[DTI5125 [EG] Data Science Applications]

Group 4

[Final Project, Question Answering]

National Museum of Egyptian Civilization Guide Question Answering Search Engine (NMEC Guide)

[Abdelmageed Ahmed Abdelmageed Hassan]

[Ahmed Shehata Mahmoud Abomooustafa]

[Sarah Hossam Abdelhameed Elmowafy]

[Mohamed Sayed AbdelWahab Hussien]

1- Introduction

Question Answering System (QAS) is one of the most promising research areas in NLP (Natural Language Processing) which consisting of multiple processing steps such as Corpus Preparations, Information Retrieval (IR), Information Extraction (IE), Linguistic and Artificial Intelligence (AI).

The Question Answering system has a lot of applications such as extracting information from documents, Online examination system, document management, Language learning.

Question Answering (QA) systems have emerged as powerful platforms for automatically answering questions asked by humans in natural language using either a pre-structured database or a collection of natural language documents, QA systems make it possible asking questions and retrieve the answers using natural language queries and may be considered as an advanced form of Information Retrieval (IR), Question Answering systems in information retrieval are tasks that automatically answer the questions asked by humans in natural language using either a pre-structured database or a collection of natural language document

2 - Problem formulation

National Museum of Egyptian Civilization has open months ago, they announced that they need a system that helps any visitor to ask any question about Ancient Egypt, the visitor is interested in a concise, comprehensible, and correct answer, which may refer to a word, sentence, or a paragraph. Information based question answering systems (**NMEC**) is an appropriate solution to this case.

3 - Methodology

A lot of studies have been presented in this area. Most of them defined an architecture of Question Answering systems in three macro modules as shown in (Fig 1) [1], Question Processing, Document Processing and Answer Processing as showed in the next figure.

Besides the main architecture, QA systems can be defined by the paradigm each one implements [1]:

information Retrieval QA: Usage of search engines to retrieve answers and then apply filters and ranking on the recovered passage.

Natural Language Processing QA: Usage of linguistic intuitions and machine learning methods to extract answers from retrieved snippet.

Knowledge Base QA: Find answers from a structured data source (a knowledge base) instead of unstructured text. Standard database queries are used in replacement of word-based searches.

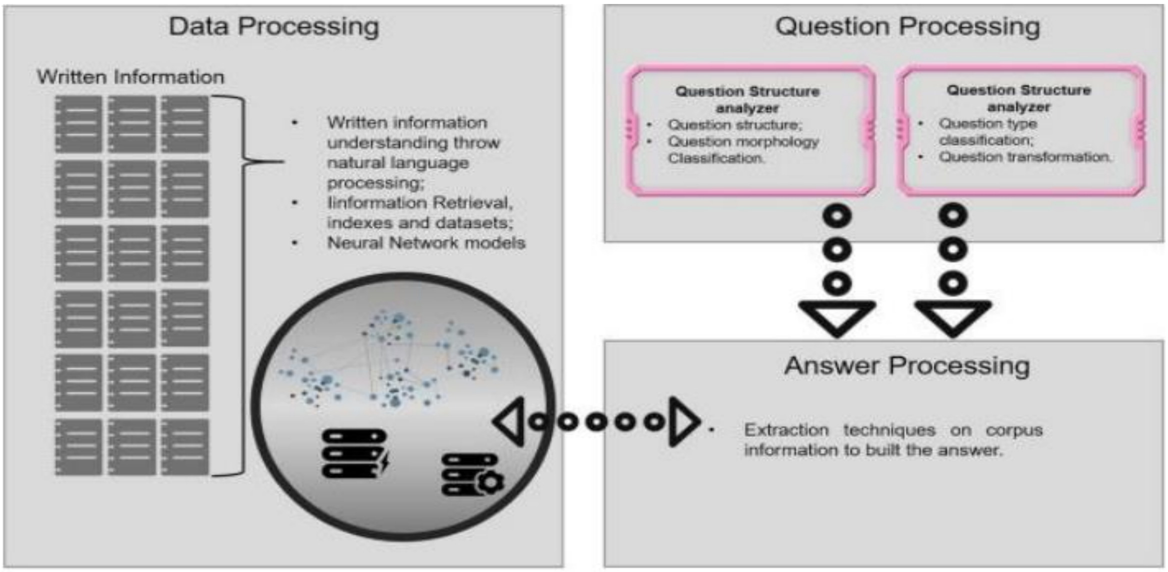

Data Processing
Written Information
Written information understanding throw natural language processing;
Iinformation Retrieval, indexes and datasets;
Neural Network models
Question Processing
Question Structure analyzer
Question structure;
Question morphology Classification.
Question Structure analyzer
Question type classification;
Question transformation.
Answer Processing
Extraction techniques on corpus information to built the answer.

Fig 1. QA architecture

On sending a question to the system as shown in figure. The NMEC system (Fig 2) will work to process both documents and questions to extract information and retrieve the answer through several steps as follows:

