

# Arabic\_QA\_System\_Alaa\_Jamila \_Hala

*by* 1200001-Alaa Saleem

---

**Submission date:** 22-Jun-2024 12:54PM (UTC+0400)

**Submission ID:** 2406658438

**File name:** 117800\_1200001-Alaa\_Saleem\_Arabic\_QA\_System\_Alaa\_Jamila\_Hala\_383200\_1821450935.pdf  
(410.69K)

**Word count:** 1805

**Character count:** 9418



Electrical and Computer Engineering Department

ENCS5342: "Information Retrieval, Web Search and NLP"

Course Project: Arabic Question Answering

Alaa Saleem 1200001

Jamila Fawaqa 1200435

Hala Gholeh 1201418

## ABSTRACT

The goal of this project is to develop an Arabic Question Answering (QA) system using Natural Language Processing (NLP) techniques. The system will use a diverse dataset of Arabic web documents as its knowledge source we choose a dataset from "Masader" website called DAWQAS. The system will be accessible through a user-friendly interface, allowing Arabic speaking users to input questions via text and receive clear answers, thereby enhancing access to information. The system will undergo data preprocessing (including tokenization, stemming, lemmatization, and removing stop words) to refine input text quality. Additionally, it integrates question analysis and information retrieval mechanisms to rank documents based on their relevance to the input query. Answer extraction utilizes state-of-the-art techniques such as BERT-based named entity recognition (NER) and rule-based approaches. The evaluation will be done using standard measures including F1-score, recall, and precision.

## 1. INTRODUCTION

For several years, Question Answering systems have been considered an important area of natural language processing. The main goal of a

question answering system is to create a system capable of accurately answering questions posed based on a large set of documents (corpus).

There are two basic types of question answering systems:

- Generative QA: In this type, the system generates an answer to the question posed by the user even if the question text is not directly mentioned in the document text.
- Extractive QA: This type relies on extracting relevant excerpts from the document text directly to answer the user's question.<sup>[1]</sup>

In this project, extractive QA was adopted in designing the system due to several advantages, including: the realistic accuracy provided by this type because the answers are extracted directly from the text, the efficiency of this type; as extracting the answer is faster than generating it, and the models used in this type are smaller than those used in Generic QA, in addition to its ability to deal with large amounts of texts. Despite the many advantages of the extractive QA, it faces some challenges, the most important of which is its inability to provide answers that are not directly present in the text,

and the efficiency of the answers depends on the source of the documents used by the system.

Arabic language is one of the most widespread languages in the world and has unique characteristics and features that pose challenges to tasks related to NLP and IR. The most important of these challenges are: diacritics, the difference in some terms in classical Arabic and colloquial Arabic, and different dialects according to the geographical region. In this project, we seek to overcome these challenges by creating an extractive Arabic question answering system that has the ability to extract an answer from a large group of documents (corpus).

## 2. METHODOLOGY

### 2.1 Data collection

QID	Site_id	Question	Document
13	14	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...
14	15	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...
15	16	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...
16	17	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...
17	18	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...
18	19	سأذكر لكم في هذا الفيديو...	فيديو تعليمي...

Figure 1: A snapshot from our chosen dataset "DAWQAS" shows the format of the dataset

We chose a dataset from the "Masader" website called DAWQAS, short for "Dataset for Arabic Why Question Answering System"<sup>[2]</sup>. The dataset contains 3209 documents on various topics in the Arabic language. For each document, there is a sample question with an answer to the question (the document may contain one answer to the question, more than one answer, or may have no answer). The dataset contains a source (reference) for each document with the year of publication.

### 2.2 Preprocessing the documents

This step is to prepare the data for later analysis. This includes several steps as follows:

- **Tokenization:** It is the process of dividing the text into separate words or tokens. This step is considered the cornerstone of NLP processes.
- **Remove stop words:** Stop words are words or letters that are repeated frequently in the language such as (في، و، على، من، إلى، هل، مع...). These words are usually removed in NLP processes because they do not have a useful meaning in the analysis.
- **Stemming:** It is the process of returning the word to its origin by removing any extra letters. The goal of this step is to simplify the word so that it can be easily compared later with similar words.
- **Limitation:** It is the process of returning the word to its root, i.e. its origin in the dictionary, to ensure that the word is available or correct.<sup>[3]</sup>

### 2.3 Create inverted index

Inverted index, also known as posting list, is a type of data structure used in information retrieval systems. It is an index that links unique (non-repeated) words (terms) in a set of documents to the document numbers in which this term appeared. Each term refers to a list of documents that contain this term. The figure below shows the structure of the inverted index.

