Handling “why” questions in Arabic

Aqil M. Azmi  
Department of Computer Science  
King Saud University  
Riyadh, Saudi Arabia  
aqil@ksu.edu.sa

Nouf AlShenaifi  
Department of Computer Science  
King Saud University  
Riyadh, Saudi Arabia  
nalshenaifi@yahoo.com

Abstract—Little effort were directed towards the development of Question Answering systems for Arabic, compared to other languages. Moreover, the existing Arabic Question Answering systems ignore why-questions because of the difficulty of handling it. In this paper, we propose an approach for automatically answering why-questions for Arabic texts. Our proposed system has four components: (1) Question analysis component that applies NLP techniques to preprocess input question and formulate a query; (2) Document preprocessing that process the documents in the same way as the why-question; (3) Use information retrieval toolkit, Lemur that retrieves a list of candidate passages possibly containing the answer; and (4) Answer extraction. The novelty of our system is in the answer extraction component which is based Rhetorical Structure Theory (RST) method. The overall success of the system is encouraging compared to existing ones.

Keywords—Arabic question-answer; Arabic NLP; Information retrieval; RST.

I. INTRODUCTION

The dramatic increase of Arabic resources on the web lead to developing Question Answering systems (QAS). Question Answering (QA) is a field of Natural Language Processing (NLP) that automatically provides an answer for a question posed by a human in natural language. The Question Answering task for the Arabic language has been lightly investigated in the last decade. The research in this area for Arabic is behind compared to other languages.

Question Answering as a problem deals with many types of questions. Factoid questions are one type that expect a short, clearly identifiable answer; typically a question that asks about some Named Entity, e.g. a person’s name, date/time, location ... etc. The second type is the questions that use the words “Why” or “How” that is harder to answer. Most of the attention in QA research has been geared towards factoid questions. Existing Arabic QAS including the most recent ones excludes “How” and “Why” form questions. This is because of the difficulty of handling these questions. Why-questions require a different approach than factoid questions because their answers tend to be longer and more complex. Fig. 1. illustrates a sample Arabic text with several question types: what, how much, and why. This figure serves to show the difficulty of answering why-questions. In this work, we specifically focus on the problem of answering why-questions for the Arabic language.

II. RELATED WORKS

The daily influx of new data on the web has forced researchers to focus on Question Answering Systems (QAS) [1]. These systems provide users with a flexible access to information while allowing them a query expressed in natural language; which returns a concise answer itself rather than some related documents. In order to handle this problem, traditional information retrieval techniques were joined with a sophisticated natural language processing approach [6][10].

Arabic is the sixth most important language in the world with more than 300 million speakers. Arabic QAS have gained great importance due to the increasing amounts of Arabic content on the Internet and the increase demand for information that regular Information Retrieval (IR) techniques cannot satisfy [13]. However, Arabic QAS has not matured in par with other languages such as English, mainly due to the

This work was funded by Research Center of the College of Computer & Information Sciences (CCIS) at King Saud University under grant number RC-140202 for which the authors are thankful.
challenges the language itself pose. Arabic is highly inflectional and derivational which makes morphological analysis a very complex task. The absence of diacritics (which represent most vowels) in Modern Standard Arabic (MSA) creates ambiguity in Question Analysis and Answer Extraction and therefore, complex morphological rules are required to identify the tokens and parse the text. Unlike English and other Latin-based languages, there is no capitalization in Arabic which makes Named Entity Recognition (NER) a lot harder [2][13].

For English there are few works that deal with why-Question in the area of QA. In her PhD dissertation, Verberne [15] explained a method for why-question answering that is based on discourse structure and relations in a pre-annotated document collection (the RST Treebank). Answers to why-questions can be extracted by matching question topic to a span in the RST tree and selecting the most relevant answer according to the RST (Rhetorical Structure Theory) relation that holds between a question topic and its answer spans. In [16] the authors described an approach for ranking answers to why-questions by evaluating a number of machine learning techniques in their performance.

