# Recent Advances and Challenges of Arabic Why Question Answering Systems

Walaa Saber Ismail
Emirates College of Technology, Abu Dhabi, United Arab Emirates, walaa.saber@ect.ac.ae

Masun Nabhan Homsi
Helmholtz Centre for Environmental Research (UFZ), Leipzig, Germany, masun.homsi@ufz.

**ABSTRACT**

Arabic Question Answering Systems have gaining a remarkable significance through the increasing amount of the Arabic contents in the Internet and the growing demand for precise answers which cannot be offered by the regular information retrieval techniques. As the amount of research in Arabic Question Answering system is behind when it is compared to other languages, and handling non-factoid QASs is the not-trivial task in Natural Language Processing (NLP), it's not surprising that few researches built Arabic why question answering (whyQA) systems. This paper addresses the main challenges and gaps analysis for Arabic whyQA systems, and some future trends have been mentioned for the guidance of the new research in that area.

**Keywords***:* WhyQA System Architectures; Information Retrieval; Semantic Web; Deep Leaning; Non-Factoid Questions.

## 1 INTRODUCTION

Arabic question answering systems (AQASs) become very important after the huge amount of Arabic resources that are available on the web. The main target of QAS is providing precise and accurate answer to the user question that are written in natural languages rather than text files. In general, the question answering as a problem deals with the two types of questions; the first type is the factoid questions that mainly asks about a named entity as a person, date/time, place or location. The second type is the non-factoid (Causal) questions that is harder to answer and using the words "Why "or "How" [1].

Non factoid questions more complex and require special techniques to handle compared to factoid questions which expects a short identifiable answer. Answering why-questions is not so an easy task as it requires deep semantic processing, and the system might need the analyzation of several documents, the extraction of multiple passages, and the combination of them to present the answer [2]. As Arabic language is a syntactically rich language and the computer-based processing of Arabic documents is not an easy task, many challenges are added to Arabic whyQA systems. Due to limitations in Arabic language and the rareness of linguistic resources, building Arabic question answering systems is as a big challenge. For causal questions, the explanation about an entity is required. Based on the interpretation a same question can have different answer. The answer to why question is subjective that can differ from a sentence to a paragraph or to a whole document. In order to generate answers in why questions, the identifications of discourse relationship are required in the source document. Most of the existing why QAS based on words model have a problem in the retrieval process due to synonymy, polysemy, and homonymy [3].

As mentioned earlier, little effort was directed towards Arabic question answering systems that have tried to handle the non-factoid questions. In order to fill this gap and guide the new researchers in that field, we have to review the state of art in handling why-questions for the Arabic languages. To best of knowledge, different kind of architectures used for building whyQA systems, gaps and challenges that have hampered progress in advancing whyQA systems did not yet surveyed extensively which have motivated us to write this paper. The key research objectives of our work:

- Compare among whyQA systems of both Arabic and nonArabic languages.
- Detect the gaps and challenges, which exist in Arabic whyQA systems
- Highlight the future trends to guide the new research in that area.

The other part of the paper is organized as follows: In section 2, the reviews of the existing architectures for whyQA systems. In section 3, the Arabic why question answering datasets are mentioned. In section 4, the presentation of the related works in the area of whyQA system. In section 5, we discuss the main challenges and gaps of Arabic whyQA systems. Conclusions with some future directions are finally presented in section 6.

## 2 WHYQA SYSTEMS ARCHITECTURES

A typical Question Answering System (QAS), as shown in Fig. 1, consists of three main phases: Question analysis, passage retrieval, and Answer extraction. There are three types of whyQA system architectures, which are derived from the general architectures of any QAS [4]. They can be classified as follows: architecture based on information retrieval (IR) techniques, architecture based on semantic web (SW) techniques and architecture based on deep learning (DL) techniques.The following sub-sections detail these three architectures.

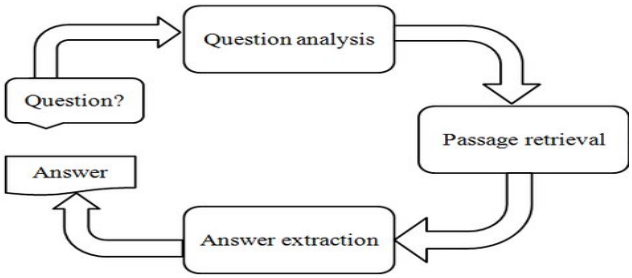


**Figure 1: Question answering system general architecture**

### 2.1 WhyQA System Architecture based on IR techniques

This kind of architecture usually consists of four main components:
Question analysis, document preprocessing, document retrieval and answer extraction. Below, we will explain each component in more details.

