American Journal of Applied Sciences 8 (6): 652-661, 2011
ISSN 1546-9239

QArabPro: A Rule Based Question Answering System for Reading Comprehension Tests in Arabic

[1]Mohammed Akour, [1]Sameer Abufardeh,
[1]Kenneth Magel and [2]Qasem Al-Radaideh
[1]Department of Computer Science North Dakota,
State University Fargo, ND 58105
[2]Department of Computer Information System,
Yarmouk University Irbid, Jordan

Abstract: Problem statement: Extensive research efforts in the area of Natural Language Processing (NLP) were focused on developing reading comprehension Question Answering systems (QA) for Latin based languages such as, English, French and German. **Approach:** However, little effort was directed towards the development of such systems for bidirectional languages such as Arabic, Urdu and Farsi. In general, QA systems are more sophisticated and more complex than Search Engines (SE) because they seek a specific and somewhat exact answer to the query. **Results:** Existing Arabic QA system including the most recent described excluded one or both types of questions (How and Why) from their work because of the difficulty of handling these questions. In this study, we present a new approach and a new question-answering system (QArabPro) for reading comprehension texts in Arabic. The overall accuracy of our system is 84%. **Conclusion/Recommendations:** These results are promising compared to existing systems. Our system handles all types of questions including (How and why).

Key words: Arabic Q/A system, Information Retrieval (IR), Natural Language Processing (NLP), Arabic language, acronyms, Information Extraction (IE), morphological root, morphological analysis, QA systems, Stemming-root extraction

INTRODUCTION

The Arabic language is the fifth most spoken language in the world. It has approximately 280 million native speakers and about 250 million non-native speakers. It is also one of the six official languages of the United Nations (Arabic, Chinese, English, French, Russian and Spanish) (UN Department for General Assembly and Conference Management, 2008). In the last decade we witness the increasing growth of Arabic textual data on the web and the increasing demand for high-quality Arabic software. In order to meet these demands, more research and more investment in the development of systems that support Arabic language are necessary.

Natural Language Processing (NLP) concentrates on achieving natural language interoperability with the computer or programs. Natural languages are convenient and intuitive methods for accessing information (Katz *et al*., 2001; Salton and Buckley, 1988; Hirschman *et al*., 1999). The need for high quality systems capable of understanding and answering NL questions for Arabic language is paramount (Katz *et al*., 2001).

Today, there are well-established systems to assess information in natural languages such as English in specific and Latin based language in general. However, in the case of the Arabic language such systems are immature because of the unique aspects of the Arabic language (Abufardeh and Magel, 2008; Al-daimi and Abdel-amir, 1994); Habash and Rambow, 2005). These aspects include:

- Arabic is highly inflectional and derivational, which makes morphological analysis a very complex task
- The absence of diacritics (which represent most vowels) in the written text creates ambiguity and therefore, complex morphological rules are required to identify the tokens and parse the text
- There is no capitalization in Arabic. This complicates the process of identifying proper names, acronyms and abbreviations
- The writing direction is mostly mixed from right-to-left and from left-to-right. Furthermore, some characters change their shapes based on their location in a word

Corresponding Author: Mohammed Akour, Department of Computer Science North Dakota State University Fargo, ND 58105

The backbone of any natural language application is a lexicon. It is critical for parsing, text generation, text summarization and for question answering systems (Abuleil and Evens, 1998). Furthermore, Information Retrieval is one of the first areas of natural language processing in which statistics were successfully applied (Hiemstra and De vries, 2000). Two models of ranked retrieval methods were developed in the late 60s and early 70s and are still in use today: Salton vector space model (Polettini, 2004) and Robertson and Sparck Jones probabilistic model (Robertson and Sparck, 1976).

The rest of this study is structured as follows: Section two discusses related work. Section three describes a generic architecture for the new Arabic QA system. Section three explains the new approach. Section four discusses testing and evaluation results of the new system. Section five discusses to the results. Section six contains our conclusions and future work to further improve our QA system.

MATERIALS AND METHODS

Related work: Mohammed *et al*. (1993) introduced a QA system for Arabic called AQAS (1993). The AQAS system is knowledge-based and therefore, extracts answers from structured data and not from raw text (non structured text written in natural language); moreover, no results were reported. Hirschman *et al*. (1999) described an automated reading comprehension system for English text that can be applied to random stories and answers questions about them. Their work was executed on a corpus of 60 development and 60 test stories of 3rd and 6th grade materials. Each story was followed by a short answer test. The tests ask the students to read a story or article and then answer the questions about it to evaluate their understanding of the article, they used several metrics: precision, recall, HumSent (compiled by a human annotator who examined the texts and chose the sentence (s) that best answers the question) and AutSent (an automated routine that examines the texts and chooses the sentences that had the highest recall compared against the answer key). Humsent and AutSent compared the sentence chosen by the system to a list of acceptable answer sentences, scoring one point for a response on the list and zero points otherwise.

