AraQA-BERT: Towards an Arabic Question Answering System using Pre-trained BERT Models

AFNAN H. ALSHEHRI
Faculty of Informatics and Computer Systems Department
College of Computer Science,
King Khalid University,
Abha,
SAUDI ARABIA

Abstract: - To increase performance, this study presents AraQA-BERT, an Arabic question-answering (QA) system that makes use of pre-trained BERT models. The study emphasizes how important QA systems are for promptly and accurately responding to user inquiries, especially when those inquiries are made in native tongues. Arabic QA systems are necessary because of the complexity and linguistic variances of the Arabic language, even if English QA systems have made substantial progress. The study examines the use of pre-trained language models, such as AraBERT and Arabic-BERT, for Arabic QA tasks with a focus on Modern Standard Arabic (MSA). The study's contributions include the creation of a web-based application named AraQA-BERT for open-domain QA, trials on TyDi and ARCD datasets, and a methodology for employing pre-trained models.

Key-Words: - AraQA-BERT, Modern Standard Arabic, Question-answering, QA systems, Linguistic variations, Arabic-BERT, Arabic-BERT.

Received: August 9, 2023. Revised: May 17, 2024. Accepted: July 6, 2024. Published: August 8, 2024.

1 Introduction

Information retrieval (IR) approaches are used with comprehension algorithms to obtain pertinent sections or publications that address user inquiries. The three steps of an IR-based QA system are explained: question processing, passage retrieval, and answer processing.

Ontologies like DBPedia or Freebase are used to represent structured databases. The process involves creating a semantic representation of the question and using parsers or query languages to extract answers. Rule-based and supervised methods are discussed as approaches for implementing knowledge-based QA systems.

Pre-trained language models (PTMs) are introduced to enhance natural language processing (NLP) tasks by leveraging knowledge from a pre-trained model. The advantages of PTMs, such as saving time and cost, are highlighted. BERT (Bidirectional Encoder Representations from Transformers) is mentioned as a significant pre-trained language model for QA systems. The pre-training and fine-tuning processes of BERT are explained.

The paper addresses several challenges in developing an Arabic QA system, including the ambiguity of the Arabic language, the absence of capital letters, and variations in dialects. However, advancements in pre-trained models like AraBERT and Arabic-BERT have shown promising results in NLP downstream tasks, including QA systems. Additionally, the availability of Typologically Diverse Question Answering (TyDi QA) and Arabic Reading Comprehension Dataset (ARCD) has facilitated high-quality research on QA systems for different languages.

The paper aims to build an effective and precise Arabic QA model by fine-tuning the AraBERT and Arabic-BERT models using the Fast-BERT library. The performance is assessed using F1 and Exact Match measures, and the TyDi and ARCD datasets are employed for this purpose. The design and implementation of the AraQA-BERT system, a web-based tool that extracts precise answers from the model's output, are also presented in the study. Usability evaluation is performed to explore the extent to which the suggested technique meets the user experience.

2 Related Works

In [1], proposed WebShodh to do the job of taking questions in telegraphic languages varying from Hindi to Telugu transforming them into English and

extracting the answers. WebShodh is describing the whole system for factoid questions. The system works on a Google Search API for relevant result searches. The MRR of the Hindi evaluation was found to be 0.37 while for Telugu it was 0.32.

As in [2] are the magicians, they presented the "Ask Your Neuron" protocol where visual and textual questions are the inputs to generate the answers. The system's performance was appraised using the scientifically approved DAUQAR and VQA datasets. This proposed approach was found to outperform the existing methods in terms of numeracy and recall which tend to give higher accuracy and recall by the textual questions.

In [3], constructed a neural network encoder-decoder, NQG++, which not only takes natural language as its input and accomplishes the task of harnessing its questions but also performs a decoding function on their answers. The performance of the proposed approach had been evaluated on a dataset called the Stanford Question Answer Dataset (SQuAD). Our proposed model NQG++ has got better scores, achieving a higher mean score of Neural Question Generation than the old PCFG-Trans model in the baseline.

Also, researchers have been looking into designing more automatic QA systems that could be used to navigate through the web in recent years. In [4] have got a system for basketeer for QA systems. The method was taking user questions in natural language, extracting various attributes to build the answer, and comparing the questions with items already present in the knowledge network. Named Entity Recognition (NER) self-training achieved an accuracy of 95.96%, and the system's overall test dataset accuracy was 93%.

English QA systems are using recent advances in pre-trained language models, including BERT [5]. To normalize answer scores and enhance performance on benchmarks such as Quasar-T, SearchQA, TriviaQA, and OpenSQuAD, in [6] employed BERT in numerous passages. Researchers in [7] similarly improved the BERT model by employing a community-based QA approach, and they were able to obtain a Mean Square Average of 0.046.

Research studies on English QA systems have incorporated several important datasets. In [8], authors assessed their system with the DAUQAR and VQA datasets, which have a lot of images and questions associated with them. In [9], made use of the Wikidata and DBpedia datasets. The DBpedia dataset is taken from a question-and-answer dataset based on the encyclopedic source Wikipedia and Wikidata is an open-source structured data where a considerable number of resources is included, Stanford Question and Answer Dataset (SQuAD), which naturally has a large dataset with human-made questions and answers in articles written on Wikipedia.

Adequate processing of questions is one of the principal components of such systems, which is to identify and classify user questions for corresponding answers. In this case, the "question understanding process" aims at distinguishing types of questions and meanings to eradicate ambiguity, and properly answer the question. ML techniques can function both in a way that answers questions are moved into processing by manners or through automated ways. Arabic QA systems question processing has been the center of several review studies although.

As stimulated in [10], there was a development of an Arabic-based Language Analyzer utilizing the NooJ system. Particularly, this project directly involved the difficult "why" and "what-is" questions in the Arabic process. The proposed approach presents important results, as shown by a recall value of 68% and mean reciprocal rank (MRR) of 0.62 which yields higher accuracy than the results of the control group.

