ACQAD: A Dataset for Arabic Complex Question Answering

Abdellah Hamouda Sidhoum, M'hamed Mataoui, Faouzi Sebbak, Kamel Smaïli

ACQAD: A Dataset for Arabic Complex Question Answering

Abdellah Hamouda Sidhoum[1*], M'hamed Mataoui[1], Faouzi Sebbak[1], and Kamel Smaïli[2]

[1] Computer Science Department, Ecole Militaire Polytechnique, Algiers, Algeria
[2] SMarT Group, Loria, University of Lorraine, France

Abstract. In this paper, we tackle the problem of Arabic complex Question Answering (QA), where models are required to reason over multiple documents to find the answer. Indeed, no Arabic dataset is available for this type of questions. To fill this lack, we propose a new approach to automatically generate a dataset for Arabic complex question answering task. The proposed approach is based on using an effective workflow with a set of templates. The generated dataset, denoted as ACQAD, contains more than 118k questions, covering both comparison and multi-hop types. Each question-answer pair is decomposed into a set of single-hop questions, allowing QA systems to reduce question complexity and explain the reasoning steps. We then provide a statistical analysis of the produced dataset. Afterwards, we will make the corpus available to the international community.

Keywords: Question answering, Arabic complex questions, QA dataset.

1 Introduction

Question Answering (QA) is a challenging task in natural language processing (NLP) that is used to evaluate machine reading comprehension (MRC). QA systems are designed to provide short answers to questions formulated in natural language. The majority of QA researches focus on single-hop QA, where a single paragraph is supposed to be sufficient to answer the question. Although, models' performance has been further boosted in recent years, particularly since the introduction of machine learning techniques such as BERT[5], they still lack the ability to perform multi-hop reasoning across multiple documents. A Multi-hop system has to aggregating dispersed pieces of evidence to predict the right answer (sentences highlighted in *blue italic* in Figure 1). In this example, the question can not be answered by matching its tokens with a single sentence in one paragraph. This area of research has recently received considerable attention, especially since the release of large-scale complex QA datasets, such as hotpotQA. [14] and ComplexWebQuestions [13].

Research in Arabic QA remains in its beginning stage. This delay is mainly due to the lack of datasets compared with those available for other languages, such as English [1],[10]. The existing Arabic QA corpora are either small datasets

Paragraph A, Algeria at the FIFA World Cup:
Algeria have appeared in the finals of the FIFA World Cup on four occasions in 1982, 1986, 2010 and 2014. They have once qualified for the knockout rounds.,...,. *In 2014, Algeria qualified for the first time into the round of 16.*
Paragraph B, 2014 FIFA World Cup:
The 2014 FIFA World Cup was the 20th FIFA World Cup, the quadrennial world championship for men's national football teams organized by FIFA. *It took place in **Brazil** from 12 June to 13 July 2014, after the country was awarded the hosting rights in 2007....*
Question: Where was the world cup hosted that Algeria qualified for the first time into the round of 16?
Answer: Brazil

Fig. 1: An example of the multi-hop questions in HOTPOTQA.

or unavailable publicly, and do not cover all categories of questions in terms of type, domain and complexity. Furthermore, these datasets remain limited to simple questions, where answers can be extracted from a single document. In contrast, complex questions, that require reasoning over multiple documents to infer the answer, have not been studied. To the best of our knowledge, there are no datasets for complex question answering in the Arabic language. Moreover, collecting complex questions is not trivial.

To address the above challenges, we propose a workflow to automatically generate questions in order to produce a dataset for Arabic complex QA. The approach covers both comparison and multi-hop questions, where the reasoning over more than one document to find the answer is required. We rely on a structured representation of information about a subject, named infobox, from Arabic Wikipedia articles as data source. Moreover, we use a set of predefined templates to create questions. The produced dataset provides, for each question-answer pair, a set of passages as context and a set of sub-questions along with their corresponding answers as a decomposition of the complex question into simpler questions.

The remainder of the paper is organized as follows: Section 2 reviews existing Arabic question answering datasets. The proposed methodology followed to build the new dataset is detailed in section 3. Several statistics about the generated dataset are presented in section 4. Finally, a conclusion and future work will conclude this paper.