#### 2.1.1 Question Analysis

It is responsible for exploring the questions before sending it to the document retrieval component.
It consists of the following six steps [5]:

- Tokenization: the question is divided into small elementary units, generating a list of terms.
- Normalization: the diacritical markings are removed and all variants of the letters are unified.
- Stop-words removal refers to the process of eliminating all those words that occur very frequently in the documents and cannot add any benefits to its context. Prepositions and conjunctions are good examples of noisy words.
- Stemming: A stemmer is applied on each keyword to obtain its root.
- Formulation and generation for the query.
- Question Expansion refers to adding synonyms for nouns and adjectives found in the question to the list of question terms with the objective of reducing words ambiguity between the questions and candidate documents and increasing the effectiveness of retrieval [6][1].

#### 2.1.2 Document preprocessing

The documents in data source are processed before moving to the document retrieval subtask by applying the normalization, stop-word removal, tokenization, and stemming.

#### 2.1.3 Passage Retrieval

The Passage Retrieval (PR) is the main component in any QA system. The effectiveness of whyQA system depends mainly on the quality of the PR component. PR utilizes the query that is generated by the previous component, Question Analysis, to retrieve and extract the ranked list of candidate's passage that have the highest probability of containing an expected answer to the user query [7]. There were different techniques that have been investigated for the PR. The candidate's passage retrieved by that components are passed to answer the extraction for further extracting and processing an exact answer.

#### 2.1.4 Answer Extraction

The final phase of QAS is the answer processing. Its goal is to return the appropriate answer that is extracted from the retrieved passages by using the previous component, PR. The inputs for answering the extraction component is the "bag of words" of why questions and a list of ranked relevant passage. This component might fail in returning the exact answer if the passages retrieved weren't relevant to the question keywords and do not contains the answer [5].

### 2.2 WhyQA System Architecture based on SW techniques

This architecture has four main components, which are domain knowledge, question analysis, ontology mapper and answer retrieval. The different QA systems might use different implementation for each module [8][9].

#### 2.2.1 Domain Knowledge

Domain knowledge is represented by ontology where key domain-specific concepts and their relationships are stored. The ontology could be constructed manually [10] through a knowledge management tool to generate the ontology dictionary as Resource Description Framework (RDF)/Web Ontology Language(OWL) file, or automatically from text documents [2][9]. In whyQA systems, the representation of causality in domain ontology is taken in consideration which includes cause and purpose relations [10].

#### 2.2.2 Question Analysis

This component also analyzes and processes the user query following the same steps mentioned in section 2.1.1, but here as part of a speech tagger POS and a name entity recognizer NER are added. POS helps in determining the type of a verb, word, noun, adjective, etc., while NER classifies words into predefined categories like the names of organizations, locations, persons, expressions of times, monetary values, percentages, quantities, etc.

#### 2.2.3 Ontology Mapper

This component is responsible to map keywords extracted from the user's query with the contents of the ontological dictionary. The matched keywords are used later to build the RDF triple patterns of the SPARQL query [11].

#### 2.2.4 Answer Retrieval

This component is considered as the most important part of the whyQA system because it is responsible for translating the natural language query to SPARQL and executes it with the aim to retrieve results from the RDF store. The construction of SPARQL query involves two steps: First, terms are extracted and then relation patterns. Verb classification and the causality representation are utilized together for building the query-triple. There are two kinds of why-answers comprising cause and motivation, where their detection depends on verb classification. Why questions with process, affect, existential verbs, adjective phrase and modal auxiliary verbs has a cause answer type, while those questions with actions verbs have a motivation answer type [10].

### 2.3 WhyQA System Architecture Based on DL Techniques

WhyQA systems that utilize this kind of architecture do not need linguistic tools and it can be used in different domains or languages. The approaches for non-factoid QAS are usually categorized in three directions [12]:

- The deep framework learn the distributed vectors representation of given questions and it answers the candidates and then the use of a similarity metric for measuring the matching degree [13][14].
- The joint feature vectors are constructed based on both the answer and the question, then the task can be converted to learning-to-rank problem or a classification [15].
- Using notions of machine translation evaluation (MTE) to rank answers [16].

All the above approaches share a general architecture which is composed out of two basic blocks word Embedding and neural networks (NNs).