Rilo and Thelen, (2000) developed a rule based system called Quarc for English text that can read a short story and find the sentence presenting the best answer to a question. Each type of WH questions looks for different types of answers, so Quarc used a separate set of rules for each question type (e.g., WHO, WHAT, WHEN, WHERE, WHY). Each rule gave a certain number of points to a sentence. After applying all rules, the sentence that obtained the highest score was returned as the answer. All question types are similar in using a common WordMatch function, which counts the number of words that appear in both the question and the sentence being considered. Two words match if they share the same morphological root (Rilo and Thelen, 2000).

Following the Rilo approach, Hammo *et al*. (2002) introduced a rule-based QA system for Arabic text called QARAB. QARAB excludes two types of questions "كيف، ماذا"(How and Why) because" they require long and complex processing." The QARAB system did not report any data regarding precision or recall. The system was evaluated by the developers (four native Arabic speakers). They fed 113 questions to the system and evaluated the correctness of the answers. Such testing cannot be reliable and possibly is biased. In addition, such accuracy was not achieved for any other language using state-of-the-art QA systems.

Rotaru and Litman, (2005) worked on evaluating the process of combining the outputs of several question answering systems, to see whether they improved over the performance of any individual system. Their study included a variety of question answering systems on reading comprehension articles especially the systems described in (Hirschman *et al*., 1999; Rilo and Thelen, 2000). The training and testing data for the question answering projects came from two reading comprehension datasets available from the MITRE Corporation for research purposes. They concluded that none of those systems combined was globally optimal. The best performing system varied both across and within datasets and by question type.

Kanaan *et al*. (2009) described a new Arabic QA system (QAS) using techniques from IR and NLP to answer short questions in Arabic. Similar to QARAB, QAS excludes the questions "كيف، ماذا"(How and Why) citing the same reason cited by Hammo *et al*. Both stated that (How and Why) " require long and complex processing." Furthermore, the authors reported a test reference collection consisting of 25 documents gathered from the Internet, 12 queries (questions) and some relevant documents provided by the authors. Kanaan *et al*. reported different recall levels {0, 10 and 20%} where the interpolated precision was equal to 100% and at recall levels 90 and 100% it was equal to 43%. We should also note that, the study instructs the reader to see their result in a figure that is missing from the study.

In VSM a document is conceptually represented by a vector of keywords extracted from the document, with associated weights representing the importance of the keywords in the document and within the whole document collection. Similarly, a query is modeled as a list of keywords with associated weights representing

the importance of the keywords in the query (Salton *et al*., 1975; Salton and Buckley, 1988).

Term weighting is an important task in many areas of Information Retrieval (IR), including Question Answering (QA), Information Extraction (IE) and Text Categorization. The purpose of term weighting is to assign to each term w found in a collection of text documents a specific score s(w) that measures the importance with respect to a certain goal of the information represented by the word. For instance, passage retrieval systems weigh the words of a document in order to discover important portions of text and discard irrelevant ones. Other applications such as, QA, Snippet Extraction, Keyword Extraction and Automatic Summarization are used for the same purpose.

There are many term weighting approaches, including, IDF, TF.IDF, WIDF, ITF and log(1+TF). Term weighting techniques have been investigated heavily in the literature (Robertson and Sparck, 1976; Salton and Buckley, 1988; Rotaru and Litman, 2005). However, little consensus exists to conclude which weighting method is best. Different methods seem to work well for different goals (Kolda, 1997). Salton and Buckley (1988), confirmed that the most used document term weighting was obtained by the inner product operation of "the within document" term frequency and the Inverse Document Frequency (idf), all normalized by the length of the document. Salton and Buckley proposed (augmented normalized term frequency * idf) normalized by cosine as the best term weighting scheme. Further discussion follows in section 3.

Polettini (2004), analyzed and compared different techniques for term weighting and how much normalization improves retrieval of relevant documents. The presented two reasons that necessitate the use of normalization in term weights. According to Polettini (2004), the success or failure of the vector space method depends on term weighting. Term weighting plays an important role for the similarity measure that indicates the most relevant document.