Relying on the theories of [11], the researchers concentrated on answering the “why” and “how” questions in Arabic. The RST-based algorithm allowed the machine to produce texts according to the rules and the RST approach was applied. The performance of LEMAZA is as anticipated, sweeping the floor with already existing methods (Precision, 79.2%; Recall, 72.7%).

In [12], presented temporal problem-solving for Arabic QA systems that they strive to provide exact answers to questions. Extraction of verbs and named entities from the text and a corpus consisting of temporal sentences and part-of-speech representations were produced. The NLP mechanism fed the system with temporal pattern features that enabled it to compare the questions and answers to conclude. The constructed model seemed well suited to locate suitable solutions to time questions.

To build an Arabic Question-and-answering system, in [13] have applied planting lexical knowledge and knowledge rules in this procedure. They categorized time and position nouns and numbered entities by way of the morphological analysis. The system relied on NooJ's grammatical tapes and data-detection technology for answer recognition and conversion to class. The combined technique proved to be effective as the experimental results showed good precision and F-measure.

Study from [14] concentrated on Arabic QA systems and messages' question processing and answer pattern development. They produced responses by the time, focus, and topic categories they used to classify the questions. The precision, recall, and F-measure of the experiment were all good.

The study [15] developed an Arabic quality assurance system that contained answers and used natural language processing to extract passages. The system that was used for the identification of the queries and the generation of the replies was based on the dependency parsing and the named entity recognition. There was a good performance of the experiments with reasonable recall and precision.

In [16], the author introduced an Arabic quality assurance system based on pattern matching and semantic analysis in another work. The algorithm identified relevant segments from the questions through the use of a semantic analyzer. We employed pattern matching algorithms on the passages to search for the responses and recognize them. The results of the conducted experiment showed how effective the proposed approach is in Arabic QA systems.

For Arabic QA systems, researchers proposed a transfer learning-based method [17]. In the procedure of improving the efficiency of QA systems, large-scale datasets with pre-trained models were applied. The results of the trial demonstrated how transfer learning is done in Arabic QA systems; subsequently, the performance and accuracy improvements were realized.

In [18] explored the authors questioning as to how the BERT model could be enhanced for Arabic QA systems. They employed Arabic datasets from Quora and Stack overflow for enhancing the BERT model. The results of the trial were MSA of 0. 046, demonstrated enhanced performance.

Each of the research studies presented in this section contributed to the development of the guidelines or conventions for design and processing of questions. Based on four criteria, Table 1 compiles studies pertinent to a rules-based approach: The “Target source” denotes the name and the type of the analyzed data; the “Question analysis techniques correspond to the techniques used by the system under consideration as well as the “Question classification techniques correspond to the classification techniques adopted; and finally, the “Performance” section provides some of the experimental results obtained.

Authors in [20] presented an Arabic QA system with ontological development. They constructed an ontology for the pathology domain, establishing relationships between diseases, organs, reasons, and symptoms.

Table 1. Certain Arabic QAS rule-based features and techniques

System	Target source	Question analysis techniques	Question classification techniques	System Performance
QARAB [18]	Al-Raya Arabic newspaper text	NLP technique and IR	Using a set of known question type	Precision: 97.3%. Recall: 97.3%
QASAL [19]	Arabic Documents	NLP technique	Defined question types and forms using NooJ local grammar	Not mentioned
A Discourse-Based Approach for Arabic Question Answering [10]	Arabic corpus belonging to the Health, Science & Technology categories	NLP technique	Defined "why" and "how" questions	recall of 68% and MRR of 0.62
LEMAZA [11]	Open-Source Arabic Corpora (OSAC)	Rhetorical Structure Theory (RST) based algorithm	Defined "why" type of questions	Precision is 79.2%, and recall is 72.7%
A Rules-based Approach for Arabic Temporal Expression Extraction [13]	Pilot Arabic Propbank data	Morphological analysis using NooJ local grammar	No	An F-measure score of 95.5%
Generating Answering Patterns from Factoid Arabic Questions [14]	Not mentioned	Morphological Analysis of Factoid Arabic Questions using NooJ local grammar	Using a set of known question type	75% precision, 72% recall, and an F-measure of 73%

The dataset was manually generated using the Protégé tool for precise and relevant answer formation.

Basic NLP techniques such as tokenization, stemming, stop word removal, and lemmatization were applied to process incoming questions. The proposed methodology achieved a high precision of 81%, recall of 93%, and an F-measure of 86%, outperforming existing approaches.

In [21], developed a Community Question-Answer (CQA) model for Arabic using the SemEval-2017 Task 3 dataset. They coupled term-based and word2vec similarity measures with the QU-BIGR system similarity measure, and they used the Support Vector Machine (SVM) for data training. In a competition, the suggested method placed second out of all the systems in use, illustrating its usefulness for answering optimization in CQA systems.

In [22], researchers addressed the CQA issue and concentrated on question classification in Arabic QA systems. They used neural networks and supervised machine learning algorithms (SVM, logistic regression, and random forest) to categorize questions. SPLIT and MADAMIRA were utilized to encode the input data from the SEMEval2017 CQA dataset. Lexical and semantic features were the two feature types that were employed. The SVM method achieved a Mean Average Precision (MAP) of 62.85%, outperforming other algorithms. In the experiments, mean reciprocal rank (MRR) and average accuracy (AvgRec) were taken into consideration as evaluation criteria.

In [23], proposal included the Unstructured Information Management Architecture (UIMA) and a community-based Arabic QA system developed using neural network models. The Farasa NLP tool served as the foundation for the UIMA Arabic NLP architecture, and the system's performance was assessed using Mean Average Precision (MAP), Average Precision (AvgP), and P(k). The outcomes demonstrated encouraging MAP values without testing-related trimming.

