International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Mar-24)

http://dx.doi.org/10.12785/ijcds/150187

Exploring Sentence Embedding Representation For Arabic Question Answering

Imane Lahbari[1] **and Saïd Ouatik El Alaoui** [1]

[1]*Engineering Sciences Laboratory, Ibn Tofail University, Kenitra, Morocco*

Received 26 May 2023, Revised 7 Feb. 2024, Accepted 24 Feb. 2024, Published 1 Mar. 2024

Abstract: Question Answering Systems (QAS) are made to automatically provide accurate response to user questions that are phrased in natural language. Most of the existing QAS adopting traditional representations like word embedding and bag-of-words, have shown promising results. However, only a few works take into account the contextual information and meaning within texts to extract answers from huge sources of information. Moreover, dealing with Arabic open-domain question-answering systems is still challenging due to its rich morphology and ambiguity of words. To address these limitations, we introduce, in this study, a novel QA approach for the Arabic language that is based on passage retrieval and Sentence Embedding (SE) representation. It consists of three steps: (1) Question classification and query formulation, (2) Documents and passages retrieval, and (3) Answers extraction. In this work, we make use of the AraBERT transformer model to compute vector representation. This allows for considering both implicit semantics and the words' context within the text. Furthermore, in order to collect potential passages for user questions, we investigate a method for retrieving Arabic passages using the BM25 model, a query expansion process, and SE representation. The final answer is generated by fine-tuning AraBERT parameters and ranking passages so that the most relevant ones can be extracted. To assess our approach, we carried out several experiments on CLEF and TREC datasets using two different taxonomies. The obtained results show that the proposed method achieves 92% in terms of F1-score.

Keywords: Natural Language Processing, Arabic Question Answering, Word Embedding, Sentence Embedding, AraBERT, Elmo, FastText, Fine-tuning

1. Introduction

With the huge number of textual documents available in electronic formats, classical search engines become unable to satisfy user needs that are expressed as natural questions. These Information Retrieval Systems (IRS) retrieve just a list of documents ordered by their relevance to a given query. Therefore, human intervention is required to retrieve the requested information from the returned documents. Finding the correct information seems to be a complex and expensive task in terms of time consumption. In contrast to the IRS, Question Answering Systems (QAS) are intended to automatically provide direct and accurate responses to user inquiries phrased in natural language.

Question Answering Systems for Arabic open-domain remain a challenging task in Natural Language Processing (NLP). This study aims to propose a novel question-answering approach for the Arabic language that consists of three stages, consisting of question classification, passage retrieval, and answer extraction. We investigate the potential of an Arabic Sentence Embedding (SE) pre-trained model to extract the most valuable features from texts and build the appropriate text representation that considers both contextual information and semantic links between words in the Arabic sentence.

With more than 420 million speakers, Arabic is one of the six official languages of the United Nations and is regarded as one of the most widely spoken languages in the world. Dealing with Arabic Question Answering Systems presents a real challenge regarding its complex morphology (diacritics, dualities, etc.) and its rich vocabulary (more than 10. 000 roots).

Text representation is a crucial process that impacts the effectiveness of most natural language processing tasks such as text summarization, QAS, and information retrieval (IR). Unlike the conventional bag-of-words representation, Word Embedding (WE) such as Word2vec [1], Doc2vec [2], and Glove [3] has revealed to be effective text representation since these pre-trained models can capture the semantic and syntactic relationships between words. Words are represented as dense vectors in low-dimensional vector space. Therefore, words having the same meaning are represented similarly by identical vectors. Although these models have demonstrated good performance, they are unable to take into account the meaning and relationships among multiple words within entire sentences. For instance, consider these

E-mail address: imane.lahbari@uit.ac.ma, ouatikelalaoui.said@uit.ac.ma

two examples:

دخلت الأم الحديقة الخلفية

(The mother entered the back garden) and

لحديقة الأم مدخل خلفي

(The mother's garden has a back entrance.) These two statements will have the same vectors using word embedding representation even though their meanings are entirely different.

To overcome this limitation, Sentence Embedding (SE) models including Elmo [4], BERT [5], and mBERT [6] have emerged as crucial text representation and advanced several NLP tasks. These transformer models provide a more precise depiction of questions and sentences that take into account both context and sentence structure in the case of many languages. SE approaches can represent complete sentences and their semantic information as dense vectors in low-dimensional vector space.

In this paper, we propose an effective method for Arabic question-answering based on sentence embedding representation and passage retrieval. Our system is composed of three main components: (1) question classification and query reformulation, (2) document and passage retrieval, then (3) answer extraction. The main contributions of this work are as follows:

- We investigate the AraBERT [7] pre-trained model to represent both questions and passages. This allows for considering both implicit semantics and the context of words within the text.

- We propose an Arabic passage retrieval module by combining the BM25 model and query expansion process using Arabic Wikipedia.

- We extract the final Answer for different question types by fine-tuning AraBERT [7] parameters based on the text classification task.

- We achieved several evaluations using the standard Arabic corpora Trec and Clef to demonstrate that the AraBERT [7] model performed better than the state-of-the-art representations.

The paper is structured as follows: Related research is shown in Section 2. The suggested approach is explained in Section 3. The experimental techniques and results are covered in Section 4. In Section 5, we provide a conclusion and outline suggestions for future works.

2. Related Work

In this part, we discuss the QAS-related works that have used Wikipedia as a knowledge source and we present some QAS dedicated to the Arabic language. Moreover, we describe the most used sentence embedding representations.

A. Wikipedia And Question Answering Systems

Qakis [8] (2012) is a framework for open-domain question answering over connected data. By leveraging relational textual patterns that are automatically acquired, it addresses the issue of question interpretation by matching question fragments to binary relations in the triple store. The RDF data grouping is located in DBpedia, while the relational patterns are automatically extracted from Wikipedia.

WikiQA [9] (2016)'s authors propose an all-purpose question-answering system that can address why-interrogated queries. This system's knowledge base is Wikipedia's data. Major QA system components like Question Classification, Answer Extraction, Named Entity Tagging, and Information Retrieval have been implemented by WikiQA. Implementing a cutting-edge entity tagging technique has succeeded when tools like OpenEphyra or DBpedia Spotlight have failed to identify entities.

Mindstone [10] (2020), is an open-domain question-answering system, that provides users with replies to their queries based on a sizable library of documents. This quality control system is composed of a brand-new multi-stage pipeline that utilizes a traditional information retriever based on BM25, neural relevance feedback based on RM3, a neural ranker, and a machine reading comprehension stage. This method provides a baseline for end-to-end performance on question answering for the Wikipedia/SQuAD dataset.