The new QA system (QArabPro): The new QA system assumes that the answer is located in one document (i.e. it does not span through multiple documents). With this in hand the processing cycle of a QA system is composed of the following steps (Hammo *et al*., 2002):

- Process the input question and formulate the query
- Retrieve the candidate documents that contain answers using an IR system
- Process each candidate document in the same way as the question is processed and
- Return sentences that may contain the answer

The generic architecture of our QA system is shown in Fig. 1. It is composed of the following components:

- Question Analysis-Question classification and Query formulation
- IR system-Documents (passages) Retrieval
- NLP System-Answer Extraction

Question analysis-query reformulation: In general, question understanding requires deep semantic processing, which is a non-trivial task in NLP. Furthermore, Arabic NLP research at the semantic level is still immature (Hammo *et al*., 2002; Mohammed *et al*., 1993; Kanaan *et al*., 2009). Therefore, current Arabic QA systems do not attempt to understand the content of the question at the semantic level. Instead, they rely on shallow language understanding, i.e., the QA system uses keyword-based techniques to locate relevant passages and sentences from the retrieved documents (Hammo *et al*., 2002).

Stemming-root extraction: Conflating various forms of the same word to its root form, called stemming in IR jargon, is the most critical and the most difficult process especially for Arabic. The root is the primary lexical unit of a word, which carries the most significant aspects of semantic content and cannot be reduced into smaller constituents (Root and Stem, 2010). On the other hand a stem is the main part of a word to which prefixes and suffixes can be added and may not necessarily be a word itself. For example, the English word friendships contains the stem friend, to which the derivational suffix -ship is attached to form a new stem friendship, to which the inflectional suffix -*s* is attached (Root and Stem, 2010.). Another example where the stem may not be a word itself is "dod" as the stem in "doddle." The extracted roots are used for indexing purposes.

Several studies suggested that indexing Arabic text using roots significantly increases retrieval effectiveness over the use of words (Abuleil and Evens, 1998; Hammo *et al*., 2002; Habash and Rambow, 2005; Khoja, 1999; Al-Kharashi and Evens, 1994).

Stemming in Arabic language is difficult compared to English. The English language has very little inflection and hence a tendency to have words that are very close to their respective roots. The distinction between the word as a unit of speech and the root as a unit of meaning is more important in the case of languages where roots have many different forms when used in actual words, as is the case in Arabic.

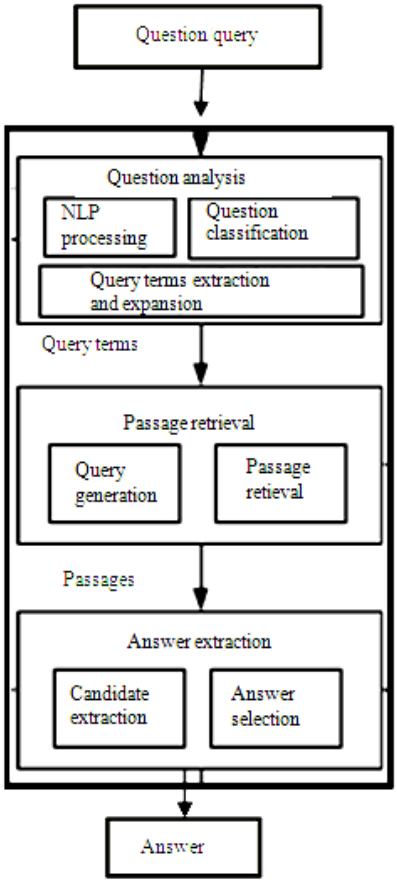

Fig. 1: The generic architecture of the QA system

While removing prefixes and suffixes in English may create problems, in Arabic both prefixes and suffixes are removed. The difficulty arises because Arabic has two genders, feminine and masculine; three numbers, singular, dual and plural; and three grammatical cases, nominative, genitive and accusative. A noun has the nominative case when it is a subject; accusative when it is the object of a verb; and genitive when it is the object of a preposition. The noun gender, number and grammatical case determine its form. (Abufardeh and Magel,(2008; Habash and Rambow, 2005; Al-Kharashi and Evens, 1994).

The stemming process for Arabic text involves the following steps to extract the root from Arabic words (Khoja, 1999):

- Removing or adding the definite article "ال" from the start
- Removing the conjunction letter "و "
- Removing or adding suffixes such as the letter "ون,ين,ان,ات,يت,ه,ة" from the end
- Removing or adding prefixes such as the special letter "ء"
- Pattern matching. Word pattern help in detecting the letters of the root of the word. For example The pattern of " يَكْتُبُونَ " is " يَفْعَلُونَ ", the letters ف،ع،ل " " replace the letters of root of " يَكْتُبُونَ " and the pattern of " نَالَ " is " فَعَلَ"

The IR system: Hammo *et al* (2002) used traditional text retrieval techniques as the basis for their QA system. The system is rule-based, but it relies mainly on indexing keywords. Then a key word matching strategy between the question and the document that contains the answer was used to identify the answer. Unfortunately, keywords or index terms alone cannot adequately capture the document contents, resulting in poor retrieval performance.