In [24], presented an approach for question classification using SVM and Convolutional Neural Networks (CNN). The coarse class generated by SVM was used as input for neural networks to predict the subclass. The approach achieved an accuracy of 82% and included six rough classes: abbreviation, entity, description, human, location, and numeric value. The fine-grained classification was challenging due to the similarity among fine classes and the difficulty training instances within each fine class.

In [25], introduced a new Arabic taxonomy for QA systems using an ML approach. They collected data from various datasets and trained an SVM classifier. The proposed methodology achieved an accuracy of 90%, indicating its enhancement of existing QA systems.

In [26], presented the Visual QA System, which utilized bidirectional recurrent neural networks (BiSRU and BiLSTM) for feature extraction. The bidirectional approach improved answer prediction by considering historical and future contextual information. The Stacked Attention Model (SAM) and EnSAN were adopted for query attention modeling. Experimental results on the COCO-QA dataset showed an optimal accuracy of 2.2%.

These studies demonstrate the effectiveness of ML approaches, with the SVM classifier commonly used for accurate question classification. Neural networks have also gained popularity and have shown significant results in question categorization using ML techniques in Table 2.

Table 2. Some Arabic QAS machine learning features and techniques

System	Target source	Question analysis techniques	Question classification techniques	System Performance
Arabic Question Answering Using Ontology [21]	Their own Corpora, comprising 100 Questions	NLP technique	Using a set of known questions	Accuracy of 81%, recall of 93%, and an F-Measure score of 86%
QU-BIGR [22]	Arabic collection of questions and their potentially related question-answer pairs	NLP technique	SVM classifier	MAP score of 43.41, F1 score of 62.57
Arabic Question Classification Using Support Vector Machines and Convolutional Neural Networks [25]	TALA-AFAQ dataset	NLP technique	SVM and CNN classifier	The accuracy of the proposed architecture is 82%
Community Question Answering (CQA) for the Arabic language [23]	SEMEval2017 CQA dataset	NLP technique	SVM, logistic regression, and random forest classifier	SVM outperforms other algorithms as it generates 62.85% MAP
Visual Question Answer System Based on Bidirectional Recurrent Networks [25]	COCO-QA Dataset	NLP technique	Bidirectional recurrent neural networks (BiSRU and BiLSTM)	Best results for a COCO-QA accuracy of 2.2%

The lack of large pre-training corpora has been a challenge for Arabic language processing. In [23], addressed this issue by training four Arabic-BERT language models from scratch. These models,

named Mini, Medium, Base, and Large, were pre-trained on approximately 8.2 billion words from the OSCAR Arabic version, Arabic Wikipedia dumps, and other Arabic resources. The models were trained using Google BERT's repository with modifications to the training settings. These pre-trained models have been made publicly available and have contributed to enhancing Arabic NLP applications.

Another significant development is the AraBERT model. AraBERT is a transformer-based model for Arabic that provides a language representation based on the BERT model. It was trained using the BERT base configuration and 70 million sentences from large Arabic corpora. The model was fine-tuned for downstream NLP tasks, including question answering (QA). AraBERT achieved state-of-the-art performance in various Arabic NLP tasks and is publicly available for research and applications. Furthermore, BERT has been used for open-domain factoid Arabic QA systems, [27]. Their approach combined a document retriever using hierarchical TF-IDF and BERT for answer extraction, achieving superior results compared to existing methods.

Additionally in [28], introduced multilingual BERT, which includes 104 languages. This development has contributed to the success of NLP applications and led to the creation of language-specific BERT models.

These studies demonstrate using sizeable pre-trained language models, such as BERT, for various Arabic NLP tasks, including QA. These models have significantly advanced the field and provided researchers with powerful tools for Arabic language processing. In [29], contributed to passage extraction by formulating queries using POS tagging and stemmed words, followed by passage extraction from relevant documents using an Arabic Wikipedia dataset. Their approach showed promising results in terms of query-passage similarity.

In [30], presented the dataset for the Arabic Why Question Answer System (DAWQAS), consisting of 3205 "why" questions. The dataset was constructed through several steps: obtaining data, data preprocessing, identification of the same type of questions, and probability-based labeling. DAWQAS was trying to cover the absence of datasets that explain "why" questions in the Arabic language since no other datasets existed.

In [31], proposed the QA model TyDi which consists of 922 pairs of questions and answers in Arabic. TyDI QA covers 11 languages that are typologically different and is designed to expose linguistic characteristics as well as to help reconstruct the data for various types of speech using data scenarios and systems. Table 3 is an example of the accessible datasets of the QA in the Arabic language.

Table 3. Some Available Arabic QAS Datasets

Dataset	Source	Formulation	Size
Arabic-SQuAD [29]	Translated SQuAD	Paragraph, Question and Answer	48,344
ARCD [30]	Arabic Wikipedia	Paragraph, Question and Answer	1,395
ArabiQA [32]	Wikipedia	Question and Answer	200
DefArabicQA [33]	Wikipedia and Google search engine	Question and Answer with documents	50
Translated TREC and CLEF [34]	Translated TREC and CLEF	Question and Answer	2,264
QAM4MRE [35]	Selected topics	Document Question and multiple answers	160
DAWQUAS [30]	Auto-generated from a web scrape	Question and Answer	3205
QArabPro [36]	Wikipedia	Question and Answer	335

3 Methodology

The choice of the data set TyDi QA [31] has been made for two main reasons: (1) the dataset supports a number of languages and (2) the data was gathered using a realistic approach. The collection comprises 11 genetically not related languages, such as Arabic, with the number of QA pairs being equal to 200k and a pack of 922 pairs developed for the Arabic language.