B. Arabic Question Answering Systems

The majority of QA research has focused on Latin-based languages. However, there are interesting examples in other languages as well; we give a few Arabic QAS as an example.

The passage retrieval module in DefArabicQA [11] compiles the top-n results returned by the search engine. This specific query is made up of the question topic that the question analysis module determines. Only the top-n snippets that contain the integrated question topic are maintained after gathering the top-n snippets.

Although QArabPro [12] is a rule-based system, it excels at indexing keywords. The solution is then identified using a keyword-matching technique between the question and the response document. A local IR system is used to locate and retrieve the necessary papers.

SOQAL [13] is an Arabic open-domain question-and-answer system that draws its information from Wikipedia. This system's foundation consists of two parts: a document retriever that employs the hierarchical TF-IDF approach, and a neural reading comprehension model that employs the trained BERT [5] bi-directional transformer.

Recently, a few papers introduced deep learning techniques and embedding representations in Arabic QAS like in [14], [15] and [16]. In other research [17], authors use a probabilistic model, and algorithms such as Viterbi to a

contextual spelling correction for the Arabic text.

C. Word and Sentence Embedding

The term "word embedding" describes the introduction of a word's distributional vector form to represent its meaning and grammar. Individual words are represented as real-valued vectors in a predefined vector space, where each word is mapped to a single vector; Words with identical meanings are represented similarly. This method was applied to many languages, like Hindi [18] and Arabic [19].

Sentence embedding (SE) techniques encode complete sentences and their semantic content as vectors as an extension of word embedding. This makes it easier for the machine to understand the context, intention, and other intricacies of the entire text. In the following, we describe the most common sentence embedding representations.

Doc2vec [2] is an algorithm (also known as Paragraph Vector) proposed by Quoc Le and Tomas Mikolov, both research scientists at Google, in 2014. It is built on Word2vec and adheres to the same guidelines for building a machine-learning model that predicts the following word by using the words around it.

One of the first projects to use a pre-trained language model for downstream tasks was ELMo [4] (2018), and it still gives good results [20]. Long Short-Term Memory (LSTM) is used in a two-layer bi-directional architecture, and task-specific weights are used to combine characteristics from all LSTM outputs. It is a generalized classic word embedding study in a different dimension by removing context-sensitive features from a right-to-left and a left-to-right language model. The right-to-left and left-to-right representations of each token are combined to form its contextual representation. In [21], authors have used an LSTM model to investigate the effectiveness of NNs in Arabic NLP. LSTM was combined with convolution neural networks (CNNs) in [22], for Arabic text categorization.

Bidirectional Encoder Representations from Transformers model (BERT) [5] is based on a new pre-training process that will enable the training of a deep bidirectional transformer. BERT improves upon standard Transformers by removing the unidirectionality constraint by using a masked language model (MLM) pre-training objective. The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary ID of the masked word based only on its context. Unlike left-to-right language model pre-training, the MLM objective enables the representation to fuse the left and the right context, which allows us to pre-train a deep bidirectional Transformer. In addition to the masked language model, BERT uses a next-sentence prediction task that jointly pre-trains text-pair representations.

BERT model [5] has been applied in many NLP applications. Its architecture has been adopted in different enhanced models. For instance, in [23] a novel ranking model based on the neural network has adapted a deep BERT model for information retrieval. Other pre-trained Arabic language models have been inspired form BERT's architecture such as AraBERT [7] and marBERT [24].

Recently, a few studies like [25] and [26] have presented comprehensive overviews of Arabic Question Answering Systems. The authors compared the main existing techniques in Arabic QAS and their components using available Arabic and multilingual datasets. In [27], the authors analyzed deep learning methods for question answering in the English language and studied network structure characteristics and their effectiveness. However, Arabic QAS has not been addressed. On the other hand, word embedding models have been widely investigated in QAS as text representation. Specifically, Arabic word embedding representation has been explored and evaluated in [28] using a benchmark containing analogy items. In [29], the authors have exploited word embedding to compute the semantic similarity between terms in Arabic sentences.

More recently, an open-domain QAS using a deep learning approach has been developed in [30]. In this study, a QAS with limited resources has been constructed based on both large language models and a quaternion long-short-term memory neural network (QLSTM). In [31], the authors introduced an Arabic QAS based on a deep learning approach using semantic and logical representations. Arabic texts were converted into conceptual graphs where textual information was represented by concepts and relations.

We notice that many QAS have emerged for different languages in specific domains. Recent QAS adopting word embedding and sentence embedding representations have shown to be effective. However, researchers in the Arabic QAS are still facing limited resources and tools, and are challenged by the Arabic linguistic complexity and richness. Moreover, the number of Arabic QAS studies remains limited, especially in the open domain and only a few works take into account the contextual information and meaning within Arabic texts.

In this work, we investigate a novel approach based on the AraBERT [7] transformer model dealing with Arabic open-domain question-answering systems. Unlike most existing works, our approach consists of considering both implicit semantics and the word context within the Arabic text known for its rich morphology and ambiguity of words.

3. Method

In this section, we will go through each component of our QAS in detail. We start with the question (processing, categorization, and reformulation), then we move on to the retrieval of documents and passages, and ultimately we extract the answers.

A. Question Classification And Query Formulation

The purpose of this component is to develop and classify a query from a user's natural question. Before classifying

TABLE I. Alami et al. Arabic taxonomy

Classes	Explanation
Human, Group	who, whom, whose
Entity, Animal,..	who, whom, whose
Status, Structure,..	how, what
Location	where
Time	when
Number	how many
Yes/No	(directly)

the question, the pre-processing stage is necessary. During this stage, diacritical markings, punctuation, and any other foreign characters are initially eliminated. Excepting stop words, because they could be utilized as interrogative tools. After classifying the questions and identifying the various types of queries, we eliminate them. White space is then used to tokenize the remaining portion of the sentence. Then we apply a lemmatizer and add part-of-speech (POS) tags to our text. While lemmatizing question words, we employ an Arabic lemmatizer that gives each word a unique lemma while also considering the word's context. The lemma of a word is its most fundamental form and communicates its primary meaning. It is used as input for linguistic dictionaries. With part-of-speech (POS) tagging, each word is given a tag that contains various pieces of information (syntactic category, gender, tenses, etc.). The number of passages that must be retrieved is decreased and the retrieving process is more focused when named entities (NEs) are identified using the POS tagger. As the group of nouns in an Arabic sentence determines the meaning of the sentence rather than the verbs, we also utilize the POS tagger to remove the verbs from each question.