#### 2.3.1 Word Embedding

Word embedding is the most popular algorithm for dense representations of a word in the form of numeric vector [17]. It maps a word with similar meaning to get similar representations. The two main methods for learning word embedding are dependent on the contextual knowledge.

- Count-based is unsupervised method based on the matrix factorization of global word occurrence matrix with an assumption that a word in the same context share similar or related semantic meanings [17].
- Context-Based is a supervised method that could be:
  - Skip-gram model predicts the probability of words being a context word for the given target. The words that are at the middle of a sliding window represents the target, while those words on its left and right are the context.
  - Continuous bag-of-words: Instead of predicting context words from a word vector, it predicts one word from the sum of all word vectors in its context.

Global Vector (GloVe) model is another algorithm for word embedding and it was proposed by [18], aiming to make a combination between the context-based skip-gram model and the count-based matrix factorization.

#### 2.3.2 Neural Networks

There are different variants of NNs that were employed and combined to construct whyQA systems, such as sequence-to-sequence model, long short-term memory (LSTM), convolutional neural network (CNN), and recurrent neural network (RNN) [19] [20]. The detailed descriptions and architectures of these NNs are out of the scope of this paper. Neural Networks can be used alone, or be stacked with others forming different layers [13][21][14]. Examples of deep architectures for why-answers selection could be:

- Stacked layers could be built on the questions and the answers of the candidates, then use the similarity metric for measuring the distance of the question answer pairs [14].
- Four stacked columns of NN could be also employed for processing the relevant causality expressions, a question, an answer passage, and the inner passage causality expressions [13].
- Other approaches employed direct adaptations of the feed-forward NN for MTE. It takes two answer candidates c1 and c2- which plays a role of the two translation hypotheses-The NN has to make a decision whether c1 is better answer than c2 to question q- which will play the role of the translations reference [16].

## 3 ARABIC WHY QUESTION ANSWERING DATASETS

One of the main drawbacks of Arabic whyQA systems is the scarceness of available Arabic training datasets. The key factor behind the progress and the promising performance of the Arabic systems is the release of large datasets, whether it is restricted-domain or open-domain. In the following we present the available Arabic why question answering dataset:

The dataset utilized in [22] consist of 98 why-and-how questions which were extracted from news website. The authors used a number of texts, each 150-350 words. They distributed the collected text to 15 people from different disciplines and they asked them to formulate why-and-how questions.

The dataset used in [23] were collected from the website Wikipedia. The authors had courage to consider all types of questions, including why-and-how. The dataset contains 335 questions included 43 how and 45 why questions. The answer was generated by two

methods, automatically by determining which of the sentences contain the highest weight and manually by a human expert. Authors tested the system using a collection of 75 reading-comprehension tests with 45 why questions.

Authors in [24] conducted their evaluation on a set of 250 why questions selected by 30 professionals and who were Arabic native speakers, in different fields (politic, computer, religion, history, science). Each field has 50 questions.

Corpus used in [25] consisted of 500 documents extracted from the website Arabic Wikipedia. The questions set consisted of 80 questions which is divided into two sets: one set consists of 40 why questions and the other set consists of 40 how questions.

Authors in [26] tested their system using 110 why questions posed by people whose answer is for sure to be in the data set using a corpus of 700 documents extracted from the Open Source Arabic Corpora (OSAC). This corpus with about 5000 from CNN Arabic and another 4700 articles from BBC Arabic.

399 questions included 58 why-questions are used in [27]. The questions which contain seven categories (what, when, where, who, how many/much, how and why) have been collected from several sources. Frequently asked questions(FAQ), namely, discussion forums and some of the questions translated from Text REtrieval Conference(TREC).

The Arabic Reading Comprehension Dataset (ARCD) is one of the two datasets that were used to develop and test the system in [28] which composed of 1,395 questions included 30 why questions posed by crowdworkers on Wikipedia articles, the other dataset used in [28] includes 48,344 translated questions from the Stanford Question Answering Dataset (Arabic-SQuAD) with 159 why questions.

Authors tested the system in [29] using 414 automatic questions that have been extracted from the Arabic websites related to Dubai's e-government services, they considered all types of standard questions, including 41 complex questions of how and why, but the authors did not mention the number of why systems that used in the dataset.

Dataset for Arabic Why Question Answering System(DAWQAS): It includes 3205 pairs of whyQAs covering 8 different domains which are: politics and economy, sports, arts and celebrities, technology and science, religion and philosophy, nature and animals, society & women, and health and nutrition. The dataset scraped from public Arabic websites [17].