Besides posing questions, people using TyDi ask them without any knowledge, and as a result, the dataset applies perfectly to our project. The dataset is collected without translation, ensuring authenticity and language-specific characteristics. The Arabic Reading Comprehension Dataset (ARCD) [27] is chosen for system evaluation. The ARCD consists of 1,395 questions written by proficient Arabic speakers. It offers answers to diversity and questions reasoning categories of a comprehensive assessment.

The methodology involves the following steps:

- Data Pre-processing: The input QA datasets (TyDi and ARCD) undergo cleaning and

processing to prepare the data for training and testing.
- Dataset Splitting: The datasets are divided into training and testing sets using the random state in train_test_split.
- Fine-tuning: The pre-trained models are fine-tuned for the QA system task, enabling the models to learn specific textual properties relevant to Arabic QA.
- Answer Selection: The model is trained to select a span of text containing the answer given a question and passage. This is achieved by predicting 'start' and 'end' tokens that mark the answer span.
- Passage Retrieval: The system retrieves the passage containing the exact answer based on the predicted answer span.
- Evaluation: The results of the QA system are evaluated using accuracy, precision, F-measure, and Exact Match metrics. A benchmark approach, such as the TF-IDF reader, is used for comparison.

The high-level architecture of the model is illustrated in Figure 1.
- The fine-tuning process involves training the model to predict the 'start' and 'end' tokens in the passage, followed by passage retrieval and answer classification.
- The evaluation metrics are used to assess the performance of the proposed approach compared to the benchmark method.

Software Tools

Python, a high-level, general-purpose, and interpreted programming language, is used for experiments and evaluation. Python provides various tools and a broad standard library suitable for multiple tasks. Project Jupyter is used as the execution environment for running the Python code. It supports interactive computing in various programming languages and is widely adopted by cloud providers.

Collaboratory, or Colab, is a free Jupyter notebook environment running in the cloud and storing notebooks on Google Drive.
- The proposed approach is evaluated using four metrics: precision, recall, F-measure, and Exact Match.
- Precision measures the accuracy of the system in identifying correct answers.
- Recall determines the coverage of the system in retrieving relevant answers.
- The F-measure combines precision and recall into a single metric.
- Exact Match measures the percentage of predictions that exactly match the ground truth answer.

The experiments evaluate the pre-trained AraBERT and Arabic-BERT models using different train-test splits and epochs of the training dataset. A baseline comparison is performed using a TF-IDF reader based on 4 grams, a popular method for feature extraction.

Two sets of experiments are conducted:
1. Different train-test splits (80/20 and 50/50) and 15 epochs of training.
2. Other train-test splits (80/20 and 50/50) and five epochs of training.

Each dataset (TyDi and ARCD) is tested separately, and then the datasets are merged for further evaluation. The Fast-BERT model, supporting QA tasks, is used in all experiments. The results of each model and the TF-IDF reader baseline are reported and compared in Table 4.

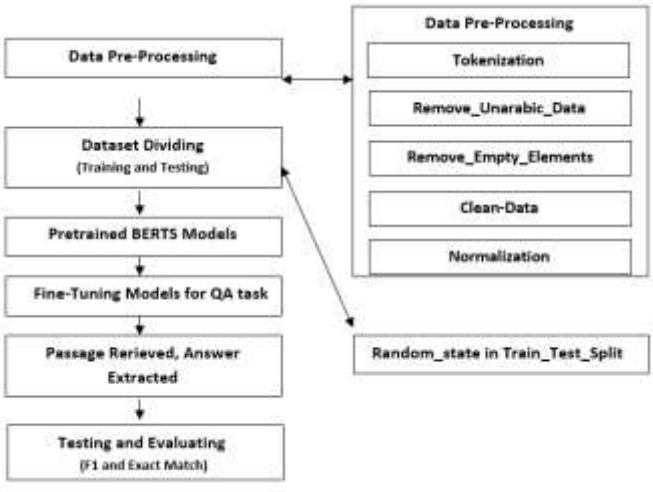

Fig. 1: Question Answering Environment Model

Table 4. Comparison of the different reader modules results for the TyDi and ARCD datasets, alongside the merged dataset's training on 15 epochs. For all evaluation processes, time was measured using the minutes unit

Method	TyDi			ARCD			TyDi + ARCD		
	EM	F1	Time	EM	F1	Time	EM	F1	Time
TF_IDF Reader	0	3.6	-	0	7.7	-	0.22	4.6	-
AraBERTv1	40	58.3	30	24	54.1	46	32.7	62.2	81
AraBERTv0.1	49.1	63.4	15	29	59.6	45	34.9	63.8	79
Arabic-BERT Mini	27.5	41.8	3	11.4	37.4	5	22.1	47.2	9
Arabic-BERT	42.7	55.3	10	19.7	48.5	16	29.3	54.6	30
Medium	**57.8**	**71.6**	**87**	30.4	65.7	138	37.7	68.1	162
Arabic-BERT Large	48.1	63.3	27	22.9	54.2	43	31.8	60.2	79
Arabic-BERT Base									

Table 5. Comparison of the different reader modules results for the TyDi and ARCD datasets, alongside the merged dataset's training on five epochs. For all evaluation processes, time was measured using the minutes unit

Method	TyDi			ARCD			TyDi + ARCD		
	EM	F1	Time	EM	F1	Time	EM	F1	Time
TF_IDF Reader	0	3.6	-	0	7.7	-	0.22	4.6	-
AraBERTv1	40	57.1	9	20.7	56	15	29	59.5	15
AraBERTv0.1	50.8	63.4	9	24	59	15	27.1	51.7	17
Arabic-BERT Mini	24.3	36.7	1	8.9	30.5	2	9.2	25.4	3
Arabic-BERT Medium	35.1	48.2	3	16.1	45.1	5	22.4	48.8	7
Arabic-BERT Large	**49.1**	**64.5**	**29**	25.4	60.5	55	34.9	63.9	54
Arabic-BERT Base	44.8	58.6	9	21.8	54.5	14	30.8	58.3	17