2 Related Work

Arabic is one of the most spoken languages in the world, mainly in the Middle East and North Africa region. Despite the large community of speakers, research in Arabic QA is limited in terms of linguistic resources compared to other languages, with only a few datasets proposed. The investigated existing datasets, recently published in [1, 3] surveys, can be classified according to their construction approaches into three classes:

Machine Translation (MT) based datasets: It is a practical method to generate datasets by translating well established datasets from other languages. For instance, Arabic-SQuAD [8] which is a machine translation of SQuAD 1.1 [12], is composed of 48k paragraph-question-answer tuples. Atef et al. [2] presented AQAD, consisting of more than 17k questions and answers translated from the SQuAD 2.0 [11].Othman et al. [10] use Google translation to translate into Arabic a dataset released by [15]. The dataset was harvested from all categories in the popular Yahoo! Answers community platform. However, this class of datasets suffers from poor translation due to linguistic differences and complexity.

Crowd-sourced datasets: depend on hired crowd workers to create the dataset from scratch or to eliminate issues presented in existing resources. For this category, we cite the ARCD (Arabic Reading Comprehension Dataset)[8] and TyDiQA [4]. ARCD is composed of 1395 factoid questions asked by crowd workers on articles from Wikipedia. TyDiQA is a multilingual QA dataset that covers eleven typologically diverse languages including Arabic, with 204K question-answer pairs. However, Crowd-sourcing approach is time-consuming and requires funds to hire crowd workers.

Web scraping based datasets: rely on automatic process to retrieve QA resources from the Web such as community QA (cQA) sites. In this direction, a medical Arabic corpus for cQA named CQA-MD was proposed by Nakov et al. [9]. The corpus contains over 100k questions-answers pairs collected from Arabic Medical websites. Ismail and Homsi [7] introduced DAWQAS, a dataset for Arabic *Why* QA systems. The dataset contains 3205 *Why* question-answer pairs scraped from public Arabic Websites. The Web scraping based approach is preferable when the targeted question type or domain are available on the Web. However, it requires considerable efforts to annotate the crawled data.

In order to overcome the drawbacks of existing approaches, another method for constructing QA datasets could be based on **Automatic generation**. This method mainly relies on structured sources using logical rules and templates. This approach is more appropriate when the resources for the chosen question type are scarce. This motivate us, through the current work, to develop a method to automatically produce a dataset for Arabic complex QA.

3 Methodology

A carefully designed dataset has an impact on the robustness of the systems as well as the performance of the models built on it. For this reason, and due to the unavailability of a dataset for the Arabic complex QA task, we designed a simple workflow to automatically generate a dataset for the Arabic complex QA task. Figure 2 describes the main steps involved in generating two types of questions: comparison and multi-hop questions.

3.1 Comparison questions generation process

A comparison question is the type of questions that compares two or more similar entities in some aspects of the entity [14]. For instance, the compared entities

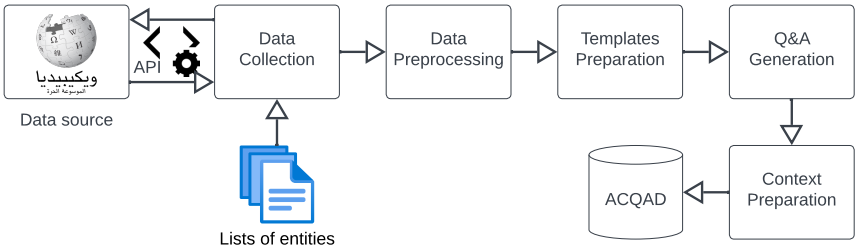

Fig. 2: The workflow of the proposed methodology to create ACQAD

from the question : ؟ من أصغر مساحة، السويد أم الأوروغواي (Which country has a smaller area, Sweden or Uruguay?), are *Sweden* and *Uruguay*, and the aspect of comparison is the area. The idea here is to find pairs of similar entities (A, B) that share common aspects or properties f, and then create the question Q using a template T. To find the answer a, property values $f(A)$ and $f(B)$ for entities A and B respectively are compared and the result determines the answer.

Data Collection. We start by manually curating lists of entities from the same category: animals (92 entities), Arabic cities (22 entities), and world countries (191 entities), totaling 305 entities. Then, we need to retrieve properties of these entities to be used as comparable aspects. To accomplish this task, we used the Wikipedia API[3] and BeautifulSoup[4] library to crawl and parse the infobox from the Wikipedia page of each entity (see Figure 3). Properties are presented in the infobox as (property; value) tuples. This structured representation facilitates the data collection process and eliminates the need for advanced NLP techniques to extract the properties of an entity from plain text.