Few efforts were conducted to build QAS oriented for the Arabic language. Hammo et al. [5] described an approach to QAS system that provides short answers to natural language questions expressed in the Arabic language. The proposed system called QARAB. The system’s primary source of knowledge is a collection of text extracted from Al-Raya, a newspaper printed in Qatar. It is a factual QA system that uses both information retrieval to identify candidate passages, and natural language processing techniques to parse the question. QARAB does not handle questions of type how and why since they require long and complex processing.

Kanaan et al [6] described QAS for the Arabic language. The system uses grammar and morph syntactic patterns rather than complicated linguistic analyses for questions and candidate answers. According to the authors, this system cannot handle how or why questions because of the complex processing involved in handling such questions.

Akour et al [2] introduced QArabPro for reading comprehension texts in Arabic language. This system handles all types of questions including how and why. The overall accuracy for these two types of questions is low, 62% for why and 69% for how.

The authors in [14] proposed DefArabicQA which identifies and extracts the answers (i.e., exact definitions) from Web resources based on a pattern approach. It was a first research that focused on definitional QA systems for the Arabic language.

Brini et al [4] proposed QASAL, a system for factoid and definition questions. It employs the NooJ platform which represents a valuable linguistic development environment. QASAL takes advantage of some linguistic techniques from IR and NLP to process a collection of Arabic documents and the Arabic version of the search engine Google. It is worth noting that definition questions are different from why-questions. QASAL, for example, only handles definition questions pertaining to profile of a person, full name of an organization, or a concept. Salem et al [12] presented the first study for automatically finding answers to why and how-to-questions for Arabic language based on Rhetorical Structure Theory (RST). The system proposed by [12] has two problems. Their system has low performance; the authors reported recall of 55%. Also, the authors did not use a standard corpus; rather they relied on Arabic websites.

III. RHETORICAL STRUCTURE THEORY

RST was mainly developed by W. Mann and S. Thompson [9] in late 1980s. The central constructs in RST are rhetorical relations. The coherence of the text is attributed mostly to the presence of these relations. The claim is that the relations in RST suffice to analyze most of the English texts, except for some unusual documents, e.g. poems. The rhetorical relations are defined in terms of the effect the writer intends to attain by presenting two text spans side by side.

While representing the relationship between two text spans, rhetorical relations also convey information about which span is more significant to writer’s purposes. The nucleus is the more central span, while the satellite is the less important one. According to Mann and Thompson the majority of the text is structured using nucleus-satellite relations. RST provides a set of 23 rhetorical relations, among these relations are: elaboration, circumstance, condition, interpretation, restatement, summary, sequence, motivation, background, evidence, justify, concession, … etc.

In RST, the clause is selected as the minimal unit of organization; and so text spans are clauses, or larger units made up of clauses. The relations in RST are fitted onto structures called schema applications, and these are fitted into text. The rhetorical structure tree (RS-tree) is a hierarchical system of schema applications. The schema application links a number of consecutive spans, and creates a complex span which can in turn be linked by a higher level schema application. A central claim of RST is that the structure of any coherent discourse can be described by a single RS-tree.

IV. OUR PROPOSED ALGORITHM

Unlike factoid questions, why-question answering is an interesting and challenging problem which involves a linguistically motivated approach for reasoning in natural language to information retrieval. Following the works of Verberne [15], and Salem et al [12] for handling why-questions in English and Arabic respectively, we intend to use the same underlining mechanism, namely RST. It is one of the leading theories in computational linguistics and has been applied successfully in a number of NLP linguistic applications, e.g. Arabic text summarization [3]. As the work of Salem et al [12] indicates, RST alone is insufficient and we do need to incorporate some other schemes as well, since the results reported in [12] were not that good (reported recall of 55%). For testing our system we use OSAC (Open Source Arabic Corpora) [11]. This is a huge corpus with about 4700 articles from BBC Arabic, and another 5000 from CNN Arabic.
Fig. 2. An overview of our Arabic why-question answering system.

Fig. 2. is an overview of our proposed why-question answering system for the Arabic language. The system accepts as input the why-question expressed in Arabic. It will output a paragraph that contains the most efficient and appropriate answer. In general there are four components to the system,

- Question analysis.
- Document preprocessing.
- Document/passage retrieval.
- Answer extraction.