Table 6. Comparison of the different reader modules results for the TyDi and ARCD datasets, alongside the merged dataset's training on 15 epochs. For all evaluation processes, time was measured using the minutes unit

Method	TyDi			ARCD			TyDi + ARCD		
	EM	F1	Time	EM	F1	Time	EM	F1	Time
TF_IDF Reader	0	4.1	-	0.14	6.4	-	0.17	4.2	-
AraBERTv1	34.4	51.4	25	25.4	56.5	40	28.8	58.3	140
AraBERTv0.1	39.4	54.3	23	24.3	55.1	38	31.1	58.5	82
Arabic-BERT Mini	24	40.5	3	10.9	34.8	5	15.7	38.4	10
Arabic-BERT Medium	32.9	50.9	9	20.5	46.8	15	26.6	52.1	21
Arabic-BERT Large	**41.6**	59	71	30.1	61.7	118	34.9	**63.7**	235
Arabic-BERT Base	37.3	54.2	24	20.3	50.9	38	29.1	55.2	79

Table 7. Comparison of the different reader modules results for the TyDi and ARCD datasets, alongside the merged dataset's training on five epochs. For all evaluation processes, time was measured using the minutes unit

Method	TyDi			ARCD			TyDi + ARCD		
	EM	F1	Time	EM	F1	Time	EM	F1	Time
TF_IDF Reader	0	4.1	-	0.14	6.4	-	0.17	4.2	-
AraBERTv1	33.6	52.4	7	20.6	54.2	14	29.1	59.2	29
AraBERTv0.1	36.2	51.8	8	24.7	53.2	13	31.9	57.5	28
Arabic-BERT Mini	21.2	37	1	4.7	23.5	2	11.4	31.5	3
Arabic-BERT Medium	29.2	45.1	3	19.9	46.2	5	25.1	51.2	18
Arabic-BERT Large	**40.9**	58.8	14	28.2	60.13	41	35.7	**63.7**	46
Arabic-BERT Base	31.4	50.5	8	21.3	52.1	13	27.6	54.8	27

Table 5 reports all of the model results for the previously mentioned datasets in experiment-1, with different training of the models using the training set for five epochs with the same learning rate.

We split the datasets into 50-50% train/test. The TyDi training set contained 460 pairs, and the testing set included 461 questions and answers. Subsequently, we used the ARCD, split into a training set containing 693 teams and a testing set containing 702 pairs of questions and answers. Lastly, we merged both previous datasets, ARCD + TyDi, splitting them into a training set containing 1152 pairs and a testing set containing 1163 questions and answers. We trained the models using

a training set for 15 epochs, with a learning rate of 2e-5. Table 6 reports all of the model results along with the TF-IDF reader baseline, which evaluates the testing sets for each previously mentioned dataset.

Table 7 reports all of the model results for the previously mentioned datasets, with different training of the models using the training set for five epochs with the same learning rate.

The results of the experiments demonstrate significant improvements over the baseline TF-IDF Reader. The Arabic-BERT Large model consistently achieved the highest accuracy in all cases, while the Arabic-BERT Mini model showed lower accuracy but faster evaluation times.

Regarding specific datasets, the TyDi dataset achieved the most accurate results, with an F1 score of 71.6 and an Exact Match (EM) score of 57.8. However, the evaluation process for this dataset was time-consuming. On the other hand, the ARCD dataset had lower accuracy, with an F1 score of 23.5 and an EM score of 4.7.

It was observed that AraBERTv0.1, which does not require pre-segmentation, achieved better accuracy with an F1 score of 63.7 and an EM score of 50.8 compared to AraBERTv1 in all cases while also having shorter evaluation times.

The improved performance of the pre-trained models can be attributed to several factors. The data size of Arabic-BERT Large played a crucial role, as training with a larger dataset led to more accurate results. Conversely, the smaller data size of Arabic-BERT Mini resulted in lower accuracy. The larger data size also provided more diversity in the pre-training distribution, contributing to better performance. Additionally, pre-segmentation before BERT tokenization, as in AraBERTv1, reduced the effectiveness of the QA task, while omitting pre-segmentation, as in AraBERTv0.1, yielded more accurate results.

4 Results

4.1 AraQA-BERT System

Given the lack of Arabic open-domain QA systems, we implemented the AraQA-BERT system, considered an available domain QA system for the Arabic language. It is comprised of three module components: 1) a document retriever (TF_IDF) for obtaining relevant documents from Arabic Wikipedia sources, 2) a neural reading comprehension pre-trained AraBERT module for extracting answers from retrieved documents, alongside 3) an answer ranking module for ranking the answers in order of relevance, based on taking in scores from the document retriever and AraBERT reader. Figure 2 presents our proposed system architecture.

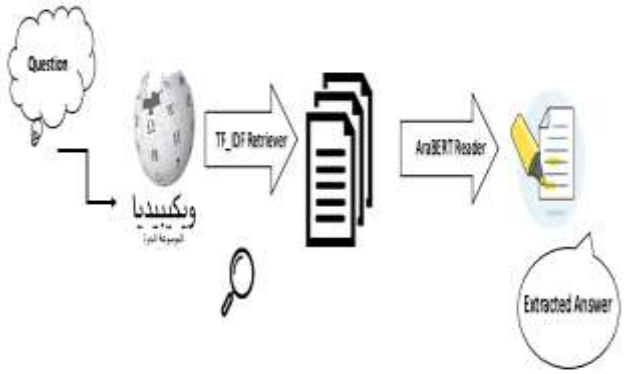

Fig. 2: Overview of the Architecture for our Question Answering system

4.1.1 TF-IDF Document Retriever

A practical document retrieval method following classical QA systems was adopted to initially narrow our search space, concentrating on reading only articles with a likelihood of being relevant. The module aims to select the most pertinent documents from Arabic Wikipedia for the question, thus limiting our reader's search span.