## 4 OVERVIEW OF THE EXISTING WHYQA SYSTEMS.

To answer why questions presents a big challenge as it requires some kinds of reasoning and a whole different approach than factoid questions, as their answer is not named entity. The answers to why questions are often complex and long passages with the possibility of implicit explanations [6] [30]. This explain why we see little attempts to build QA systems that answers why questions. We present some examples of whyQA systems of both Arabic and nonArabic languages to make a comparative study among them and to detect those gaps and challenges, which exist in Arabic whyQA systems to build a holistic view of them.

### 4.1 Non-Arabic WhyQA Systems

Authors in [30] proposed a whyQA system based on the discourse structures and relations in a pre-annotated documents collection, the Rhetorical Structure Theory (RST)-Treebank. The most relevant answer could be extracted based on the RST relations between the questions topic and its answers span. The proposed system was tested using 336 why question and answer pairs and reported the recall of 53.3%. Authors suggested filtering out the questions with the aim to raise the recall to 73.9% and Mean Reciprocal Rank (MRR) of 66.2. In a subsequent work for the same authors, they described approaches for ranking answer to why questions by evaluating number of machine learning technique in their performance that were described by a set of 36 linguistically motivated features [31]. The best score of MRR (0.35) was obtained by using Pairwise Support Vector Regression algorithm with tuning.

NAZEQA is a Japanese whyQA system which was presented in [32]. Authors explored the utilities of intra- and inter-sentential causal relations to rank the answer of candidates to why questions. They proposed a method to evaluate the suitability of causal relation as answers to given questions using semantic orientations of the excitation. The obtained precision, recall and F1-measure were 83.20%, 71.10% and 77.00% respectively.

A whyQA system based on ontology is proposed in [10] that takes in consideration the expected answer types. When the constructed SPARQL query has been executed over the domain ontology, some additional semantic entities were added to expand the question. The proposed method was implemented in Java programming language and the supporting ontology schema was constructed using Protégé [10]. Authors conducted two kinds of evaluation. The first one was performed by comparing outputs of the system against the manual identifications of set of query-triples, a set of ontology compliant query triples, and a set of semantic entities of a why question. The average value of the precisions and the recalls measure were greater than 99%. The second evaluation was performed by conducting the experiments that retrieved document for why questions where the questions were generated in random 20, 40, 60, 80, and 100 questions. The experiment was conducted in 20 iterations. The evaluation performances were the average values of each measure from the 20 iteration results. The evaluation showed small value of efficiency, only around 45% and only about 65% of the most relevant documents were the top-10 documents.

Causality attention CA-MCNN for whyQA system [13] presented a new NN model called multi-column convolutional neural networks (CA-MCNN) with new attentions mechanism. The main contributions to this research were, first automatically recognize the explicitly expressed causalities from text archives and use them for complementing the implicitly expressed causalities in answering passage. Second, identifying implicitly expressed causes with a set of words. The system performances were better when word embedding was used and they were 54.0 and 50.5 for precision and Mean Average Precision (MAP), respectively.

A new framework for non-factoid answer selection was proposed in [14]. It is based on building bidirectional long short term memory (biLSTM) models on questions and answers respectively, connected with a pooling layer and using a similarity metric for measuring the matching degree. A CNN structure was added on top of biLSTM and an attention model for answer generation according to the question context was also also introduced. Authors conducted experiments using the TREC-QA and InsuranceQA datasets and their results achieved the best results for MAP (71.11) and MRR (83.22) metrics and outperformed the baselines.

### 4.2 Arabic WhyQA systems

As the Arabic language is a syntactically rich language and the computer based processing of the Arabic documents is not an easy task, many challenges are added to Arabic whyQA systems. As a result, we find a very few researches in Arabic whyQA systems done and compared to other languages.

In [22] authors proposed the study that addresses the problem to find the answers to how and why questions expressed in the natural Arabic language. The developed system used one of the leading theory in computational linguistics called RST. It is based on the cue phrases to determining the elementary units and sets of rhetorical relations that is relevant to the targeted question. Their system obtained performance of 55% of recall. Experiment was conducted on Arabic unstructured texts that automatically annotated.