Information can be categorized using taxonomies in a structured architecture. To establish the question type, a machine learning approach is applied to categorize questions depending on two taxonomies. We use two different taxonomies to categorize our queries. The first is proposed by Alami et al. [32], it contains seven categories that regroup all question types used in Arabic and is retrieved by studying the Arabic interrogation procedures, table I. The second is Li and Roth's taxonomy [33], and it is based on the answer type's semantic interpretation, table II.

To classify Arabic questions, three different classification algorithms were examined in one of our earlier works [34]: a decision tree, a naïve Bayesian, and a support vector machine (SVM); the latter method offered the best classification ranks. In this study, we adopt SVM to classify our queries.

After pre-processing, lemmatizing, POS tagging, and classifying the question, our queries are now created and ready to include the next component: Documents and Passages Retrieval.

TABLE II. Li and Roth taxonomy

Cross classes	Fine classes
DESCRIPTION	description, reason, manner, ,definition
ENTITY	product, language, colour, plant, instrument, event, food, creative, religion, technique, currency, substance, sport, symbol, disease/medicine, body, animal, term, vehicle, word, other
HUMAN	title, individual, group, description
NUMERIC	count, code, date, period, distance, order, money, per cent, temp, speed, volume, weight, size, other
LOCATION	city, country, mountain, other, state
ABBREVIATION	expression, abbreviation

B. Documents And Passages Retrieval

In this section, we'll go through two sorts of retrieval methods for retrieving the best passages: IR and IR employing sentence embedding. At first, we used the Google API to retrieve Arabic Wikipedia documents before using the traditional BM25 approach to retrieve passages. By using sentence embedding techniques to extract the top-ranked passages, we lower the quantity of those passages at the second level.

1) Information Retrieval

Before using the embedding representation, two information retrieval (IR) sub-phases are recommended, as indicated in the flowchart below: Google API and BM25, Figure 1.

In the first IR level, we use the Google API to match the titles of Wikipedia papers with our created query to retrieve the pertinent Arabic Wikipedia documents. The top M-ranked documents are retained. We use the identical prepossessing techniques for those documents as we did for the questions, and then we divide them into equal-length portions (T tokens in each passage).

At this stage, each recovered document is presented by a group of passages. To extract the candidate passages, we combine the entire passages into one new corpus.

As a second level of information retrieval, The top candidate passages are extracted from the entire corpus of passages using the BM25 model. Formally, the score of a passage P given with a query Q which includes the words $q_1, ..., q_n$ is given by:

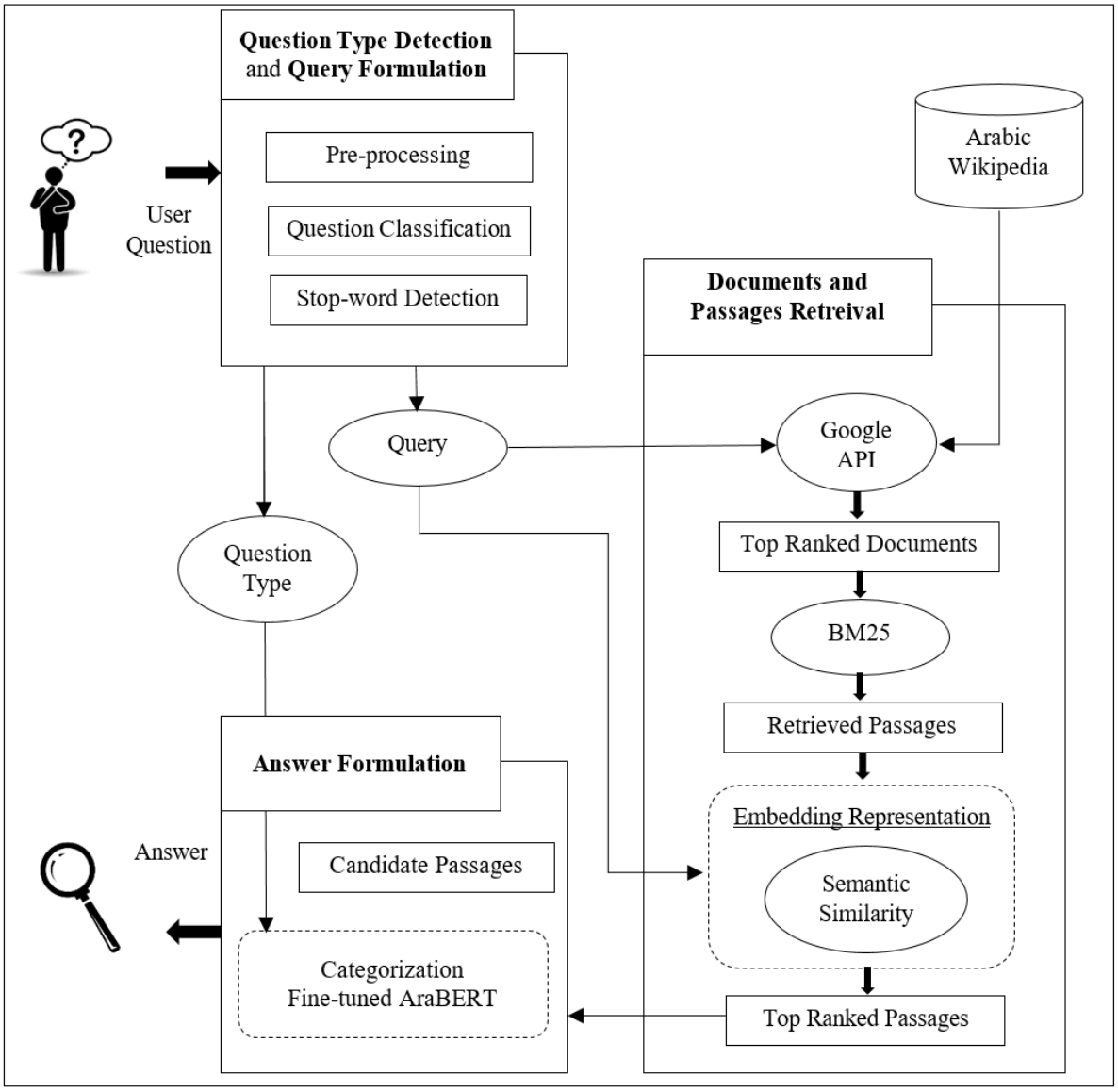

Figure 1. The flowchart of our proposed method

$$score(P,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,P)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|P|}{avgdl})} \quad (1)$$

where $f(q_i, P)$ is the frequency of the term q_i in the passage P, $|P|$ is the passage's P length in words, and $avgdl$ is the length of a passage on average in the text collection from which passages are taken. k_1 and b are free parameters.