Each document is tokenized and stemmed using Arabic NLTK, where stop words are removed, with the vector of each document then normalized. Subsequently, the TF-IDF uses the term vector model to represent documents and questions as vector weights of identifiers, and articles and questions are compared with the TF-IDF weighted bag-of-word vectors. Afterward, each document is scored using cosine similarity, which measures the similarity between two non-zero vectors (Question and paper).

Finally, the document retriever returns the top five relevant Arabic Wikipedia articles with the highest similarity to any question. These articles are then passed to the document reader for processing and answer extraction.

4.1.2 AraBERT Document Reader

Our proposed reader module is AraBERTv0.1, one of two versions of the AraBERT model, attaining an F1 score of 63.41 and 50.81% of exact match accuracy on a TyDi dataset. We selected it for our proposed system based on its high accuracy and acceptable evaluation time that gives the extracted answer 30 seconds to 1 minute compared to the

Arabic-BERT large, which takes 5 to 7 minutes to answer the question.

The input text was initially tokenized using SentencePiece (an unsupervised text tokenizer and detokenizer) consisting of 64k tokens, which are then embedded. Each question and paragraph pair input point is represented as a single sentence separated by a special pass. For each permit I in the section, we compute the probability that i is the answer's start or end. Afterward, we predict the span (i, j), where i is the token with the highest probability of being the answer's start token. In contrast, j is the token with the highest likelihood of being the answer's end token.

4.1.3 Answer Ranking

Alongside the top retrieved documents from step 1, we obtained a score from the TF-IDF retriever per document, considering cosine similarities between the paper and question. We got a score per candidate answer per retrieved document that passed as an input to the document reader.

Following normalization, we passed them through the softmax function to ensure that the answer and document scores were on the same scale, providing us with outputs and a vector representing the probability distributions for a list of outcomes.

4.2 AraQA-BERT User-Interface Design

The designed web-based tool interfaces for our AraQA-BERT open domain q QA system are presented in Figure 3 and Figure 4.

Fig. 3: Overview of our AraQA-BERT Home page

Fig. 4: Overview of AraQA-BERT Features page

4.3 Evaluation and Result

Our primary focus was previously on evaluating pre-trained models AraBERT and Arabic-BERT. Due to the time constraint, we evaluated our system by conducting simple and not extended user evaluation sessions.

The evaluation was conducted on five participants who tried the web system tool. The description of the evaluation method used is as follows:

a) Participants characteristics: users from different educational environments, and their ages ranged from 20 to 35 years. Most of the participants have experience in using website tools. They are selected to test the AraQA-BERT system to ensure that it achieves the objectives of the system.

b) Tasks: after explaining the general idea of the system, tasks were given to the participants as a list of steps to be carried out one by one as follows:

1)	Enter five different questions in the Question textbox 'الاسؤال.'
2)	Click on the button 'إرسال'
3)	Analyze whether the extracted answer is true or not.

c) Testing Sessions: The experiment was conducted at the exact location. We set the "AraQA-BERT" system for each user, read the tasks for them, and offer assistance to the participants if they counter a problem. Finally, the observers interviewed the participants after the session to ask them, "How many correct answers did they get during the experiment?" and "How did they find use of the system in general". The time set for experiencing the "AraQA-BERT" system is 10 minutes unless the participants need more time or ask for expansion. In general, the system test took one hour to complete.

The result of our post-session interview shows that out of 25 questions tried by users, seven questions correctly retrieved their answers. During the testing sessions, we observed that the system could correctly retrieve related paragraphs containing answers to the question. Still, the issue is not scoring the correct paragraph highly enough.

5 Discussion

The results reported in all of the above tables show significant improvement in the results for all of the modules, over the baseline TF_IDF Reader. Arabic-BERT Large was the most accurate model in all cases. Table 4 evidenced that the TyDi dataset attained a 71.6 F1 and 57.8 EM, which were the most accurate results the model attained across all experiments. However, its shortcoming is that the evaluation process was time-consuming. Meanwhile, the Arabic-BERT Mini was the less accurate model in all instances, albeit with less evaluation time. As presented in Table 7, the ARCD dataset attained a 23.5 F1 and 4.7 EM, thus being the less accurate results across all experiments.

Furthermore, we observed that AraBERTv0.1 necessitates no pre-segmentation, achieving advanced accuracy results with 63.7 F1 and 50.8 using the TyDi dataset over AraBERTv1 in all cases, during close evaluation periods.

This boost in performance for the pre-trained models has numerous explanations. For example, the data size of the Arabic-BERT Large is a clear factor in enhancing performance, because training with a higher data size provides more accurate results. Contrastingly, smaller data size offers less accurate results, as apparent with the Arabic-BERT mini. With the large data size, the pre-training distribution is characterized by greater diversity. Concerning the pre-segmentation applied before BERT, tokenization reduced the QA task performance efficacy, as with AraBERTv1, although with no pre-segmentation applied more accurate results were found, as with AraBERTv0.1. We believe that these factors facilitated the proposed models reaching the state-of-the-art level for the QA task.

6 Conclusion

In conclusion, this work focused on developing an Arabic QA model using pre-trained AraBERT and Arabic-BERT models. The experiments on multiple datasets demonstrated significant improvements over the TF-IDF reader baseline, with Arabic-BERT Large achieving the highest accuracy. AraBERTv0.1, which does not require pre-segmentation, showed better results than AraBERTv1.