QArabPro, for reading comprehension tests [23], authors in QArabPro, introduced a QA system. First, sentences were assigned a score using a word match function. Later the sentences with the top scores were isolated and referred to these sentences as “BEST”. Each sentence score was then reinitialized to zero and the set of why rules were applied to every sentence in the text. The system tested using the collection of reading comprehension text. The HumSent is referred to answers that human expert judge to be the best answer for every question. The AutSent answer was generated automatically by determining which sentence contained the highest weight, excluding stopwords. QArabPro assumes that the answer should exist within one of the documents that was used as a corpus. The system achieved an overall accuracy of 84%. However, the accuracy for why and how questions was low, 69.77% and 62.22%, respectively.

EWAQ is a whyQA system is based on entailment metrics. The authors in [24] used a modified Cosine directional similarity for textual entailment. This system queries the search engine. For example, Google, pick the first seven retrieved passages. After that, the system measures the degree of entailment similarities between the why question and each of the seven retrieved passages. The passages are re-ranked according to the entailment similarities, and the system select the passages with the highest measures. If the selected passage is made up of more than one sentence, the same entailment metric is applied on each sentence, and the answer is the sentence with maximum degree of entailment similarity. The accuracy report of EWAQ is 68.53%.

In [25] authors used the term frequency inverse document frequency (tf-idf) to weigh for retrieving relevant documents from the corpus. Question classification seek in identifying what the question is looking for. If a question starts with why then the question is classified as “REASON”. If the question starts with how, it is classified as “MANNER”. Authors used Vector Space Model to develop the IR module for retrieving relevant documents from Arabic Wikipedia corpus. For the total 40 why questions, the obtained precision was 67.00% and the Recall is 62%. The F1 measure was 64%. The performance of QAS for answering the why questions was 64%. Vector Space Model was used to develop the IR module for retrieving a relevant document from Arabic Wikipedia corpus.

Lemaza is an Arabic why-question answering.Authors in [26] introduced Lemza system that handles questions starting with “Why” and “What are the reasons of”. The general architecture of Lemza system is an extended version of the work proposed in [26] and [5], but the main difference is it is able to find answers anywhere in the Arabic non-structured documents. In addition to that, authors designated different priorities of RST relations for retrieving answers, in case there was more than a candidate answer. The first priority corresponds to relation “JUSTIFICATION”, the second is for relation “BASE”, and the other candidates have the same priorities. Otherwise, one candidate is selected randomly of the candidate answers. Authors conducted different experiments. In the first one, Lemza system was tested on 110 why-questions and answers pairs and the overall performances were 79.21% and 72.73% for precision and recall, respectively. In the second experiment, authors evaluated the impacts to omitting an individual preprocessing steps on the overall performances. Authors got best performance when stop words weren’t removed. In the third experiment Lemaz´s performance was compared with [30] and [22]. Authors concluded that their new system reported better recall performance. Finally, Lemza was tested using the dataset employed in [23] and the obtained accuracy was 75% which is higher than the reported accuracy (62.22%) of QArabPro system [23]. Lemza whyQA problems are oriented for open domain non-structured documents under the assumptions that the answers exist somewhere in the corpus of textual Arabic documents. Lemaza’s best performance measures were, 72.73%, 79.21% and 78.68% to recall, to precise and c@1, respectively. These were achieved when stop words were retained. c@1refers to correctness [26].

Authors in [27] focused on disambiguation based on NooJ language development platforms to use sets of linguistic resources. They are collected questions that contain seven categories. For every question, the Arabic medical texts on the internet have been studied and collected to define the disambiguation patterns. The authors constructed a transducer and five dictionaries for each type of question. Furthermore, the questions have been processed using morphological grammar, syntactic grammar and dictionaries in order to get useful keywords that allows the extraction of the correct answers. The overall performances were 93%, 78% and 89% for recall, precision, and F-measure respectively. As mentioned in [27] the special focus is on the ambiguity resolution enhances the F-Measure by 28%.

SOQAL is open domain question answering system. Authors in [28] introduced SOQAL that composed of three modules: a document retriever using a hierarchical TF-IDF approach, a machine reading comprehension module using the pre-trained bidirectional transformer BERT, and linear answer ranking module. The authors developed and tested the system using two datasets: the Arabic Reading Comprehension Dataset (ARCD) and 48,344 translated questions from the Stanford Question Answering Dataset (Arabic-SQuAD). The authors train BERT under different data regimes in order to evaluate the effectiveness to use translated data as training data on the ARCD test sets. They found that when combining both datasets, they obtain an improvement of 8.3% on the F1 score. the performance measure was 61.3 F1 score with the BERT-based reader, and 27.6 F1 score for the open domain system SOQAL.

