P.: An evaluated semantic query expansion and structure-based approach for enhancing Arabic question/answering. Int. J. Inf. Commun. Technol. **3**(3), 37–51 (2010)
16. Abdelnasser, H., et al.: Al-bayan: an Arabic question answering system for the holy Quran. In: Arabic Natural Language Processing Workshop, p. 57, Qatar (2014)
17. Pasha, A., et al.: MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of arabic. In: LREC, vol. 14, pp. 1094–1101 (2014)
18. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, vol. 41, p. 62, April 2009
19. Diab, M.: Second generation AMIRA tools for Arabic processing: fast and robust tokenization, POS tagging, and base phrase chunking. In: 2nd International Conference on Arabic Language Resources and Tools, vol. 110 (2009)

20. Elkateb, S., et al.: Arabic WordNet and the challenges of Arabic. In: Proceedings of Arabic NLP/MT Conference, London, UK (2006)
21. Fellbaum, C.: WordNet and wordnets. In: Brown, K., et al. (eds.) Encyclopedia of Language and Linguistics, 2nd edn, pp. 665–670. Elsevier, Oxford (2005)
22. Niles, I., Pease, A.: Mapping WordNet to the sumo ontology. In: Proceedings of the IEEE International Knowledge Engineering Conference, pp. 23–26 (2003)
23. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1, pp. 1–7. Association for Computational Linguistics (2002)
24. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, vol. 6, p. 12 (2007)
25. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. J. Artif. Intell. Res. **34**, 443–498 (2009)
26. Abdelali, A., Darwish, K., Durrani, N., Mubarak, H.: Farasa: a fast and furious segmenter for Arabic. In: NAACL-2016 (2016)