Furthermore, an open domain QA system was built using Arabic Wikipedia as a knowledge source, consisting of three main modules. The system achieved an F1 score of 63.41 and 50.81% exact match accuracy on the TyDi dataset, with acceptable evaluation times. A user evaluation session with 5 participants and 25 questions showed that seven were correctly answered.

Future work will expand the experiments by using different datasets and exploring different hyperparameters to improve the performance of the pre-trained models. Additionally, efforts will be made to enhance the system's paragraph selection and answer extraction capabilities.

References:

[1] Jones, Gareth JF, Seamus Lawless, Julio Gonzalo, Liadh Kelly, Lorraine Goeuriot, Thomas Mandl, Linda Capellato, and Nicola Ferro. "Experimental IR meets multilinguality, multimodality, and interaction. *In Proceedings of the Eighth International Conference of the CLEF Association, Dublin, Ireland, September 11–14, 2017. Lecture Notes in Computer Science (LNCS)* (Vol. 10456)." (2023, May). https://doi.org/10.1007/978-3-319-65813-1.

[2] Malinowski, Mateusz, Marcus Rohrbach, and Mario Fritz, "Ask Your Neurons: A Deep Learning Approach to Visual Question Answering." *International Journal of Computer Vision*, 125 (1–3), 2017: 110–35. https://doi.org/10.1007/s11263-017-1038-2.

[3] Zhou, Qingyu, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. "Neural Question Generation from Text: A Preliminary Study." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018 10619 LNAI: 662–71, (2023, April). https://doi.org/10.1007/978-3-319-73618-1_56.

[4] Li, Ying, Jie Cao, and Yongbin Wang. "Implementation of intelligent question answering system based on basketball knowledge graph." *In 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China, pp. 2601-2604. IEEE, 2019,

(2023, May). https://doi.org/10.1109/iaeac47372.2019.8997747.

[5] Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *In Proceedings of naacL-HLT*, vol. 1, p. 2. (2019, Minnesota), (2023, April). http://arxiv.org/abs/1810.04805.

[6] Wang, Zhiguo, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang, "Multi-Passage BERT: A Globally Normalized BERT Model for Open-Domain Question Answering." *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5878–5882. Hong Kong, China: Association for Computational Linguistics, 2019, (2023, May). https://doi.org/10.18653/v1/D19-1599.

[7] Annamoradnejad, Issa, Mohammadamin Fazli, and Jafar Habibi. "Predicting subjective features from questions on QA websites using BERT." In *2020 6th international conference on web research (ICWR)*, Tehran, Iran, pp. 240-244. IEEE, 2020, (2023, May). http://arxiv.org/abs/2002.10107.

[8] Hashemi Hosseinabad, Sayedshayan, Mehran Safayani, and Abdolreza Mirzaei. "Multiple answers to a question: a new approach for visual question answering." *The Visual Computer*, 37, no. 1 (2021): 119-131.

[9] Diefenbach, Dennis, Kamal Singh, and Pierre Maret. "WDAqua-Core0: A Question Answering Component for the Research Community." *Communications in Computer and Information Science*, 769: 84–89, 2017, https://doi.org/10.1007/978-3-319-69146-6_8.

[10] Sadek, Jawad, and Farid Mezian. "A Discourse-Based Approach for Arabic Question Answering." *ACM Transactions on Asian and Low-Resource Language Information Processing*, 16 (2): 1–18, 2016. https://doi.org/10.1145/2988238.

[11] Azmi, Aqil M., and Nouf A. Alshenaif, "Lemaza: An Arabic Why-Question Answering System." *Natural Language Engineering*, 23 (6): 877–903, 2017. https://doi.org/10.1017/S1351324917000304.

[12] Omri, Hajer, Zeineb Neji, Marieme Ellouze, and Lamia Hadrich Belguith. 2017. "The Role of Temporal Inferences in Understanding Arabic Text." *Procedia Computer Science*, 112: 195–204. https://doi.org/10.1016/j.procs.2017.08.228.

[13] Lhioui, Chahira, Anis Zouaghi, and Mounir Zrigui. "A Rule-Based Approach for Arabic Temporal Expression Extraction." *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, Janua: 1–6, 2018, Monastir, Tunisia, (2023, May). https://doi.org/10.1109/ICEMIS.2017.8273114.

[14] Bessaies, Essia, Slim Mesfar, and Henda Ben Ghezala. "Generating Answering Patterns from Factoid Arabic Questions." *In Proceedings of the Linguistic Resources for Automatic Natural Language Generation-LiRA@ NLG*, pp. 17-24. 2017, (2023, June). . https://doi.org/10.18653/v1/w17-3803.

[15] Al-Smadi, Mohammad, Islam Al-Dalabih, Yaser Jararweh, and Patrick Juola.. "Leveraging Linked Open Data to Automatically Answer Arabic Questions." *IEEE Access*, 7 (March), 2019: 177122–36. https://doi.org/10.1109/ACCESS.2019.2956233.

[16] Al-Kabi, Mohammed, Izzat Alsmadi, Rawan T. Khasawneh, and Heider Wahsheh. "Evaluating social context in arabic opinion mining." *Int. Arab J. Inf. Technol.*, 15, no. 6 (2018): 974-982.

[17] Hammoud, Jaafar, Natalia Dobrenko, and Natalia Gusarova. "Named entity recognition and information extraction for Arabic medical text." *In Multi Conference on Computer Science and Information Systems, MCCSIS*, pp. 121-127. 2020, (2023, July), [Online]. https://www.iadisportal.org/digital-library/named-entity-recognition-and-information-extraction-for-arabic-medical-text (Accessed Date: July 1, 2024).

[18] Hammo, Bassam, and Steven Lytinen.. "QARAB: A Question Answering System to Support the Arabic Language." *ACL2002: Computational Approaches to Semitic Languages*, 11, 2002, (2023, April), https://doi.org/10.3115/1118637.1118644.