Arabic QA system was implemented in [29] for answering user questions that is relevant to Dubai's e-government service. The authors constructed an Arabic ontology to the knowledge base for the question answering system through four stages; they developed a data tool to the automatic dataset extraction, then the ontology keywords have been extracted and linked to the ontology component by the mapping rule process. Lastly, the ontology was constructed using OWL format and Protégé tool standard [33]. Depending on the e-government services ontology dataset that has been extracted. The users' questions analyzed using NLP techniques and translated from natural language to SPARQL queries used two methods: semantics-based and keyword-based. In order to retrieve the correct answer. The test questions are executed for both approaches, semantics-based and keyword-based, and the results show high precision in the semantics-based approach with 95%, while the precision in the keyword-based approach achieved only 72%. In addition, recall in the keyword-based approach achieved 100%, that indicated a high reliability level, whereas recall in the semantics-based approach reached 94%. The accuracy result for the semantics-based and keyword-based approaches are 90% and 72%, respectively.

Table 1: Comparison among different whyQA systems.

| Ref. | Architecture Type | Dataset | Domain Type | Language | Evaluation Metrics |
|---|---|---|---|---|---|
| [30][31] | IR | 336 questions from which 279 with topics | Open | English | Recall and MRR |
| [32] | IR | 500 yahoo websites+350 manually built | Open | Japanese | Recall, Precision and F1measure |
| [16] | SW | Ontology constructed manually and using Websites+ Verberne's w*hy* question collection | Open | English | Recall, Precision, Undergeneration and Overgeneration |
| [14] | DL | TREC-QA and InsuranceQA | Open&Closed | English | MAP and MRR |
| [13] | DL | 4-billionpage web archive | Open | Japanese | Precision and MAP |
| [9] | IR | Arabic websites 98 *why*-questions | Closed | Arabic | Recall |
| [23] | IR | Wikipedia. 75 reading comprehension tests with 335 questions. | Open | Arabic | Accuracy |
| [24] | IR | Websites. 250 questions | Open | Arabic | Accuracy |
| [25] | IR | Wikipedia. 500 documents with 40 questions | Open | Arabic | Recall and Precision |
| [15] | IR | OSAC, 110 questions | Open | Arabic | Recall, Precision and c@1 |
| [27] | IR | Arabic websites / TREC | Closed | Arabic | Recall, Precision and F1measure |
| [28] | DL | Arabic-SQuAD(Translated SQuAD) / ARCD(Arabic Wikipedia) | Open | Arabic | F1measure |
| [29] | SW | Arabic webpages /414 automatic questions. | Closed | Arabic | Accuracy, Recall and Precision |

## 5 GAPS AND CHALLENGES FOR ARABIC WHYAQ SYSTEMS

Table 1 compares among the different *why*QA systems cited in the above section. As can be noted that Arabic whyQA systems suffer from the following issues:

From architecture side: none of the existing Arabic *why*QA systems has explored the fields of SW and DL [25][23][24][15][22]. All of them are based on IR techniques.

- No Domain-specific: most of the existing Arabic whyQA systems were focused on open domain, which may suffer from poor structured documents that require special parsing and retrieval tools.
- Lack of linguistic resources: researchers face many challenges due to the rareness of the linguistic resources and training corpus for Arabic whyQA systems, the limited capabilities leads the researchers to build manually their own corpus which required vast human annotation and verifications.
- The lack of a common corpus prevented different methods and approaches to be comparable to one another. This is one of the main drawbacks of Arabic whyQA systems.
- There is not a specific categorization for Arabic why-questions neither for why-answers.
- There is not a standard evaluation metric among the whyQA systems. This is observed either in Arabic or nonArabic systems. This fact makes impossible to compare their performances and conclude which methodology is the best.