To implement our QA system we used an IR system to search and retrieve relevant documents. The IR system we constructed is based on Salton's statistical VSM (Salton *et al*., 1975). Furthermore, we used simple rules for each type of WH question as in Riloff *et al* QA system for English text (Rilo and Thelen, 2000). Rules for each WH question were applied to the candidate document that contains the answer. These rules were modified to accommodate the Arabic language requirements. This included the most difficult of all types of questions (How and Why).

The IR system can be constructed in many ways. Lundquist *et al*. (1999) proposed the construction of an IR system using a relational database management system (RDBMS). The IR system we constructed contains the following database relations:

- A root table: stores the distinct roots of the extracted term from documents. The stemmer performs root extraction
- Documents table: stores document information, such as document name, category name
- Verb table: to store verbs of words, such as يدرس, يقاتل
- Stopword table: contains Stopwords for the Arabic language such as: اذا, و, حيث
- Variations table: contains all different words of the same format in documents
- Document type root: contains root information in categories, such as the frequency

Document processing: This step is an essential step for any information retrieval technique. The step involves tokenization, stop word removal, root extraction and term weighting.

Term weighting: A document is typically treated as a bag of words (i.e., unordered words with frequencies). The bag is a set that allows duplicates of the same element. The assumption is that, more frequent terms in a document are more important (i.e. more pinpointing to the topic).

Salton's Vector Space Model (Salton *et al*., 1975) incorporates local and global information. The weight of each term is calculated using the following equation:

$$\text{Term Weight} = w_i = tf_i * \log (D/df_i) \quad (1)$$

Where:

Tfi = term frequency (term counts) or number of times a term i occurs in a document. This accounts for local information

Dfi = document frequency or number of documents containing term i

D = number of documents in the system

The dfi/D ratio is the probability of selecting a document containing a queried term from a collection of documents. This can be viewed as a global probability over the entire collection. Thus, the $\log(D/df_i)$ term is the inverse document frequency, IDF_i accounts for global information.

The VSM model is vulnerable to keyword spamming; an adversarial technique in which terms are intentionally repeated to improve the position of a document in the Search Engine (SE) ranking results. Therefore, terms with high occurrences are assigned more weight than term repeated few times and ranking and retrieval is compromised. To make the mode less susceptible to keyword spamming document and query frequencies are normalized.

The normalized frequency of a term *i* in document *j* is given by:

$$f_{i,j} = tf_{i,j} / \max tf_{i,j} \quad (2)$$

Where:

f_i = Normalized frequency

$tf_{i,j}$ = Frequency of term i in document j

$\max tf_{i,j}$ = Maximum frequency of term i in document j

The similarity between a document vector d_k and a query vector q in VSM is calculated using the following equation (Baeza-Yates and Ribeiro-Neto, 1999):

$$\text{CosSim}(dj, q) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

Where:

dj = The vector for document

j, q = The query

w_{ij} = The weight of the term i in the document j

$w_{i,q}$ = The weight of the term i in the query q

NLP system-question processing: Q/A systems rely on NLP tools that perform linguistic analysis on both the question and the document. The system treats the incoming question as a "bag of words" against which the index file is searched to obtain a list of ranked documents that possibly contain the answer.

NLP starts by assigning to each word from the question its root and the proper Part-of-Speech (POS) and then stores them in the database. The NLP system contains the following modules:

- Tokenizer module: this module is used to extract the words (tokens) from the query and the documents
- Tagging module (or type-finder module): The main function of this module is to perform grammatical tagging (or part-of-speech tagging); the process assigning to each word of a sentence a tag which indicates the function of that word in that specific context. POS can be Verb, Noun, Proper Noun, Adjective, Adverb. The Tagger generally used to construct an Arabic lexicon. *Lexicon* is a collection of representations for words used by a linguistic processor as a source of word specific information; this representation contains information on the morphology, phonology, syntactic argument structure and semantics of the word (Habash and Rambow, 2005)
- Feature-Finder module: this module is responsible for determining the features of the word (gender, number, person, tense)
- Named Entity Recognition (NER) module: This module is used to extract proper nouns as well as temporal and numeric expressions from raw text. Named entities are phrases that contain proper names. Named Entities are categorized into one of the following categories: Person, Organization, Location, Date, Time, Percentage and Monetary amount

Query Expansion (QE): Like SE and IR systems, Q/A systems are generally constructed using three major modules: Question classification and analysis, document (or passage) retrieval and answer extraction. The performance of the latter is dependent on the performance of the first two modules. This is true because, the query is a simple question in natural

language and users mostly formulate questions using words that might not appear in the base document. Therefore, if the retrieved passage does not contain all or part of the question keywords, the question extraction module will not provide the expected answer. To avoid such problems, a QE process is used to generate new keywords that may exist in the base document which in turn improve the performance of the whole system.