[19] Al-Ghamdi, S., Al-Khalifa, H. and Al-Salman, A., Fine-Tuning BERT-Based Pre-Trained Models for Arabic Dependency Parsing. *Applied Sciences*, 13(2023, July), p.4225. https://doi.org/10.3390/app13074225.

[20] Albarghothi, Ali, Feras Khater, and Khaled Shaalan.. "Arabic Question Answering Using Ontology." *Procedia Computer Science*, 117,

2017: 183–91. https://doi.org/10.1016/j.procs.2017.10.108.

[21] Torki, Marwan, Maram Hasanain, and Tamer Elsayed. "QU-BIGIR at SemEval 2017 Task 3: Using Similarity Features for Arabic Community Question Answering Forums, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*", 360–64, (2023, July). https://doi.org/10.18653/v1/s17-2059.

[22] Safaya, Ali, Moutasem Abdullatif, and Deniz Yuret.. "KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media." ArXiv:2007.13184 [Cs], July, 2020, (2023, May). http://arxiv.org/abs/2007.13184.

[23] Romeo, Salvatore, Giovanni Da San Martino, Yonatan Belinkov, Alberto Barrón-Cedeño, Mohamed Eldesouki, Kareem Darwish, Hamdy Mubarak, James Glass, and Alessandro Moschitti. "Language Processing and Learning Models for Community Question Answering in Arabic." *Information Processing and Management*, 56 (2), 2019: 274–90. https://doi.org/10.1016/j.ipm.2017.07.003.

[24] Aouichat, Asma, Mohamed Seghir Hadj Ameur, and Ahmed Geussoum.. "Arabic Question Classification Using Support Vector Machines and Convolutional Neural Networks." In Natural Language Processing and Information Systems, edited by Max Silberztein, Faten Atigui, Elena Kornyshova, Elisabeth Métais, and Farid Meziane, 10859:113–25. *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2018, (2023, October). https://doi.org/10.1007/978-3-319-91947-8_12.

[25] Hamza, Alami, Noureddine En-Nahnahi, Khalid Alaoui Zidani, and Said El Alaoui Ouatik.. "An Arabic Question Classification Method Based on New Taxonomy and Continuous Distributed Representation of Words." *Journal of King Saud University - Computer and Information Sciences*, 1–7, 2019. https://doi.org/10.1016/j.jksuci.2019.01.001.

[26] Tang, Haoyang, Meng Qian, Ziwei Sun, and Cong Song.. Visual Question Answer System Based on Bidirectional Recurrent Networks. *Advances in Intelligent Systems and Computing*. Vol. 891. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-03766-6_67.

[27] Mozannar, Hussein, Elie Maamary, Karl El Hajal, and Hazem Hajj, "Neural Arabic Question Answering." *In Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 108–118. Florence, Italy: Association for Computational Linguistics, 2019, (2023, July). https://doi.org/10.18653/v1/W19-4612.

[28] Nozza, Debora, Federico Bianchi, and Dirk Hovy. "What the [mask]? making sense of language-specific BERT models." arXiv preprint arXiv:2003.02912, March, 2020, (2023, December). http://arxiv.org/abs/2003.02912.

[29] Imane Lahbari, Hamza Alami, and Khalid Alaoui Zidani. "Towards a Passages Extraction Method", *Springer International Publishing*. 2020, Morocco, 2019, (2023, Novmber), https://doi.org/10.1007/978-3-030-36653-7.

[30] Ismail, Walaa Saber, and Masun Nabhan Homsi.. "DAWQAS: A Dataset for Arabic Why Question Answering System." *Procedia Computer Science*, 142, 2018: 123–31. https://doi.org/10.1016/j.procs.2018.10.467.

[31] Clark, Jonathan H., Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki.. "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages." ArXiv:2003.05002 [Cs], March, 2020, (2023, November). http://arxiv.org/abs/2003.05002.

[32] Benajiba, Yassine, Paolo Rosso, and José Manuel Gómez Soriano.. "Adapting the JIRS Passage Retrieval System to the Arabic Language." In Computational Linguistics and Intelligent Text Processing, edited by Alexander Gelbukh, 530–41. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springe, 2007r, (2023, September). https://doi.org/10.1007/978-3-540-70939-8_47.

[33] Trigui, Omar, Lamia Hadrich Belguith, and Paolo Rosso. "DefArabicQA: Arabic definition question answering system." In Workshop on Language Resources and Human Language Technologies for Semitic Languages, *7th LREC, Valletta, Malta*, pp. 40-45. 2010, (2023, September). DOI: 10.1109/NLPKE.2009.5313730.

[34] Abouenour, Lahsen, Karim Bouzouba, and Paolo Rosso, "An Evaluated Semantic Query Expansion and Structure-Based Approach for Enhancing Arabic Question/Answering."

International Journal on Information and Communication Technologies, 3 (3): 37–51, 2010, [Online]. https://www.researchgate.net/publication/280253859_An_evaluated_semantic_QE_and_structure-based_approach_for_enhancing_Arabic_QA (Accessed Date: July 1, 2024).

[35] Anselmo Caroline Sporleder, and Eduard H. Hovy Pamela Forner lvaro Rodrigo Richard FE Sutcliffe Corina Forascu Peas. "Neural Arabic Question Answering" *In CLEF (Notebook Papers/Labs/Workshop)*, (2023, October), 2011. DOI:10.18653/v1/W19-4612.

[36] Akour, Mohammed, Sameer O. Abufardeh, Kenneth Magel, and Qasem Al-Radaideh.. "QArabPro: A Rule Based Question Answering System for Reading Comprehension Tests in Arabic." *American Journal of Applied Sciences*, 8 (6): 652–61, 2011. https://doi.org/10.3844/ajassp.2011.652.661.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.