## Inverted Index

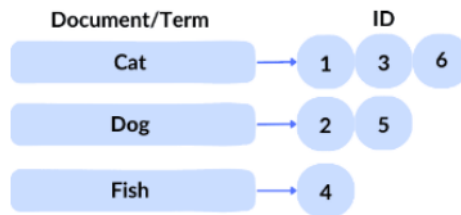


Figure 2: inverted index structure

The benefit of the inverted index is in retrieving documents that contain a specific term or a set of terms, which facilitates the process of retrieving relevant documents that are related to the query term. It also provides speed in searching, especially with a large number of documents.

### 2.4 Vectorize the documents with TF-IDF

Term frequency-inverse document frequency (TF-IDF) is a text vectorization technique that assigns a weight to each term. TF-IDF converts a vector of words into a vector of numbers for later use in compute similarity between a query and a set of documents. TF-IDF focuses on unique words that are very important in a document and gives a high value for it on the other hand it gives less attention to words that are frequently repeated in a set of documents. TF-IDF adjusts the weights of words based on their frequency in the document by taking into account the frequency of the word in the document, the total number of words in the document, the number of documents, and the number of documents that contain the unique word for which the weight is being calculated. The following equation represents how TF-IDF calculates the weight for a specific term in the document:

$$TF-IDF = W_{ij} = TF_{ij} \cdot IDF_i$$

$$= TF_{ij} \cdot \log_2\left(\frac{N}{df_i}\right)$$

Where:

$$TF_{ij} = \frac{\text{number of times term } i \text{ appears in document } j}{\text{total number of terms in document } j}$$

N: total number of documents

df<sub>i</sub>: number of documents where term *i* appears

### 2.5 Preprocessing the query

All preprocessing steps mentioned above (tokenization, removing stop words, stemming, lemmatization) are done to the query which is the question taken by the user. Then apply TF-IDF to the query vector to convert it to numerical vector.

### 2.6 Search for relevant documents

After finding the numerical vectors for each query and all documents using TF-IDF, we can find the relevant documents for the query by calculating the similarity between the query vector and each vector of the document vectors. In this project, we used cosine similarity to calculate the similarity between the query vector and the document vectors, which measures the cosine of the angle between the two vectors, the document with the highest cosine similarity value is considered the most relevant to the query. We take the top k relevant documents to extract the answer from them.

Cosine similarity (d<sub>j</sub>, q) =

$$\frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^l (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^l w_{ij}^2 \cdot \sum_{i=1}^l w_{iq}^2}}$$

## 2.7 Answer extraction

Extracting the answer depends mainly on the top ranked documents, in this step the answer to the question is extracted from the most relevant documents for the query (in our project we chose to extract the answer from the top one relevant document). In order to do this step a specific method must be adopted, we used NER (Named Entity Recognition) which is a method to identify the entities mentioned in the text such as names of people, locations and organizations, this helps in extracting the answer by finding common entities between the query and the sentences in the chosen document to extract the answer. We used the Pre-Trend BERT model for Arabic language based on the NER principle to understand the context and extract a more accurate answer.

## 3. EXPERIMENTAL RESULT

### 3.1 test cases

We test the system by giving it some questions and compare the results with what in the dataset, the first test is shown below:

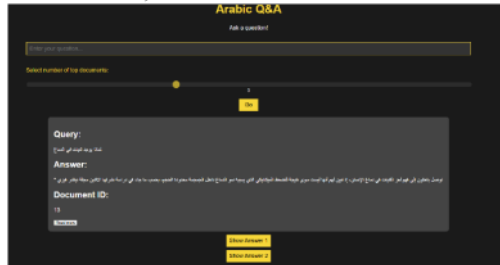


Figure 3: first test for our QA system

The question we ask is from the dataset and the system retrieve the top ranked document id "13" which is the document have the real answer in the dataset, also it extracted the answer correctly from the document 13.

Another test case, we try to ask the system question in our language (local language) and

see the results. The below figure shows this case:

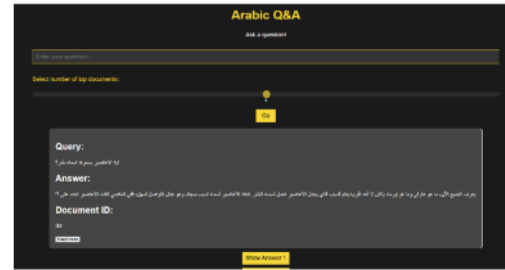


Figure 4: question with local language (paraphrased)

It retrieves the correct top one document and the correct answer.