The output from this sub-component is a collection of passages that include the final answer.

2) IR Using Sentence Embedding Representation

The usefulness of SE representations for NLP applications, particularly for IR and QAS, has been demonstrated previously in this paper. The top-ranked passages from the passages retrieved by BM25 are obtained in this section using an Arabic passage retrieval module based on embedding sentences.

The basic concept of this paper is that we use AraBERT [7] to get answers from a set of retrieved passages, not from random text collections.

BERT [5] is based on transformers, an architecture that aims to solve problems sequentially while handling long-distance relationships with ease. The BERT [5] transformer

uses bidirectional self-attention as opposed to the preceding transformers' limited self-attention, which only allows each token to attend to the context to its left. All computations of the input and output representations rely on self-attention.

In AraBERT [7], authors trained this model with a large Arabic dataset with articles from all over the Arabic region, retaining the same architecture of BERT-base (encoder layers, attention heads, hidden units and parameters).

We use the retrieved passages in their original form (before any NLP treatment) because AraBERT [7] contains a preprocessing step.

To create vector representations for queries and the passages they are associated with (which are retrieved with the BM25 model), we use pre-trained AraBERT models. We also apply pre-trained models from Elmo [4] and FastText [35], which is a word embedding method, to improve the comparative study.

Generalizing the concept of cosine similarity, soft similarity, called also soft cosine, between two vectors q and p, takes into account similarities between pairs of each vector's features [36]. In our last research [19], when we investigated several semantic similarity measures for embedding vectors, soft cosine similarity delivered the best outcomes. The soft cosine measure additionally considers the similarity of characteristics in the Vector Space Model (VSM), whose features are seen as separate or independent from the standard cosine similarity.

The soft cosine similarity can be determined using two N-dimensional vectors, q and p, as follows:

$$softcosine(\mathbf{q}, \mathbf{p}) = \frac{\sum_{i,j}^{N} s_{ij}\mathbf{q_i}\mathbf{p_j}}{\sqrt{\sum_{i,j}^{N} s_{ij}\mathbf{q_i}\mathbf{q_j}}\sqrt{\sum_{i,j}^{N} s_{ij}\mathbf{p_i}\mathbf{p_j}}} \quad (2)$$

based on the WE model, N is the dimension of the vectors and $s_{i,j} = sim(f_i, f_j)$ is the similarity between features.

We determine the similarities between each query and the passages that are connected to it using soft cosine similarity, and then we extract the top-ranked passages from the retrieved passages by the BM25 model at the first level.

C. Answer Extraction

Finding the right response from the top-ranked passages returned in the previous component is still a difficult task. To overcome this issue and obtain a definitive solution, we suggest using fine-tuned AraBERT [7]. We choose to fine-tune AraBERT [7] to obtain the final answer because, in our experience, AraBERT produces the greatest results when compared to FastText [35] or Elmo [4]. Additionally, authors demonstrate in [32] the efficiency of the enhanced AraBERT [7] model in categorizing Arabic text. They also examine AraBERT's effectiveness as a feature extractor model by integrating it with different classifiers, including SVM.

The fundamental idea behind optimizing AraBERT [7] is to fine-tune all the parameters on a downstream task after pre-training deep bidirectional representations from the unlabeled text that take context from both sides. Two of the techniques used are next-sentence prediction and masked language modeling.

The tokenization is a fundamental step before fine-tuning AraBERT [7], one vector should represent the whole input sentence to be fed to the classifier. For that, the choice is made to interpret the first token [CLS]'s hidden state as representing the entire sentence. However, BERT [5] needs to know where the first sentence ends and the second passage begins to perform the "next sentence prediction" process. As a result, [SEP] is the token used. The special token [CLS] is added to the beginning of each sentence during the tokenization process, and the special token [SEP] is added in between sentences at the end.

We tokenize the input texts and include [CLS] and [SEP] tokens to help AraBERT [7] models be fine-tuned. Then, we create an input representation for every token by adding up the vector embeddings associated with the token, its corresponding segment, and its position. After that, we feed the AraBERT [7] models these representation vectors and fine-tune them. As the question representation, we use the first [CLS] token as a final hidden state. In order to obtain the probability distribution over the anticipated output label, we use a feed-forward layer to normalize the obtained vectors.

The Transformer's self-attention mechanism makes it uncomplicated to fine-tune AraBERT [7] because it enables it to simulate a variety of downstream tasks by changing the right inputs and outputs. We train the model to alter only the weights of the top layers because they have a higher level of task knowledge while the language understanding is in the bottom layers to fine-tune AraBERT on a QA task. Also, we replace the embedding of the [CLS] (classification) token with the class of the question which is retrieved at the beginning of the process using SVM.

In our scenario, we add the vector embedding of each token in a passage to generate an input representation. After that, we feed the vectors to AraBERT [7] and modify its settings using the corpus.

The following algorithm, Figure 2, describes the entire process of the suggested method, including all earlier stages.

4. Experimental Results

To evaluate the performance of the proposed approach for Arabic QAS, we conduct various experiments on two datasets including CLEF and TREC using two different taxonomies. Sub-section A describes the used Arabic datasets and employed metrics to assess our approach. All experi-

Algorithm : Pseudo code of the proposed method

```
Input: Question (a set of terms separated by spasees)
Output: Answer (a set of terms separated by spasees)

SWL: list of stop words, Passages_list: list of passages, Docs : list of documents,
embeddingModel : sentence embedding model, embedPassg : array of N passages
Begin
                                        /* Question Type and Query Formulation */
For each term in Question do
  { If (term not Arabic) then delete term from Question
    Else Add term to Query
  }
   queryType ← getClass_SVM(Query.terms)          // Classification by taxonomies, using the SVM classifier
  End If    End for
For each term in Query do
   {  tag ← get.POStag (term)
     If (tag == "verb" or tag ∈SWL)  then delete term from Query          // delete verbs and stop words
     Else Add term to Query
  }
End For
                                        /* Documents and Passages Retrieval */
Docs ← GoogleAPI(Query, Arabic_Wikipedia)
Passages_list ← split(Docs, T)                  // T: nember of terms in each passage
Top_retreived_passages ← BM25 (Passages_list, Query)
embedPassg ←  embeddingModel.embedSentences (Top_retritived_passages)
embedQuery ← embeddingModel.embedSentences (Query)
For i = 1 to N
{semanticSimilarityScores[i] ← soft_cosine_similarity(embedPasseg[i], embedQuery)}
End For
 topIndices[k] ← getTopScors(semanticSimilarityScores[N],k)
 candidatePassages ← embedPassg[topIndices]
                                        /* Question Type and Query Formulation */
AraBertModel ← AralertEmbeddingModel(Parameters)
classificationModel ← AraBertModel( SVM )       // get the class of each  candidate passage by a supervised learning
classifiedPassages ← classificationModel ( candidatePassages, taxonomy )
do { Answer ← classifiedPassages(k).getPassage() }
while ( classifiedPassages[k].getClass() == queryType

End
```

Figure 2. The algorithm of our proposed method

ments and obtained results are detailed in sub-section B.