## 6 CONCLUSIONS AND FUTURE DIRECTIONS.

There have been a lot of studies and surveys covering Arabic QASs, tools, resources, and test-sets so far [34][6] and ignoring those difficulties that Arabic *why*QA systems are suffering, therefore our main motivation is to present the first study that fills this gap. In this paper, we analyzed thirteen *why*QA systems and their contributions either in Arabic or in other languages. Non-Arabic *why*QA is an active researching field with many diverse and existing approaches covering multitudes of research challenge, domain and knowledge base; thus we have to learn many lessons from those approaches and try to implement them in Arabic *why*QA systems with the objective to get systems with high quality in answer selection. This fact creates scope for several improvements in Arabic *why*QA systems. Some of these scopes are described as follows:

- Extraction of whyQA pairs from QA community sites such as Ejaaba.com and Asalni.com. Users of these kind of sites provides answer to the question asked by other users and best answer is selected manually either by all the participants or by voting by the same user who posed the question.
- The use of the free resources of SW. Quran Ontology [35], Quranic Arabic corpus [36] and DBpedia [37] are three good examples that can be used to build Arabic whyQA system based on SW. Qur'an Ontology and Quranic Arabic Corpus cover the Qur'an knowledge, while DBpedia is an ontology that contains knowledge from 111 language editions from Wikipedia, including Arabic language. The Qur'an Ontology was employed by author in [38] for building Arabic QASs, but why-question was not included. Salni system was proposed in [39] for only Arabic factoid questions, but authors stated that their system can't answer all factoid question due to the absence of the information in DBpedia knowledge framework. Consequently, enhancements should be done on DBpedia for achieving better results in Arabic language for factoid and non-factoid questions.
- The use of AraVec model for building *why*QA systems based on DL techniques. AraVec is a pre-trained distributed word representation open source project that aims for providing the Arabic NLP research community with free use and powerful words embedding model [40].
- Creating a standard dataset for the research community, containing big data of why question-answer pairs, covering different domains and with annotations. Annotations could include for example why-question and why answer categories, rhetorical relations expressed either explicitly or implicitly. Data could also include multimedia (images and videos) resources to enhance research in visual Arabic whyQA systems.
- Creating standard evaluation metrics for why-answer ranking with the objective to make more feasible the comparison among different whyQA systems, regardless of languages, application areas and employed architecture.

Therefore, we conclude that there are a lot of gaps for researches in the field of Arabic *why*QA systems. The recent trends in the field of *why*QA systems are to be part of chatbots for smartphones with the objective to offer more interactive dialogue with the user in different domains.

## 7 REFERENCES


[1] Lahbari, I., El Alaoui, S. O., & Zidani, K. A. (2018). Toward a new arabic question answering system. Int. Arab J. Inf. Technol., 15(3A), 610- 619.

[2] Bedini, I., & Nguyen, B. (2007). Automatic ontology generation: State of the art. PRiSM Laboratory Technical Report. University of Versailles.

[3] Mishra, A., & Jain, S. K. (2016). A survey on question answering systems with classification. Journal of King Saud University-Computer and Information Sciences, 28(3), 345-361.

[4] Kolomiyets, O., & Moens, M. F. (2011). A survey on question answering technology from an information retrieval perspective. Information Sciences, 181(24), 5412-5434.

[5] Azmi, A., & AlShenaifi, N. (2014). Handling "why" questions in Arabic. In The 5th International Conference on Arabic Language Processing (CITALA'14).

[6] S. Ray, K. Shaalan (2016). A Review and Future Perspectives of Arabic Question Answering Systems, IEEE Transactions on Knowledge and Data Engineering, 28(12):3169-3190, IEEE.

[7] Ismail, W. S., & Homsi, M. N. (2018). Dawqas: A dataset for arabic why question answering system. Procedia computer science, 142, 123-131.

[8] AbuTaha, A. W., & Alagha, I. M. (2015). An Ontology-Based Arabic Question Answering System. Central library of Islamic University of Ghaza.

[9] Kalaivani, S., & Duraiswamy, K. (2012). Comparison of question answering systems based on ontology and semantic web in different environment. In Journal of Computer Science.

[10] Karyawati, A. E., Winarko, E., Azhari, A., & Harjoko, A. (2015). Ontology-based why-question analysis using lexico-syntactic patterns. International Journal of Electrical and Computer Engineering (IJECE), 5(2), 318-332.

[11] Hartig, O., Bizer, C., & Freytag, J. C. (2009, October). Executing SPARQL queries over the web of linked data. In International Semantic Web Conference (pp. 293-309). Springer, Berlin, Heidelberg.