Pre-processing and properties selection. Since the data gathered from Wikipedia infoboxes was entered by non-professional contributors and was not subject to any formal writing guidelines. Contributors can use words in different languages and introduce information using different styles and formats. Therefore, we performed pre-processing on the raw data to make it usable for question and answer generation. We first cleaned the data by removing special characters, diacritics, links, and non-Arabic words. Then, we normalized the writing of property labels and values in order to have common properties and comparable values.

For example, the property label الفصيلة (family of an animal), may also be found written as فصيلة. This is the same word as before, though without the prefix "ال". That prefix is the definite article in the Arabic language, typically translated as " the" in English. In this situation, we remove the prefix. Another case in numerical values, the expression "نسبة مئوية", or the symbol "%" both are used to describe a percentage. We chose to replace the expression by

[3] https://ar.wikipedia.org/w/api.php
[4] https://pypi.org/project/beautifulsoup4/

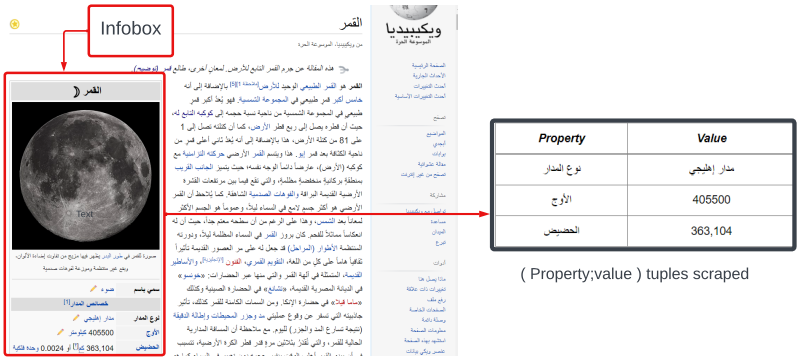

Fig. 3: An example of some data collected from a Wikipedia infobox.

the symbol. In numbers writing, hundreds and thousands parts of a number can be found separated by a dot ".", a comma "," or a space, and the same thing for decimal numbers, which causes ambiguity. For hundreds and thousands, we removed all separators, and for decimal numbers, we preserved the dot ".".
After normalization, we proceed to properties selection. We remove all uncompleted property-value tuples where the property or the value is an empty string (e.g., titles of property groups). We reject all non-comparable properties, such as الموقع الرسمي (official Website), and maintain only comparable properties with values for at least two entities. Finally, we categorized the selected properties into quantitative and qualitative.

Templates preparation. We define manually a set of templates to generate comparison questions and their sub-questions. For each property as an aspect of comparison, we create a list of templates that express the same question but in different ways. This will create more diversity while producing questions. Each template T has two variable tokens: X and Y, to be replaced by two entity labels. For instance, comparing two animals in terms of *gestation period*, ما الحيوان الذي لديه فترة حمل أطول ، X أم Y ؟ (Which animal has a longer gestation period, X or Y ?).

Templates are categorized according to the answer type into: Yes/No questions, or Choice questions where the answer is one of the compared entities. For more variety, we add templates with the opposite predicates of the ones used in the previously prepared templates. For the previous example, instead of أطول (longer), we use أقصر (shorter).

Furthermore, we provide each comparison question with its decomposition in the form of two sub-questions derived from the initial question. Each sub-question seeks the value of an entity's property. Similarly, a set of templates for the sub-questions is created for each property with one variable X to be replaced by each entity label. For the previous example, one of the sub-question templates would be: كم تدوم فترة الحمل عند X ؟ (how long is the gestation period of X?).

Generation method. From the set of entities, we make a list of combinations of two entities A and B from the same category. For each couple of entities, we generate questions about common properties in which they have values. For each property f, we select a random template and replace variable tokens in the pattern string with entity labels.

To produce the answer for the generated question, we compare the property values $f(A)$ and $f(B)$ considering that the question concerns superiority or inferiority, yes/no, or a choice between entities and whether the property is quantitative or qualitative. We obtain at the end four types of answers: yes/no, equality if $f(A) = f(B)$, and one of the compared entities A or B if it concerns a choice question. Subsequently, we generate the two sub-questions in the same manner, with the difference that the answer for each sub-question is the property value.