A. Dataset And Evaluation Metrics

To evaluate the effectiveness of our method, we use a collection of 2300 questions extracted from the Cross-Lingual Evaluation Forum (CLEF) and Text Retrieval Conference (TREC). This dataset was translated from English to Arabic and was neither categorized nor annotated where most available text-mining datasets have been focused on Latin-based languages. Moreover, the initial dataset has been annotated and classified to create our own final dataset used in experiments.

We employ a variety of Python packages in our experiments, such as the open-source machine learning platform TensorFlow, the machine learning package Scikit-learn, Gensim for topic modeling and text similarity, and the PyTorch interface for the BERT model.

To show the performance of AraBERT [7] adopted by our method, we additionally make use of FastText [35] and Elmo [4] models for the evaluation and comparison with existing methods. The dataset that is utilized to pre-train each of these three models is described in the next paragraphs.

Our method is based on AraBERT [7], and for the comparison analysis, we additionally make use of FastText [35] and Elmo [4]. The dataset that was utilized to pre-train each of those three models is described in the paragraphs that follow.

Arabic news websites manually scraped the used dataset for AraBERT [7] in search of articles. Additionally, two

sizable Arabic corpora that were openly accessible were used: (1) The cite-Corpus, 1.5 billion words of Arabic text are found in a contemporary corpus of over five million news items culled from ten major news organizations spanning eight nations. (2) OSIAN: the International Arabic News Corpus Open Source (approximately 1 billion tokens), which gathers 3.5 million articles from 24 Arab countries and 31 news sources. The final pre-training data for the AraBERT model is 77GB, which is equivalent to 82,232,988,358 characters or 200,095,961 lines, 8,655,948,860 words (before applying the Farasa [37] segmentation tool).

ELMO [4] used the following sources to train its embedding model: Tweets, the Arabic Wikipedia, Mawdoo3 articles, and so on. 68,400,000 distinct tweets were gathered using the Twitter API's default settings. After being downloaded, Wikipedia articles were divided up into sentences using punctuation. The articles could be divided into 4,600,000 sentences. An encyclopedia that offers articles in Arabic is called Mawdoo3. The Mawdoo3 entries were divided into sections, much like Wikipedia articles, resulting in 2,800,000 sentences.

With FastText [35], authors make word vectors that have already been trained on Wikipedia available in 294 different languages. These 300-dimensional vectors were acquired using the skip-gram model. The length of these vectors is referred to as their "dimensionality" in embeddings, which denotes the total number of features that encode the vector (2D vector representation example).

Our proposed method for answering Arabic questions is evaluated using three metrics including recall (R), precision ($F1$), and F1-score ($F1$), and averaging them over the number of questions in the corpus. These metrics are calculated as follows:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F1 = 2 * \frac{P * R}{P + R} \quad (3)$$

The number of potential solutions that are mentioned in both the returned list and the golden list is indicated by TP; The number of solutions that are mentioned in the returning list but not on the golden list is represented by FP; and the number of solutions listed in the golden list but absent from the list that was returned is indicated by FN.

B. Results And Discussion

To demonstrate the effects of employing transfer learning from the pre-trained models in our QAS, we have carried out extensive testing.

Before classifying questions, we first remove punctuation, diacritics, and any other foreign characters as a preprocessing step. Stop words may be used as an interrogative tool, so we do not remove them before classifying the questions and identifying the different types of queries. Then, we perform white-space tokenization on the remaining sentence. Using the tools provided by the SAFAR platform [38], we apply part-of-speech (POS) tagging to attribute tags to each question word, and we also use the POS tagger to recognize NEs. After classifying the questions using the SVM classifier and identifying the various types of queries, we eliminate stop words. In Arabic, nouns not verbs, which carry the meaning of a phrase, for this reason, we also use the POS Tagger to take the verbs out of each query. Two layers of information retrieval (IR) are proposed to extract the top-ranked passages: IR using traditional methods and IR using sentence embedding.

In the first IR level, we use the Google API to match the titles of Wikipedia papers with our created query (2300 queries) to retrieve the pertinent Arabic Wikipedia documents. We retrain the top 10-ranked documents, and we use identical pre-possessing techniques for those documents that were used for the questions. We divide each document into equal-length passages (100 tokens in each passage). At this stage, each recovered document is displayed as a series of passages, the number of which depends on the length of the document. We combine the entire passages (from the 10 retrieved documents) into one new corpus, to extract the candidate passages. As a means to reduce the number of retrieved passages, we apply the BM25 model and extract only the top 100 candidate passages from the whole set of retrieved passages.

In the second level, we adopt AraBERT [7] pre-trained models to compute similarities between vector representations for queries and their related passages; besides, we used pre-trained models from Elmo [4] and FastText [35] for comparing the results.

The main idea of this paper is that we use AraBERT [7] to get answers from a set of passages, which are relevant and contain the right answer.

AraBERT [7] is a Bidirectional Encoder Representation from Transformers, and the Transformer is an architecture that seeks to handle long-distance dependencies and solve tasks in sequence-to-sequence, for that reason we keep 100 tokens in each passage. To create vector representations for queries and the passages they are associated with (which are retrieved with the BM25 model), we use the pre-trained AraBERT [7] model because it's trained with a large Arabic dataset (about 31 GB of text). We use the retrieved passages in their original form (without any NLP treatment) because AraBERT [7] contains its own preprocessing step.

Using the vector representations, we first determine the soft cosine similarity between each query and the passages that are connected to it, and then we choose the top-ranked 20 passages.

To find the right response from the top-ranked passages, we propose fine-tuning AraBERT [7] parameters to classify those passages and extract the final answer.