The user query can be extended by adding new words deemed to be somehow (usually semantically) connected to those contained in the initial query. The QE module generally use a dictionary of synonyms, a thesaurus, an Ontology, or an index file storing words with similar roots. Because Arabic is highly inflectional and derivational, morphological analysis is a very complex task. Therefore, we need to consider the most important derivation in query reformulation to find all related words in a document. For example, "جمع المثنى المذكر السالم,جمع المؤنث السالم" and the definite article ال which is sometimes not added at the beginning of some words. QE module in our system uses a small dictionary of synonyms. For more details about QE and Query correction techniques please see (Rachidi *et al*., 2003; Abdelali, *et al*., 2003).

Query type: Questions are classified based on a set of known "question types". Question types are instrumental in determining the type of processing needed to identify and extract the final answer. Table 1 shows the main interrogative particles that precede the questions to determine what types of answers are expected. While previous Q/A system excluded questions types that are difficult to handle, our QA system handles all question types listed in Table 1.

Rule based Q/A systems: A rule-based system for question answering is a system that looks for evidence that a sentence contains the answer to a question (Rilo and Thelen, 2000). Each type of questions looks for different types of answers, so a question answering system uses a separate set of rules for each question type such as (من, ما, ماذا, متى).

The rules we adapted in our system are generally similar to those used in a rule-based QA for English text (Hirschman *et al*., 1999). However, the rules are modified and enhanced to accommodate the many unique aspects of the Arabic text. These modifications are very critical to the process of determining the correct answer. Each rule awards a certain number of points to a sentence. After applying the rules, the sentence with the highest score is marked as the answer. All question types share a common word matching function that counts the number of words that appear in both the question and the sentence under consideration.

The word match function first removes stopwords such as: { حيث, و, اذا}from a sentence and then matches the remaining words against the words in the typical question. Two words match if they share the same morphological root. Verbs are very important in determining when a question and a sentence are related, verb matches are weighted more heavily than non verb matches. Matching verbs are awarded (4) points each and other matching words are awarded (2) points each. The remaining rules used by question answering system look for a variety of clues. Lexical clues look for specific words or phrases. Unless a rule indicates otherwise, words are compared using their morphological roots. Some rules can be satisfied by the lexical items. These rules are written using the set notation (e.g., {أمس, اليوم, غدا}).

Each rule awards a specific number of points to a sentence, depending on how strongly the rule believes that it found the answer. A rule can assign four possible levels of points: clue (+3), good clue (+4), confident (+6) and slam dunk (+20). The main purpose of these values is to assess the relative importance of each clue.

Figure 2 shows the rules for (Who/Whose) "من", which use three fairly general heuristics as well as the Word Match function (rule #1). If the question (Q) does not contain any names, then rules #2 and #3 assume that the question is looking for a name. Rule #2 rewards sentences that contain a recognized NAME and rule #3 rewards sentences that contain the word "name". Rule #4 awards points to all sentences that contain either a name or a reference to a human (often an occupation, such as "writer" "كاتب"). Note that more than one rule can be applied to a sentence, in which case the sentence is awarded points by all of the rules that are applied.

The (What/Which) "ما" questions were among the most difficult to handle because they sought amazing wide variety of answers. However, Fig. 3 shows a few specific rules that worked reasonably well. Rule #1 is the generic word matching function shared by all question types. Rule #2 rewards sentences that contain a date expression if the question contains a month of the year. This rule handles questions that ask " ماذا حدث" on a specific date. We also noticed several "ما نوع" questions that looks for a description of an object. Rule #3 addresses these questions by rewarding sentences that contain the word (e.g., "يسمى..." or "...مصنوع من"). Rule #4 looks for words associated with names in both the question and sentence.

The rule set for (When) "متى" questions shown in Fig. 4, is the only rule set that does not apply the word match function to every sentence in the text.

1. Score(S) += Word Match (Q, S)
2. If~ contains (Q, NAME) and Contains(S, NAME)Then Score(S) += confident
3. If ~ contains (Q, NAME) and Contains(S, name) Then Score(S) += good clue
4. If contains(S, {NAME, HUMAN}) Then Score(S) += good_clue.