As our dataset is a text-based extractive dataset, we provide two paragraphs as context. Each paragraph is obtained by extracting the text in the infobox from the entity's Wikipedia page. These paragraphs serve also as a context for the sub-questions. The final structure of the dataset contains generated comparison question-answer pairs along with the two compared entities, the comparison aspect, two paragraphs as context, and two sub-questions with their answers. Eventually, we randomized and sampled the generated dataset.

3.2 Multi-hop questions generation process

Multi-hop questions require a model to reason using information taken from multiple documents to determine the answer [14]. Consider the following question, كم تبلغ مساحة المدينة التي نظمت الألعاب الأولمبية الشتوية لعام 1928 ؟ (How large is the area of the city that organized the 1928 Winter Olympics?). The model must identify, as a first hop, "*the city that organized the 1928 Winter Olympics*", and then "*its area*". From the example, we notice that it can be split into three parts: a hidden entity (a city), an unambiguous feature of this entity (organized the 1928 Winter Olympics), and a property of the hidden entity (the area). The idea to form a 2-hops question is to ask about a property of an entity that has an unambiguous feature. Therefore, the first hop is to find the entity that has the unambiguous feature, and the second hop is to determine the answer to the question concerning that entity's property. We propose below a formal method to generate multi-hop questions that enable to automatically produce a corpus. In the following, we note by $E = \{x\}$ a set of entities, $R = \{r\}$ a set of unambiguous features, and $F = \{f\}$ a set of properties.
With:

$$r(X) = x, x \in E \tag{1}$$

x is the hidden entity which is the answer to the first hop.

$$f(x) = y \tag{2}$$

Where y is the value of the property f for the entity x, considered as the answer to the second hop.

A 2-hops question is formed by replacing the entity x by $r(X)$ from equation 1 in equation 2.

$$f(r(X))) = y \tag{3}$$

Example:

let's take an unambiguous features r_1: أكبر بلد مساحة في أفريقيا (*the biggest country in Africa*).

$$r_1(X) = \text{الجزائر}\,(Algeria) \tag{4}$$

الجزائر (Algeria) is the hidden entity x. Let's take the property f_1 : الرئيس (*president of*).
Using the equation 2,

$$f_1(\text{الجزائر}) = \text{رئيس الجزائر}\,(president\ of\ Algeria) = \text{عبد المجيد تبون} \tag{5}$$

After substitution and adding an appropriate question word, the produced 2-hops question will be:
من هو رئيس أكبر دولة في أفريقيا؟ (*who is the president of the biggest country in Africa ?*), and the answer to this question would be: عبد المجيد تبون

Data collection. To generate multi-hop questions in the form described above, we need to collect entities having unambiguous features. We chose these features to be either unique, such as records, or time-related, such as events, to ensure that only one entity has the feature. Table 1 shows the entity classes collected accompanied with unambiguous features examples. Beside that, information such as the competition date, its round, the record set, the tournaments season, the start and the end dates are also collected to be used in the subsequent steps of the workflow. Next, we collect properties of the entities from Wikipedia infoboxes and we perform the same pre-processing steps as described in section 3.1.

Templates preparation. We propose a set of templates in order to produce a rich diversity of questions. Multi-hop questions templates are created as a concatenation of the triplet : question word, property label and unambiguous feature phrase. The question words are defined depending on the collected property types (number, date, etc.), and the property gender (masculine or feminine). Table 2 illustrates the appropriate question words used for some collected properties.

Regarding the unambiguous features phrase, we use the information available to formulate this part of the question template. For world and nature records category, the unambiguous features are in form of superlative expressions. We use these expressions as unambiguous feature phrases. Concerning Olympic records

Entity class	# Entities	Unambiguous features Type	Example
animals, countries, places, and buildings	50	world and nature records	أسرع الثدييات (The fastest mammal)
players	25	Olympic records	الرقم القياسي في سباق 100 م (100 meter world record)
players' countries	21		بلد اللاعب يوسين بولت (Usain Bolt's country)
host cities and countries	65	Olympic events	المدينة التي استضافت الألعاب الأولمبية الصيفية 2016 (The city that hosted the 2016 Summer Olympics)
tournaments	51		البطولة حيث سجل بولت رقما قياسيا في 100 م (The tournament where Bolt set 100m world record)

Table 1: Collected entity classes with unambiguous features examples

and Olympic events categories, we employ the available information, mentioned in section3.2, to formulate diverse, unambiguous features phrases. We use variable tokens that correspond to these information in the templates. For instance, اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G "The player who set the record in competition C in tournament G". C denotes the competition, and G denotes the tournament where the record was set. The full template example will be:

كم يبلغ طول اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G ؟

The decomposition of the multi-hop question consists of creating a sequence of sub-questions where each sub-answer solves a part or a hop of the question. In our case, we adapted the triplet used to construct the multi-hop questions templates for creating the sub-questions. The first sub-question template uses an appropriate question word (e.g., *who, what*) with the unambiguous feature phrase. The answer to this first sub-question is the hidden entity. The second sub-question template uses the question word and the property label from the triplet, in addition to the entity that is the answer to the first sub-question. The following sub-questions illustrate the case of the previous example:

- sub-question 1: من هو اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G ؟
- answer to sub-question 1: entity P
- sub-question 2: كم يبلغ طول اللاعب P ؟

The final step is to generate the questions and their decomposition. For each entity's property, we select the appropriate question word. Then, we randomly select one of the unambiguous feature phrases, and replace the variable tokens with their corresponding information.

Question word	property
كم تبلغ (How much)	الكثافة السكانية (population) المساحة (area) نسبة المياه (water ratio)
ما هو (What is)	رمز الهاتف (phone code) السن القانونية (legal age) نظام الحكم (regime)
ما هي (what is, for female)	الكنية (surname) العاصمة (capital) العملة (currency)

Table 2: Examples of appropriate question words for collected properties

3.3 Collecting questions contexts

To answer multi-hop questions, models need more than one paragraph as a context. For each question, we retrieve two passages called gold paragraphs. The first paragraph is the summary of the entity's Wikipedia article where the unambiguous feature appears. The second paragraph is the text in the Wikipedia infobox where we find the properties and their values. In all cases, we ensure that the answer to the first sub-question appears in the first gold paragraph, and the answer to the second sub-question appears in the second paragraph.

For both comparison and multi-hop questions, we add distracting paragraphs to the gold paragraphs, following Yang et al. [14] and Ho et al. [6] setting. These paragraphs are used to make the dataset more challenging and test the model's ability to find the answer in the presence of noise. We use the retriever module from the SOQAL system [8]. We first retrieve the top 10 articles from Wikipedia, which are very similar to the question using the 1-gram TF-IDF formula. Then we use it again to select, from the retrieved articles, the top-8 paragraphs as distractor paragraphs. Finally, we mix the gold and the distractor paragraphs to obtain the context.

4 Dataset Analysis

In this section, we analyse the dataset by providing statistics regarding the number of questions generated, questions and answers length, and the types of answers. All results are presented for comparison and multi-hop questions separately.

4.1 Quantitative analysis of generated questions

The statistics of the generated ACQAD dataset are presented in Table 3, where Q denotes the question and A denotes the answer. We provide the number of instances produced per entity type for comparison questions and per unambiguous feature type for multi-hop questions. The number of questions generated depends on the number of entities within each type and the properties retrieved for these entities.

The dataset consists of 118841 questions in total. Comparison questions have the most instances, while multi-hop questions have fewer since it is difficult to find entities with unambiguous features. The list of entities' types can be extended in the case comparison questions to cover more topics (for instance : people, companies, events, etc.).

The average answer length of comparison questions is smaller than that of multi-hop questions. This is due to the fact that there are numerous yes/no answers for comparison questions.

Question type					
Comparison			Multi-hop		
Entity type	**#Entities**	**#Examples**	**Unambiguous features type**	**#Entities**	**#Examples**
Animals	92	8625	world and nature records	50	519
Arabic cities	22	462	Olympic records	46	1425
World countries	191	106769	Olympic events	116	1049
Total	305	115856	**Total**	212	2985
#Avg. Q	10.47		**#Avg. Q**	19.05	
#Avg. A	1.14		**#Avg. A**	3.77	

Table 3: Statistics per type related to the generated questions

Figure 4 shows the distribution of question length for comparison and multi-hop questions. The varied lengths of questions represent various complexity levels. We obtain almost the same range of questions length compared to the benchmark hotpotQA [14] where most questions contain between 10 and 40 tokens.