Figure 3. Obtained Recall results using Alami et al. taxonomy

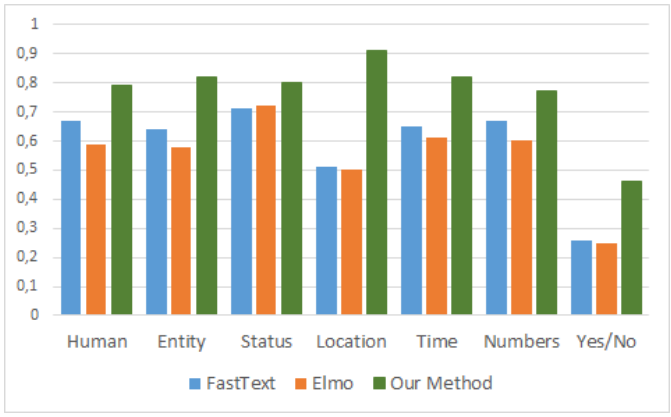

Figure 4. Obtained Precision results using Alami et al. taxonomy

We train the model to change the weights of the top layers, which have a higher level of task knowledge, while the language understanding is in the bottom layers, to fine-tune AraBERT [7] on a QA task. The first input of the AraBERT [7] model is the embedding of the classification token (CLS), which we replace with the class of the question which is retrieved in the first component using SVM.

Tables III and IV report the metric values regarding the use of each embedding model. It gives us an overview of the performance of our system within each type of question and regarding two taxonomies. We can see that our method, which is based on AraBERT [7], yields the highest scores, with an F1-score up to 0.92 for the ”Location” question type (QT) in both taxonomies. Even though FastText [35] (a word embedding representation method) it gives better results than Elmo [4], which is a sentence embedding representation method.

In Figures 3, 4, 5, and 6 we compare the recall and the precision results of the three applied methods. With the Alami and al. [32] taxonomy, the recall results are very close to the progress of our method; With Li and Roth [33] taxonomy, recall results are almost similar for ”Description” and ”Numeric” QTs, Figure 5, but FastText [35] gives better recall results for ”Abbreviation”, ”Humain” and ”Numeric” QTs. However, our method gives the best precision for all QTs, Figure 6.

In Figures 7 and 8 we compare the F1-score of the three used models according to the two taxonomies. With both of them, our method gives the highest results. In Figure 7, by applying Alami and al. [32] taxonomy, the three methods give almost the same results for the ”Status” question type; even for ”Numbers” QT, our method and FastText [35] give similar results. Moreover ”Yes/No” questions are not well answered by the three methods. In the second, Figure 8, when we apply Li and Roth [33] taxonomy, the three models give the same results for ”Description” QT, but ”Abbreviation” and ”Numeric” QT are not well answered.

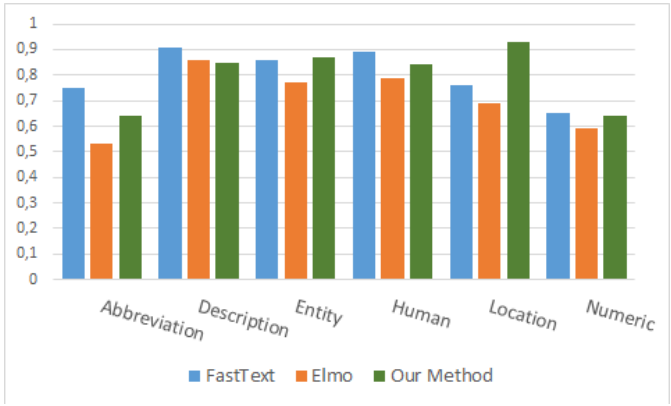

Figure 5. Obtained Recall results using Li and Roth taxonomy

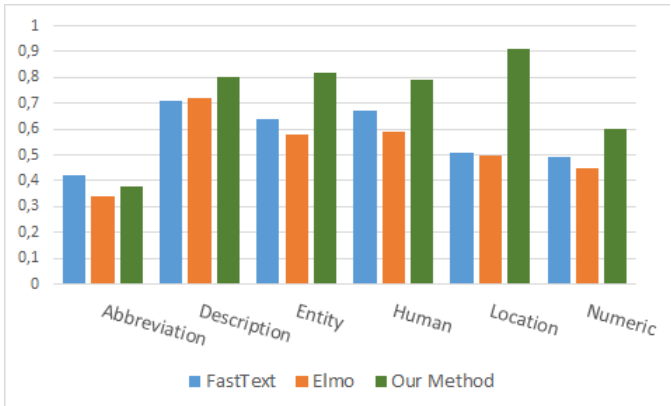

Figure 6. Obtained Precision results using Li and Roth taxonomy

TABLE III. Results with Alami et al. taxonomy

	FastText			Elmo			Our Method		
	Recall	Precision	F1-score	Recall	Precision	F1-score	Recall	Precision	F1-score
Human	**0,89**	0,67	0,76	0,79	0,59	0,68	**0,84**	0,79	**0,81**
Entity	**0,86**	0,64	0,73	0,77	0,58	0,66	**0,87**	**0,82**	**0,84**
Status	**0,91**	0,71	**0,80**	**0,86**	0,72	0,78	**0,85**	**0,8**	**0,82**
Location	0,76	0,51	0,61	0,69	0,5	0,58	**0,93**	**0,91**	**0,92**
Time	**0,85**	0,65	0,74	0,79	0,61	0,69	**0,87**	**0,82**	**0,84**
Numbers	**0,89**	0,67	0,76	0,78	0,6	0,68	**0,83**	0,77	**0,80**
Yes/No	0,38	0,26	0,31	0,34	0,25	0,29	0,47	0,46	0,46

TABLE IV. Results with Li and Roth taxonomy

	FastText			Elmo			Our Method		
	Recall	Precision	F1-score	Recall	Precision	F1-score	Recall	Precision	F1-score
Abbreviation	0,75	0,42	0,54	0,53	0,34	0,41	0,64	0,38	0,48
Description	**0,91**	0,71	**0,80**	**0,86**	0,72	0,78	**0,85**	**0,80**	**0,82**
Entity	**0,86**	0,64	0,73	0,77	0,58	0,66	**0,87**	**0,82**	**0,84**
Human	**0,89**	0,67	0,76	0,79	0,59	0,68	**0,84**	0,79	**0,81**
Location	0,76	0,51	0,61	0,69	0,5	0,58	**0,93**	**0,91**	**0,92**
Numeric	0,65	0,49	0,56	0,59	0,45	0,51	0,64	0,6	0,62