Fig. 2: Rules for "من" (Who/Whose)

1. Score(S) += Word Match (Q, S)
2. If contains (Q, MONTH) and Contains(S, {الليلة السابقة,البارحةالليلة الماضية,غدا,امس ,اليوم})Then Score(S) += clue
3. If contains (Q, نوع) andContains (S, {يدعى,يطلق,من}) Then Score(S) += good_due
4. If contains (Q, اسم) and Contains(S, {اسم,يدعى,يطلق,عرف})Then Score += slam_dunk

Fig. 3: Rules for "ما" (What/Which)

If contains(S,TIME) Then Score(S) += good_clue.
Score(S) += Word Match (Q, S)
2. If contains (Q, ماض) and Contains(S, {الاول, الاخر, مضى, منذ}) Then Score(S) += slam_dunk
3. If contains (Q, {بدا,انطلق }) and Contains(S, {سنة, بدا,منذ,عام})
Then Score(S) += slam_dunk

Fig. 4: Rules for "متى" (When)

1. Score(S) += Word Match (Q, S)
2. If contains(S, LocationPrep)
Then Score(S) += good_clue
3. If contains(S, LOCATION)
Then Score(S) += confident

Fig. 5: Rules for "اين" (Where)

1. If SeBEST Then Score(S) += clue
2. If S immed, precedes member of BEST Then Score(S) += clue
3. If S immed follows member of BEST Then Score(S) += good_clue
4. If contains(S, يريد, يتطلب)
Then Score(S) += good_clue
5. If contains(S, {لغاية, لهذا,لأنه ,لذلك,بسبب})
Then Score(S) += good_clue

Fig. 6: Rules for "لماذا" (Why)

"متى" questions usually require a TIME expression, so sentences that do not contain a TIME expressions are only considered in special cases. Rule #1 reward all sentences that contain a TIME expression with a good_clue points as well as Word Match points. The remaining rules look for specific words that suggest duration of time. Rule #3 is interesting because it recognizes that a certain verb ("يبدأ",) can be an indicative of time even "متى" no specific time is mentioned.

The (Where) "اين" questions usually look for a specific place or location, so the "اين" rules are much focused. In Fig. 5, rule #1 applies the general word matching function and Rule #2 looks for sentences with a location preposition. Our Question answering system recognizes a number of prepositions as being associated with locations, such as "في", "الى", "عن" and "الى". Rule #3 looks for sentences that contain a word belonging to the LOCATION semantic class.

The (Why) "لماذا" questions are one of the most difficult and are handled differently from all other question types. The "لماذا" rules are based on the observation that the answer to a "لماذا" question often appears immediately before or immediately after the sentence that most closely matches the question. We believe that this is due to the causal nature of "لماذا" questions. First, all sentences are assigned a score using the word match function. Then the sentences with the top score are isolated. We will refer to these sentences as BEST. Every sentence score is then reinitialized to zero and the "لماذا" rules shown in Fig. 6 are applied to every sentence in the text.

Rule # 1 rewards all sentences that produced the best word match score because they are plausible candidates. Rule # 2 rewards sentences that immediately precede a best word match sentence and Rule # 3 rewards sentences that immediately follow a best word match sentence. Rule # 3 gives a higher score than Rules # 1 and # 2 because we observed that WHY answers are somewhat more likely to follow the best word match sentence. Finally, Rule # 4 rewards sentences that contain the word "يريد". Rule # 5 rewards sentences that contain the word "لذلك"or "بسبب". These words are indicative of intentions, explanations and justifications.

In English, a question that starts with "كم" (How many/much) is usually followed by a noun *X* where the question's target is the amount of *X*. "How many" is used for countable nouns such as days or cars. while "How much" is used for uncountable nouns such as coffee or milk. In Arabic, there is only one word "كم" which is used to express both (How many/much).

```
1-Score(S) += Word Match (Q, S)
2. If contains (Q,criteria,) and Contains(S,
{Number}) Then Score(S) += slam_dunk
3. If contains (Q, criteria) and
Contains(S, { تقدر, يقارب, تحت,فوق,أكثر,
أقل,تقريبا,حوالي})
Then Score(S) += confident
4- If contains (Q, criteria) and
Contains(S, { يبلغ,يساوي,تضم,عدد})
Then Score(S) += confident
```

Fig. 7: Rules for "كم" (How many/much)

Table 1: Question types processed by the question answering system

Question word	Query type
Who / Whose "**من**"	Person
When / "**متى**"	Date, Time
What/Which "**ماذا**/"**ما**"	Organization, Product, Event
Why / "**لماذا**"	Reason
Where / "**اين**"	Location
How many/much "**كم**"	Number/Quantity

The nouns after "كم" must be always singular and in the accusative case. If the noun following "كم" were part of a genitive construction, it would not be in the accusative case but in the regular nominative case. Furthermore, the noun following "كم" can be omitted from the question. When asking about price, "كم" will be preceded by either one of the following preposition "in/by/with" and the noun is generally omitted. Furthermore, "كم" can be used in a style that is used to state numerousness instead of interrogation. These are a few of the many cases that govern this type of question. There are many more.