[12] Tan, M., Dos Santos, C., Xiang, B., & Zhou, B. (2016). Improved representation learning for question answer matching. In Proceedings of the 54th Annual Meeting of

the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 464-473).
[13] Oh, J. H., Torisawa, K., Kruengkrai, C., Iida, R., & Kloetzer, J. (2017, February). Multi-column convolutional neural networks with causality-attention for why-question answering. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 415-424). ACM.
[14] Tan, M., Santos, C. D., Xiang, B., & Zhou, B. (2015). LSTM-based deep learning models for non-factoid answer selection. arXiv preprintarXiv:1511.04108
[15] Azmi, A. M., & Alshenaifi, N. A. (2017). Lemaza: An Arabic why-question answering system. Natural Language Engineering, 23(6), 877903.
[16] Guzman, F., Màrquez, L., & Nakov, P. (2016). MTE-NN at SemEval-2016 task 3: Can machine translation evaluation help community question answering? In SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings (pp. 887-895). Association for Computational Linguistics (ACL).
[17] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
[18] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
[19] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7(3–4), 197-387.
[20] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press.
[21] Li, H. (2017). Deep learning for natural language processing: advantages and challenges. National Science Review.
[22] Salem, Z., Sadek, J., Chakkour, F., & Haskkour, N. (2010, September). Automatically finding answers to" Why" and" How to" questions for Arabic language. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 586-593). Springer, Berlin, Heidelberg.
[23] Akour, M., Abufardeh, S., Magel, K., & Al-Radaideh, Q. (2011). QArabPro: A rule based question answering system for reading comprehension tests in Arabic. American Journal of Applied Sciences, 8(6), 652.
[24] AL-Khawaldeh, F. T. (2015). Answer Extraction for Why Arabic Questions Answering Systems: EWAQ. World of Computer Science & Information Technology Journal, 5(5).
[25] Ahmed, W., & BabuAnto, P. (2016). Answer Extraction for how and why Questions in Question Answering Systems. International Journal of Computational Engineering Research. 06. 18-22.
[26] Azmi, A. M., Alshenai_, N. A. (2016). Answering arabic why-questions: Baseline vs. rst-based approach. ACM Transactions on Information Systems 35(1):6:1{6:19, DOI 10.1145/2950049, URL http://doi.acm.org/10.1145/2950049
[27] Dardour, S., Fehri, H., & Haddar, K. (2020). Disambiguation for Arabic Question-Answering System. In International Conference on Automatic Processing of Natural-Language Electronic Texts with NooJ (pp. 101-111). Springer, Cham.
[28] Mozannar, H., Hajal, K. E., Maamary, E., & Hajj, H. (2019). Neural arabic question answering. arXiv preprint arXiv:1906.05394.
[29] Albarghothi, A. B. E. (2018). An Ontology-based Semantic Web for Arabic Question Answering: The Case of E-Government Services (Doctoral dissertation, The British University in Dubai (BUiD)).
[30] Verberne, S., Boves, L. W. J., Oostdijk, N. H. J., & Coppen, P. A. J. M. (2007). Discourse-based answering of why-questions ,Traitement automatique des Langues (TAL), Published by Association pour le traitement automatique des langues (ATALA), Paris France 47(2): 21–41.
[31] Verberne, S., van Halteren, H., Theijssen, D., Raaijmakers, S., & Boves, L. (2011). Learning to rank for why-question answering. Information Retrieval, 14(2), 107-132. Conference Name:ACM Woodstock conference
[32] Oh, J. H., Torisawa, K., Hashimoto, C., Sano, M., De Saeger, S., & Ohtake, K. (2013). Why-question answering using intra-and intersentential causal relations. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1733-1743).
[33] Albarghothi, A., Saber, W., & Shaalan, K. (2018). Automatic construction of e-government services ontology from Arabic webpages. Procedia computer science, 142, 104-113.
[34] Shaheen, M., & Ezzeldin, A. M. (2014). Arabic question answering: systems, resources, tools, and future trends. Arabian Journal for Science and Engineering, 39(6), 4541-4564.
[35] Hakkoum, A., & Raghay, S. (2015). Ontological approach for semantic modeling and querying the Qur'an. In Proceedings of the International Conference on Islamic Applications in Computer Science And Technology.
[36] Atwell, E., Brierley, C., Dukes, K., Sawalha, M., & Sharaf, A. B. (2011). An Artificial Intelligence approach to Arabic and Islamic content on the internet. In Proceedings of NITS 3rd National Information Technology Symposium (pp. 1-8). Leeds.
[37] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2015). DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2), 167-195.
[38] Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N., & Torki, M. (2014). Al-Bayan: an Arabic question answering system for the Holy Quran. In Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP) (pp. 57-64).
[39] Tahri, A., & Tibermacine, O. (2013). DBPedia based factoid question answering system. International Journal of Web & Semantic Technology, 4(3), 23.
[40] Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. Procedia Computer Science, 117, 256-265.