Figure 7. Obtained F1-score results using Alami et al. taxonomy

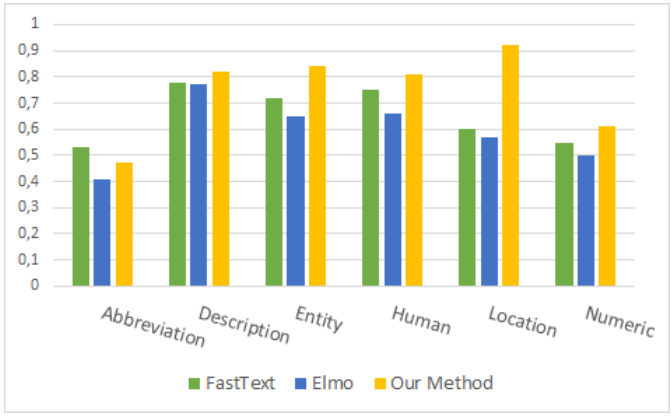

Figure 8. Obtained F1-score results using Li and Roth taxonomy

According just to F1-score results (tables 7 and 8), our method shows the best results compared to the other methods. Regarding Li and Roth [33] taxonomy, our method, which is based on AraBERT [7], gives the best results, except for the abbreviation QT. FastText [35], which is a word embedding method, is in the second rank, and the lowest results are given by Elmo [4]. Using Alami and al. [33] taxonomy, our method gives the best results for all QTs. However, FastText [35] and Elmo [4] provide almost the same lower results. We notice that the "Abbreviation" QT gets the lower results in the second taxonomy, regarding the three used methods, which is normal because "Abbreviation" is not determined in classical Arabic language, it's a foreign shortened form of a word, and it's used in modern Arabic.

The next table (Table V), gives us a few examples of the answered questions from the used dataset. We notice that the "yes/no" QT is still answered by a whole text (as in the last example).

TABLE V. Example of some retrieved answers

Question from the corpus	Type	Retrieved Answer
من الذي اكتشف امريكا؟ Who discovered America?	Human	ابحر كريستوف كولومبوست لاول مره في المحيط الاطلسي بثلاثه سفن تحمل العلم الاسباني وب ١٢٠ بحارا وذلك انطلاقا من بلده بالوس
اين تقع مدينه الاقصر؟ Where is Luxor located?	Location	محافظه الاقصر هي محافظه مصريه تقع في اقليم جنوب الصعيد وتتوزع مراكزها او مدنها على ضفتين نهر النيل وعاصمتها هي مدينه الاقصر
متى اخترع تيليغراف؟ When was the telegraph invented?	Time	في عام ١٨١٠ اخترع العالم والكهربائي التقني الامريكي صامويل مورس التلغراف الذي يعيد طباعه الاحرف
كم تبلغ سرعه الضوء؟ What is the speed of light?	Number	سرعه الضوء في الفراغ هي ثابت فيزيائي هام عُفي العديد من المجالات الفيزيائيه يرمز له في العاده بالرمز ستساوي قيمته ٢٩٩٧٩٢٤٥٨ متر لكل ثانيه
ماذا تعني ناسا ؟ What does NASA mean?	Description	الاداره الوطنيه للملاحه الجويه والفضاء هي وكاله تابعه لحكومه الولايات المتحده الامريكيه وتختصر ناسا اصا
ماذا يقيس البارومتر؟ What does a barometer measure?	Entity	البارومتر او مقياس الضغط الجوي جهاز لقياس الضغط الجوي
هل قمه افرست اعلى جبل على الارض؟ Is Mount Everest the highest mountain on Earth?	Yes/No	جبل افرست (بالتيبيتية شومو لانغما وبالنيباليه ساجار ماثا) أعلى جبل على وجه الارض حيث يرتفع حوالي ٩ كلم عن سطح البحر

5. Conclusion And Future Work

In this study, we suggested an open-domain method using passage retrieval and sentence embedding representation for answering Arabic questions. First, we classified the user question and formulated the query using machine learning techniques. Then, we extracted the related passages of the generated query from the selected Wikipedia documents by using both the pre-trained model AraBERT as text representation and the BM25 as information retrieval model. Top passages were then extracted from the retrieved passages by combining the BM25 model and query expansion process. Finally, we provided a suitable answer to the user's question by applying the fine-tuning to AraBERT parameters on the text classification task. Our method can generate specific and correct answers for different question types using two taxonomies, including Li and Roth, and Alami et al.. Moreover, investigating the AraBERT transformer model to represent both questions and passages allows us to consider implicit semantics and the context of words within the text. To evaluate the proposed approach, we have conducted many experiments using TREC and CLEF datasets. The obtained results demonstrate the effectiveness of our approach achieving up to 0.92% in terms of F1-score.

Despite the relevance of obtained results in terms of retrieved answers, our system still has some limitations in dealing with only closed-ended questions such as "yes/no" question type but not addressing the case of disjunctive questions such as:

أليس كذلك ؟

(Is not it?). A second limitation is linked to the response time at the level of our information retrieval step to find relevant documents and candidate passages. These two limitations will be addressed in future work. Specifically, the search performance can be improved by integrating hierarchical K-Means clustering in the information retrieval component. We also intend to apply various translation techniques to build a cross-language QAS.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *International journal on Learning Representations*, 2013.

[2] L. Q.V. and M. T., "Distributed representations of sentences and

documents," *the 31st International journal on Machine Learning*, pp. 1188–1196, 2014.

[3] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," *Proceedings of the 2014 journal on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Oct. 2014. [Online]. Available: https://aclanthology.org/D14-1162

[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *Proceedings of the 2018 journal of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, Jun. 2018. [Online]. Available: https://aclanthology.org/N18-1202

[5] L. K. Devlin J., Chang M.W. and T. K. ., "Bert: Pre-training of deep bidirectional transform- ers for language understanding," *the 2019 journal of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, p. 4171–4186, 2019.

[6] J. Libovický, R. Rosa, and A. Fraser, "How language-neutral is multilingual bert?" 2019.

[7] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," *the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, p. 9–15, 2020.

[8] E. Cabrio, J. Cojan, A. P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon, "Qakis: An open domain qa system based on relational patterns," *Proceedings of the 2012th International journal on Posters & Demonstrations Track - Volume 914*, p. 9–12, 2012.

[9] F. Abbas, M. K. Malik, M. U. Rashid, and R. Zafar, "Wikiqa — a question-answering system on wikipedia using freebase, dbpedia and infobox," *2016 Sixth International journal on Innovative Computing Technology (INTECH)*, pp. 185–193, 2016.