In the first component we employ NLP techniques to parse and process the input question in order to obtain some useful information, e.g. question keywords, and also to formulate the query. In a similar way we process the documents in the second component. The third component retrieves a list of candidate passages that possibly holds the answer(s). Finally the system returns the answer to the user. This is the task performed by the fourth component, answer extraction. In designing our system we assume that the answer always exist in our corpus. Next we go over each component in more detail.

Do question tokenization.
Normalization.
Remove stop words.
Stemming – root extraction.
Query formulation and generation.
Query expansion.

Algorithm 1. The six steps to do the question analysis, the first of four components that make up our Arabic why-question answering system.

The question analysis is a six step process (Algorithm 1). For stemming we use Khoja’s stemmer [7] which returns the root of the word. The normalization step includes unifying all forms of the letter alif, (ا → ا), (ة → ﺛ) and remove all the diacritical marking. For the last step we use Arabic WordNet (AWN) ontology to add new words, e.g. synonyms that semantically connect to those contained in the query.

In the second component we process the document and it is processed in the same way as the query in the first component. Namely we do, tokenization, normalization, stop word removal and root extraction.

The third component, passage retrieval is the core module in any QA system. We use an off the shelf retrieval module, Lemur [8] for processing the data set and retrieve a list of candidate documents. The system returns the documents ranked in some order, we select high ranked document relevant to user’s query and pass it on to the final component for further processing. The reason for using Lemur is that it has built-in support for text documents written in Arabic. Another factor is that Lemur supports passage retrieval rather than retrieving the full document.

The input to the fourth component is the “bag of words” of why-question and a list of ranked relevant documents. Here we use RST which provides a framework for describing text and rhetorical relations among parts of a text. The text is divided into many spans of arbitrary size. The span can be either nucleus or a satellite. Next we identify the rhetorical relations that hold between text spans. By identifying relations between spans of text, a full rhetorical structure representation called schema or RS-tree is created. Of our interest are five rhetorical relations: interpretation (تفصيل), justification (توضيح), result (نتيجة), explanation (تفصيل) and base (قاعدة). To extract the answers using RST we divide candidate document into units based on cue-phrases then indicate the RST relation that holds between these two units. The answer is picked based on the priority of the RST relation. Suppose we issue a query and the pulled document provides two different RST relations, one is justification and the other is result. We pick justification as it has a higher priority.

V. IMPLEMENTATION, EVALUATION AND RESULTS

We implemented the system using Java programming language. To test the system we used a collection of Arabic text extracted from Open Source Arabic Corpora (OSAC) [7]. There were 700 documents in our data set.

Fig. 3. is our index builder. This application provides a GUI interface for building indices of our corpus. Our main application’s GUI (Fig. 4.). The input is a why question and the answer is shown beneath it.

For our experiment, the system was evaluated by native Arabic speakers who formulated a series of why questions for which the answer could be found in the data set. We collected a set of why questions along with the correct answer. The answers were manually generated. The collected questions were passed to the system for an answer. To judge the correctness of the system retrieved answers, it was compared with those manually generated by the humans. If both answers match, we mark it as correct answer. Since the assessment involved human labor, we limited the evaluation to 100 why questions. Out of 100 questions in our experiment text, the system was able to get 71 of them correct. This means our system achieved an accuracy of 71%.

To measure the performance of the system we used, recall and precision. Recall is the number of correct answers over the number of questions to be answered, while precision is the number of correct answers over the number of questions.
answered. For our system the measures were: recall = 71/100 = 71%, and precision = 71/91 = 78%. These results are better than the one reported by [12].

VI. CONCLUSION AND FUTURE WORK

Most of the Questions/Answering systems in Arabic ignore why-questions since these are more challenging to deal with. In this paper we described a system to answer Arabic why-questions. Our approach is based on rhetorical structure theory (RST), a leading theories in computational linguistics. For evaluation we used the open domain corpus OSAC. We picked 700 documents out of this corpus and collected a set of 100 why-questions posed by humans whose answer is guaranteed to be in the data set. The result of the evaluation was: recall 71% and precision 78%.

For future work we plan to increase the number of questions in the test and use a bigger corpus. Another improvement is to let the system consult the web in case it fails to find an answer in the data set to a posted why question.

REFERENCES