The third test case we tried to ask another question with our local language (paraphrased) and the results is shown below:

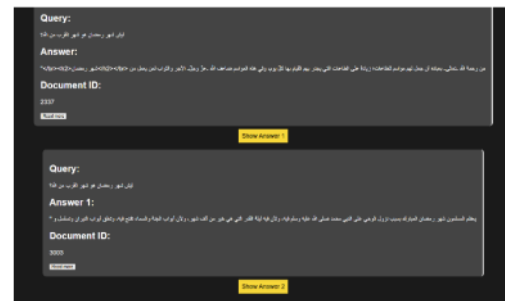


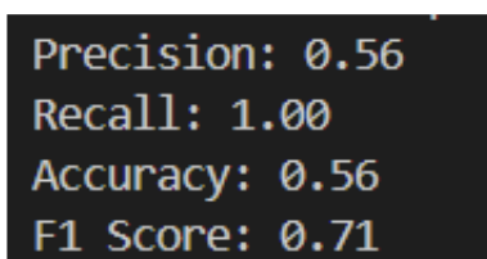
Figure 5: the third test case, the correct answer shown in the second rank

In this case it retrieves the top one document and it is relevant, but it has not the correct answer. however, it retrieves the second top relevant document and it has the correct answer.

### 3.2 Evaluation Metrics

- Precision (P): Of all the responses the system returned, 56% were accurate and relevant.
- Recall (R): No false negatives are left when the system obtains every relevant document that contain the answer, as indicated by a recall of 1.00.

- Accuracy: The overall accuracy of 0.56 most likely relates to the proportion of right answers to all the questions that were answered. Accuracy might be false due to working with imbalanced datasets which means there are few relevant documents.
- F1 Score: A score of 0.71 indicates that the system can retrieve relevant information and provide accurate responses with some degree of effectiveness.



Precision: 0.56  
Recall: 1.00  
Accuracy: 0.56  
F1 Score: 0.71

Figure 6: the evaluation metrics we got

#### 4. CONCLUSION

By effectively using and optimizing models such as BERT for Arabic, the Arabic Question Answering (QA) system has shown that it is possible to achieve precise, context aware question answering in the language. Addressing language subtleties, producing excellent QA datasets encompassing a range of dialects, and attaining good accuracy and relevance in responses are some of the major accomplishments. Notwithstanding obstacles like managing many dialects and enhancing contextual comprehension, the project establishes a strong basis for further investigation and advancement. The potential of natural language processing (NLP) technology to improve Arabic speakers' access to information is highlighted by this work.

#### 5. REFERENCES

- [1] <https://www.ontotext.com/knowledgehub/fundamentals/what-is-extractive-question-answering/#:~:text=Formally%2C%20Extractive%20QA%20is%20a,to%20answer%20a%20user's%20question>
- [2] [https://arbml.github.io/masader/search?name=D AWQAS&fbclid=IwZXh0bgNhZW0 CMTAAAR32JgppGD2LfgQULIcC\\_uuP7RjKfpJztBtEP0rkkYxy2y1cfcJPC4WTPY\\_aem\\_ZmFrZWR1bW15MTZieXRlcw](https://arbml.github.io/masader/search?name=D AWQAS&fbclid=IwZXh0bgNhZW0 CMTAAAR32JgppGD2LfgQULIcC_uuP7RjKfpJztBtEP0rkkYxy2y1cfcJPC4WTPY_aem_ZmFrZWR1bW15MTZieXRlcw)
- [3] <https://medium.com/@abhishekjainindore24/all-about-tokenization-stop-words-stemming-and-lemmatization-in-nlp-1620ffaf0f87>



# Arabic\_QA\_System\_Alaa\_Jamila\_Hala

---

## ORIGINALITY REPORT

---

10%

SIMILARITY INDEX

6%

INTERNET SOURCES

4%

PUBLICATIONS

5%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1

Submitted to City University of Hong Kong

Student Paper

1%

---

2

[www.mdpi.com](http://www.mdpi.com)

Internet Source

1%

---

3

[www.slideshare.net](http://www.slideshare.net)

Internet Source

1%

---

4

[www.ijcaonline.org](http://www.ijcaonline.org)

Internet Source

1%

---

5

Submitted to Universitas 17 Agustus 1945  
Surabaya

Student Paper

1%

---

6

[www.ncbi.nlm.nih.gov](http://www.ncbi.nlm.nih.gov)

Internet Source

1%

---

7

Submitted to University of Central Florida

Student Paper

1%

---

8

Submitted to Liverpool John Moores  
University

Student Paper

1%

---

9

[www.patentsencyclopedia.com](http://www.patentsencyclopedia.com)

10

S.M. Archana, Naima Vahab, Rekha Thankappan, C. Raseek. "A Rule Based Question Answering System in Malayalam Corpus Using Vibhakthi and POS Tag Analysis", *Procedia Technology*, 2016

Publication

1 %

11

[mafiadoc.com](http://mafiadoc.com)

Internet Source

1 %

12

[patents.justia.com](http://patents.justia.com)

Internet Source

&lt;1 %

13

Jiang, Shaohua, Haiyan Zhang, and Jian Zhang. "Research on BIM-based Construction Domain Text Information Management", *Journal of Networks*, 2013.

Publication

&lt;1 %

Exclude quotes

Off

Exclude matches

&lt; 6 words

Exclude bibliography

On