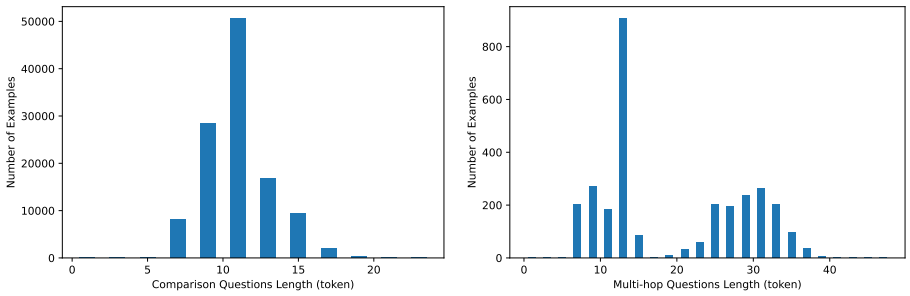

Fig. 4: Distribution of Questions lengths

4.2 Answer types

Answer types for comparison questions are restricted to yes/no, one of the compared entities for choice questions, or equality word تساوي if the property values being compared are equal. Statistics of answer types are given in Table 4. Yes/No type is abundant since most templates used while generating comparison questions are of type yes/no. As it is uncommon to find two equal values, the equality type is nearly non-existent.

Table 5 presents the answer types for the multi-hop questions. As is shown, the generated dataset covers a variety of questions centered around persons, organizations, locations, dates, and numbers, as well as other answer types. We notice clearly the domination of the number type because the majority of properties values are quantitative.

Answer Type	**%**
Yes/No	71.67
Entity	27.49
Equality	0.84

Table 4: Type of Answers for Comparison questions

Answer Type	**%**	**Examples(s)**
Number	53.9	62 سنة
Person	7.8	جو بايدن
Date	10.1	26 ديسمبر 1991
Location	5.7	منطقة التبت
Organization	1.3	جامعة ستانفورد
Other	21.2	دولار أمريكي

Table 5: Type of Answers for Multi-hop questions

5 Conclusion and future work

In this paper, we presented the creation process of ACQAD, an automatically generated dataset for Arabic complex question answering task. To the best of our knowledge, no dataset for Arabic language is available for this task. Our corpus consists of more than 118k questions. We relied on Wikipedia as data source and a set of predefined templates to generate high quality questions. The dataset provides with each question a set of sub-questions as decomposition. The proposed method can be adapted to any language that is lacking datasets for complex QA task. Future work will aim to establish baseline models with which researchers may compare their approaches and results. Furthermore, the focus will be on extending ACQAD by including more multi-hop examples. Since we now have a corpus, we will develop methods that leverage the data available to answer complex questions.

References

1. Alwaneen, T.H., Azmi, A.M., Aboalsamh, H.A., Cambria, E., Hussain, A.: Arabic question answering system: a survey. Artificial Intelligence Review pp. 1–47 (2021)

2. Atef, A., Mattar, B., Sherif, S., Elrefai, E., Torki, M.: Aqad: 17,000+ arabic questions for machine comprehension of text. In: 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA). pp. 1–6. IEEE (2020)
3. Biltawi, M.M., Tedmori, S., Awajan, A.: Arabic question answering systems: Gap analysis. IEEE Access **9**, 63876–63904 (2021)
4. Clark, J.H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., Palomaki, J.: Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics **8**, 454–470 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Ho, X., Nguyen, A.K.D., Sugawara, S., Aizawa, A.: Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. arXiv preprint arXiv:2011.01060 (2020)
7. Ismail, W.S., Homsi, M.N.: Dawqas: A dataset for arabic why question answering system. Procedia computer science **142**, 123–131 (2018)
8. Mozannar, H., Hajal, K.E., Maamary, E., Hajj, H.: Neural arabic question answering. arXiv preprint arXiv:1906.05394 (2019)
9. Nakov, P., Màrquez, L., Moschitti, A., Mubarak, H.: Arabic community question answering. Natural Language Engineering **25**(1), 5–41 (2019)
10. Othman, N., Faiz, R., Smaïli, K.: Learning English and Arabic Question Similarity with Siamese Neural Networks in Community Question Answering services. Data and Knowledge Engineering (101962) (Dec 2021). https://doi.org/10.1016/j.datak.2021.101962, https://hal.archives-ouvertes.fr/hal-03500114
11. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822 (2018)
12. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
13. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint arXiv:1803.06643 (2018)
14. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600 (2018)
15. Zhang, W.N., Ming, Z.Y., Zhang, Y., Liu, T., Chua, T.S.: Capturing the semantics of key phrases using multiple languages for question retrieval. IEEE Transactions on Knowledge and Data Engineering **28**(4), 888–900 (2015)