Figure 7 shows a few specific rules that worked very well with this type of question. Rule #1 is the generic word matching function shared by all question types. Rule #2 rewards sentences that contain numbers as the answer of question containing one or more measurement criteria such as (Approximation) "نسبة". It handles questions that ask for countable/uncountable name and the predicted answer contains an exact number such as "عشرة". Rule #3 addresses such questions by rewarding sentences that contain words such as ("أقل ,تقريبا,حوالي") and it handles questions that ask for countable/uncountable name and the predicted answer contains approximation rather than an exact number. Rule #4 addresses such questions by rewarding sentences that contain words such as ("يساوي,تضم,عدد") and it handles questions that ask for countable/uncountable name and the predicted answer contains an exact number.

RESULTS

We tested our system using a collection of reading comprehension texts. The data used were collected from WIKIPEDIA (Root and Stem, 2010). The data set contains 75 reading comprehension tests with 335 questions. The HumSent answers are sentences that a human expert judged to be the best answer for each question. The AutSent answers are generated automatically by determining which sentence contains the highest weight, excluding stopwords. Our parser uses a small dictionary, so that words can be defined with semantic classes. The semantic classes used by our system along with a description of the words assigned to each class are the following:

- Human: 52 words, including titles such as دكتور,سيد,خليفة
- Location: 135 words, including country names and city names such as عمان, مكة, المدينة.
- Names: 621 words, including common first name, last name such as محمد, علي, محمود
- Times: 42 words, including years, 12 month and 7 days such as الجمعة, السبت, شباط
- Stopwords: 1457 words, including words that don't have meaning on their own such as مما, اذا, اكثر اقل
- Criteria: 34 words, which are enumerated some measurement criteria for countable and uncountable names such as تقدر, يقارب, تحت,فوق, أكثر

Table 2 shows the evaluation results of our QA system for each type of questions.

Figure 8 shows a summary of each question type and its corresponding accuracy. The system achieved 84% overall accuracy. The system performed the best on (Who/Whose) "من" WHER"أين" and (What/Which) "ما" questions and performed the worst on WHY "لماذا" and "كم" (How many/much) questions, reaching only 62% and 69% accuracy respectively. The low results for WHY "لماذا" and "كم" (How many/much) questions were expected because of the difficulty in handling such questions. These questions were completely excluded by QA systems introduced by (Hammo *et al*., 2002; Kanaan *et al*., 2009).

Table 2: Overall Results

Total # of questions	Correct percent	Incorrect answer	Correct answer	Question type
53	94.34%	3	50	**من**
48	91.67%	4	44	**ما**
45	88.89%	5	40	**ماذا**
47	93.62%	3	44	**أين**
54	85.19%	8	46	**متى**
45	62.22%	17	28	**لماذا**
43	69.77%	13	30	**كم**
335	84.18%	53	282	overall

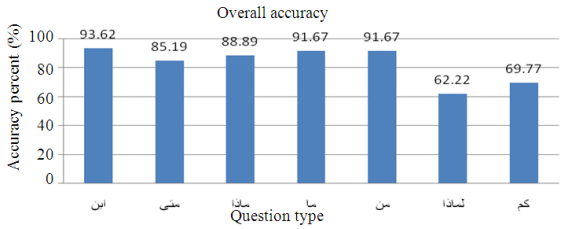

Fig. 8: The overall accuracy

DISCUSSION

Handling "لماذا" (Why) questions was the most difficult because this type of question is usually concerned with causal information and requires deep semantic processing, which is a non-trivial task in NLP. The system has to find more keywords that are useful in identifying intentions, explanations and justifications. In general, a better understanding of causal relationships and deeper semantic processing would increase the accuracy of answers to this type of question.

Handling the "كم" (How many/much), questions was also difficult because of the many rules "كم" (How many/much) required. At this point, the rules we used are considered simple and generic. More work is needed to cover all cases for this type of question.

The performance of the IR system is strongly dependent on correct processing of the query. The IR achieved an overall Precision (P) of 93%, a Recall (R) of 86% and an F-Measure of 89%. There is a trade-off between precision and recall. Greater precision decreases recall and greater recall leads to decreased precision. The F-measure is the harmonic-mean of *P* and *R* and takes account of both measures.

A major lesson learned was the importance of utilizing high Arabic experts in formulating the heuristics/ rules to accommodate the many unique aspects of the Arabic text and increase the performance of the process of determining the correct answer.

CONCLUSION

In this study, we introduced a new QA system (QArabPro) for Arabic. The system achieved 84% overall accuracy on our test set. These results are promising compared to existing systems. Existing Arabic QA systems excluded one or both types of questions (How and Why) from their work because of the difficulty of handling such types of question. Our system handles all types of questions including (How and Why). While the overall accuracy for these two types of questions is low, {62% for "لماذا" (Why) and 69% for "كم" (How many/much)} compared to other types of questions we consider this an important milestone and an improvement to current Arabic QA system.