[10] S. J. Semnani and M. Pandey, "Revisiting the open-domain question answering pipeline," *ArXiv*, vol. abs/2009.00914, 2020.

[11] O. Trigui, L. H. Belguith, and P. Rosso, "Defarabicqa: Arabic definition question answering system," 2010.

[12] M. Akour, S. Abufardeh, K. I. Magel, and Q. A. Al-Radaideh, "Qarabpro: A rule based question answering system for reading comprehension tests in arabic," *American Journal of Applied Sciences*, vol. 8, pp. 652–661, 2011.

[13] D. Kiyasseh, T. Zhu, and D. A. Clifton, "Soqal: Selective oracle questioning for consistency based active learning of cardiac signals," 2022.

[14] H. Alami, A. El Mahdaouy, A. Benlahbib, N. En-Nahnahi, I. Berrada, and S. E. A. Ouatik, "Daqas: Deep arabic question answering system based on duplicate question detection and machine reading comprehension," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101709, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S131915782300263X

[15] A. Hamza, S. E. Alaoui Ouatik, K. A. Zidani, and N. En-Nahnahi, "Arabic duplicate questions detection based on contextual representation, class label matching, and structured self attention," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part B, pp. 3758–3765, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157820305735

[16] A. Hamza, N. En-Nahnahi, A. El Mahdaouy, and S. El Alaoui Ouatik, "Embedding arabic questions by feature-level fusion of word representations for questions classification: It is worth doing?" *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6583–6594, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157822000994

[17] S. Laaroussi, A. Yousfi, S. Aouragh, and S. Ouatik El Alaoui, "Global spelling correction in context using language models: Application to the arabic language," *International Journal of Computing and Digital Systems*, vol. 13, pp. 361–370, 01 2023.

[18] P. Tripathi, P. Mukherjee, M. Hendre, M. Godse, and B. Chakraborty, "Word sense disambiguation in hindi language using score based modified lesk algorithm," *International Journal of Computing and Digital Systems*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225044170

[19] I. Lahbari and S. O. E. Alaoui, "Exploring semantic similarity measure based on word embedding representation for arabic passages retrieval," *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)*, pp. 978–989, 2022.

[20] M. Hendre, P. Mukherjee, R. Preet, and M. Godse, "Efficacy of deep neural embeddings-based semantic similarity in automatic essay evaluation," *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 17, pp. 1–14, 01 2023.

[21] K. Alrajhi and M. Elaffendi, "Automatic arabic part-of-speech tagging: Deep learning neural lstm versus word2vec," *IJCDS Journal*, vol. 8-3, p. 9, 05 2019.

[22] F. Alami, S. Ouatik El Alaoui, and N. Ennahnahi, "Deep neural models and retrofitting for arabic text categorization," *International Journal of Intelligent Information Technologies*, vol. 16, pp. 74–86, 04 2020.

[23] O. Khattab and M. Zaharia, "Colbert: Efficient and effective passage search via contextualized late interaction over bert," *SIGIR '20: The 43rd International ACM SIGIR journal on research and development in Information Retrieval*, pp. 39–48, 07 2020.

[24] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7088–7105, Aug. 2021. [Online]. Available: https://aclanthology.org/2021.acl-long.551

[25] T. H. Alwaneen, A. M. Azmi, H. Aboalsamh, E. Cambria, and A. Hussain, "Arabic question answering system: a survey," *Artificial Intelligence Review*, vol. 55, pp. 207–253, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237829190

[26] S. Yassine and M. Gammoudi, "A comprehensive review of arabic question answering datasets," *Neural Information Processing, ICONIP, Communications in Computer and Information Science*, vol. 1961, pp. 278–289, 11 2023.

[27] T. Hao, L. Xinxin, Y. He, F. L. Wang, and Y. Qu, "Recent progress in leveraging deep learning methods for question answering," *Neural Computing and Applications*, vol. 34, pp. 1–19, 01 2022.

[28] S. Yagi, A. Elnagar, and S. Fareh, "A benchmark for evaluating arabic word embedding models," *Natural Language Engineering*, vol. 29, no. 4, p. 978–1003, 2023.

[29] B. Dahy, M. Farouk, and K. Fathy, "Arabic sentences semantic similarity based on word embedding," *20th International journal on Language Engineering (ESOLEC), Cairo, Egypt*, pp. 35–40, 10 2022.

[30] A. Rahaman Wahab Sait and Y. Alkhurayyif, "Developing an open domain arabic question answering system using a deep learning technique," *IEEE Access*, vol. 11, pp. 69 131 – 69 143, 07 2023.

[31] S. Kamel, S. Hassan, and L. Elrefaei, "Vaqa: Visual arabic question answering," *Arabian Journal for Science and Engineering*, vol. 48, 03 2023.

[32] F. zahra El-Alami, S. Ouatik El Alaoui, and N. En Nahnahi, "Contextual semantic embeddings based on fine-tuned arabert model for arabic text multi-class categorization," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part A, pp. 8422–8428, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821000446

[33] X. Li and D. Roth, "Learning question classifiers," *COLING 2002: The 19th International journal on Computational Linguistics*, 2002. [Online]. Available: https://aclanthology.org/C02-1150

[34] I. Lahbari, S. O. E. Alaoui, and K. A. Zidani, "Toward a new arabic question answering system," *Int. Arab J. Inf. Technol.*, vol. 15, pp. 610–619, 2018.

[35] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. [Online]. Available: https://aclanthology.org/Q17-1010

[36] G.-A. H. Sidorov G., Gelbukh A. and P. D., "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Computacion y Sistemas*, vol. 18, p. 491–504, 2014.

[37] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," *Proceedings of the 2016 journal of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 11–16, Jun. 2016. [Online]. Available: https://aclanthology.org/N16-3003

[38] "Software architecture for arabic," *http://Arabic.emi.ac.ma/safar/.*

Imane Lahbari ; I'm a PhD candidate at Kenitra's Engineering Sciences Laboratory at Morocco's Ibn Tofail University.

In 2015, I graduated with a master's in Information Systems, Networks, and Multimedia. Natural language processing, information retrieval, and question-answering systems are some of my areas of research interest.

Said Ouatik El Alaoui has been a professor since 1997. At the National School of Applied Sciences at Ibn Tofail University, Kenitra in Morocco, he currently holds the positions of Professor and Head of the Engineering Sciences Laboratory.

Image Retrieval, and High-dimensional indexing and Content-Based and Information Retrieval, Natural Language Processing, Arabic Document Clustering and Categorization, and Biomedical Question Answering are some of his current research interests.