Query expansion and relevant keywords extraction both require a robust Named Entity Recognition (NER) module. NER is an integral part of any language lexicon. We expect that improving an automatic Arabic lexicon with techniques that acquire semantic knowledge automatically will improve the performance of the system in general. More specifically, it will improve the performance of the system when dealing with How and Why questions.

ACKNOWLEDGMENT

The researchers would like to thank Professor Tariq King from North Dakota State University for his helpful comments and suggestions that reflected many improvements in the presentation of this study.

REFERENCES

Abdelali, A., J. Cowie, D. Farwell, W. Ogden and S. Helmreich, 2003. Cross-language information retrieval using ontology. Proceedings of the TALN '2003, Batz-sur-Mer, France.

Abufardeh, S. and K. Magel, 2008. Software localization: The Challenging Aspects of Arabic to the Localization Process (Arabization). IASTED Proceeding of the Software Engineering SE 2008, Innsbruck, Austria, pp: 275-279.

Abuleil, S., M. Evens, 1998. Discovering lexical information bytagging arabic newsstudy text. Proceeding of the Workshop on Semantic Language Processing Coling-ACLl.

Al-daimi, K., M. Abdel-amir, 1994. The syntactic analysis of arabic by machine. Comput. Humanities, 28: 29-37.

Al-Kharashi I.A. and M.W. Evens, 1994. Comparing words, stems and roots as index terms in an Arabic information retrieval. *JASIS*, pp: 548-560.

Baeza-Yates, R. and B. Ribeiro-Neto, 1999. Modern Information Retrieval. Addison Wesley, Reading, MA, USA.

Habash, N. and O. Rambow, 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp:573-580, June 25-30, Ann Arbor, Michigan.

Hammo, B., H. Abu-Salem and S. Lytinen, 2002. QARAB: A Question Answering System to Support the Arabic Language. Annual Meeting of the ACL Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic pp: 1-11.

Hiemstra, D. and A. De Vries, 2000. Relating the new language models of information retrieval to the traditional retrieval models. Technical Report tr-ctit-00-09, Centre for Telematics and Information Technology.

Hirschman, L., M. Light, E. Breck and j. Burger, 1999. Deep read: A reading comprehension system. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics.

Kanaan, G., A. Hammouri, R. Al- Shalabi and M. Swalha, 2009. A new question answering system for the Arabic language. American J Applied Sci., 6: 797-805, ISSN 1546-9239.

Katz B., J. Lin and S. Felshin, 2001. Gathering knowledge for a question answering system from heterogeneous information sources. Proceedings of the Workshop on Human Language Technology, ACL-2001, Toulouse.

Khoja, S. 1999. "Stemming Arabic Text". Available on the Web at: http://zeus.cs.pacificu.edu/shereen/research.htm#stemming.

Kolda, T., 1997. Limitedmemory matrix methods with applications. Applied Mathematics Program, PHD Thesis, University of Maryland at College Park, 59-68.

Lundquist, C., D. Grossman and O. Frieder, 1999. Improving relevance feedback in the vector space model. Proceedings of the 6th ACM Annual Conference on Information and Knowledge Management (CIKM), pp: 16-23.

Mohammed, F.A., K. Nasser and H.M. Harb, 1993. A knowledge-based Arabic Question Answering System (AQAS). In: ACM SIGART Bulletin, pp: 21-33.

Polettini, N., 2004. The vector space model in information retrieval - term weighting problem. University of Trento: http://sra.itc.it/people/polettini/STUDYS/Polettini_Information_Retrieval.pdf. (Accessed, Aug. 2009).

Rachidi T., M. Bouzoubaa, L. ElMortaji, B. Boussouab and A. Bensaid. 2003. Arabic user search Query correction and expansion", in Proc. Of COPSTIC'03, Rebat Dec. 11-13.

Rilo, E. and M. Thelen, 2000. A rule-based question answering system for reading comprehension tests. In Proceedings of the Anlp/Naacl 2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-based Language Understanding Systems.

Robertson, S.E. and K. Sparck, 1976. Relevance weighting of search terms. J. Am. Soc. Inform. Sci., 27: 129-146.

Root and Stem, 2010 (linguistics) http:// ar.wikipedia.com.

Rotaru M., D. Litman, 2005. Improving question answering for reading comprehension tests by combining multiple systems. In Proceedings of the American Association for Artificial Intelligence (AAAI) 2005 Workshop on Question Answering in Restricted Domains.

Salton G. and C. Buckley, 1988. Term weighting approaches in automatic text retrieval. Inform. Process. Manage.

Salton, G., A. Wong and C.S. Yang, 1975. A vector space model for automatic indexing. Commun. ACM., 18: 613-620.

UN Department for General Assembly and Conference Management, 2008. http://www.un.org/Depts/DGACM/faq_languages.htm