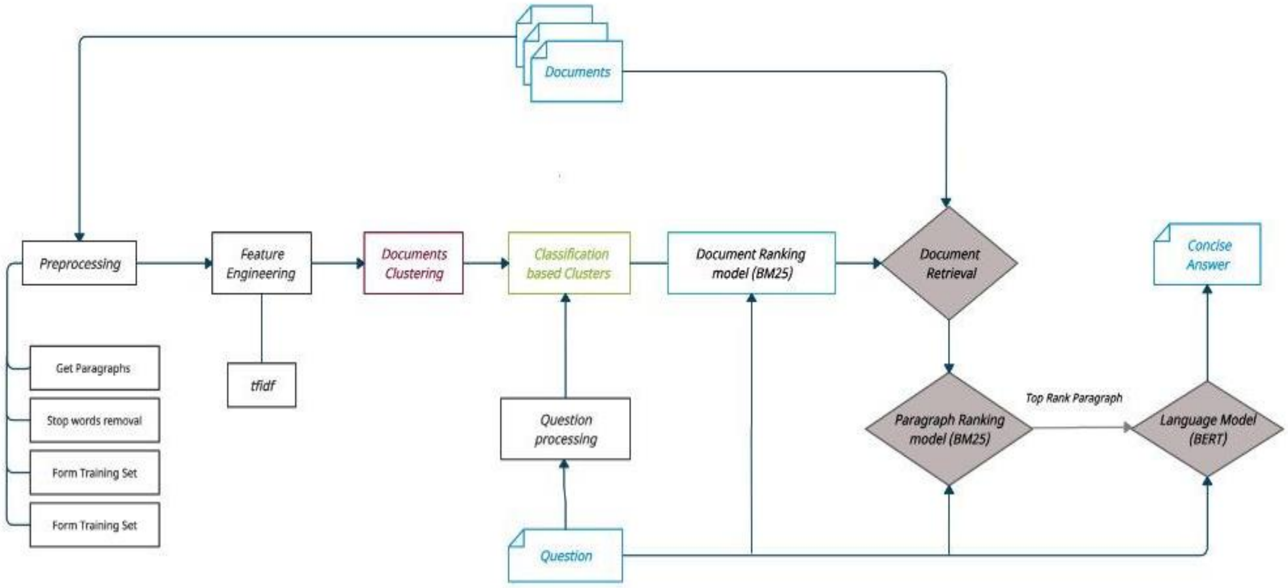


Fig 2. NMEC Guide Question Answering Workflow

The first step in any question answering system architecture is to understand the question of the user which is fed as input to the system for finding the answer. the main process in question and document processing is to clean the document as well as questions to reduce search space and for proper indexing.

The task of document retrieval **is** to find the document that contains the answer, proper indexing of documents is necessary for retrieving relevant documents for user questions. this can be done by using the NLP techniques such as tokenization, stop-word removal and lemmatization.

3.1 Feature Engineering

To make the system understand Natural Language, each text needs to be converted to vectors. tf-idf approach is used, which converts each document and question in such way that can be used by clustering and classification.

The tf-idf applied to the question of which entered to the first classification is different from the tf-idf applied to the question entered to the second classification, as the length of documents which the first classification model train on is different from the length of the documents that the second classification model trained on, so the tf-idf object should match the length of each model.

3.2 Clustering

After applying feature Engineering TF-IDF over data, we feed it to k-means clustering algorithms to produce different segments based on different historical eras, Clustering is used to reduce the search space for the required question, The model achieved Silhouette Score of K-means With TF IDF : **0.3522** Fig 3

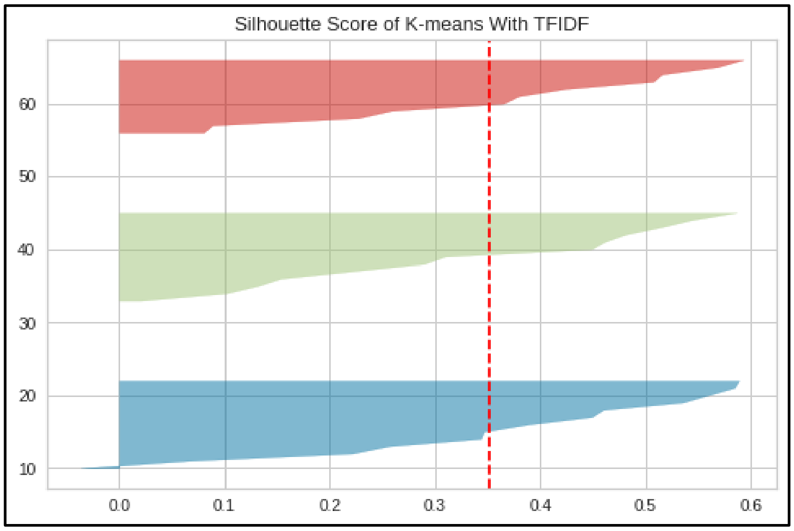


Fig 3.silhouette coefficient for the clusterd

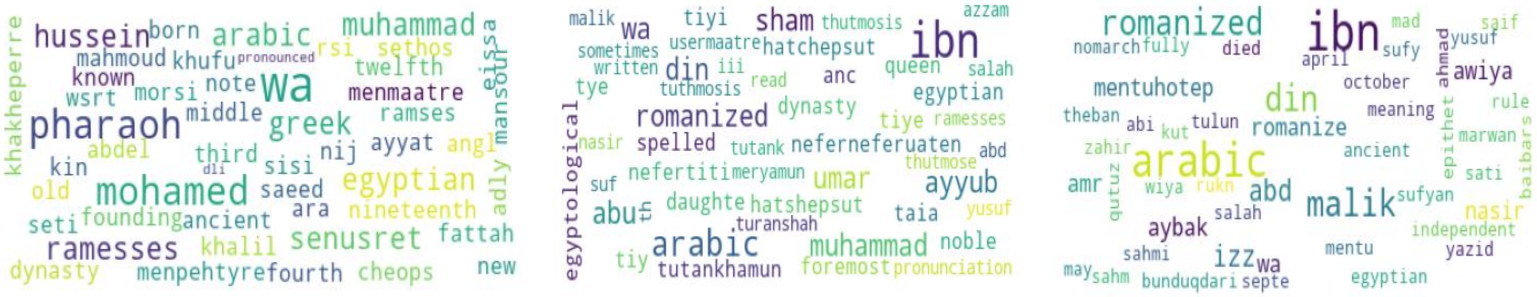
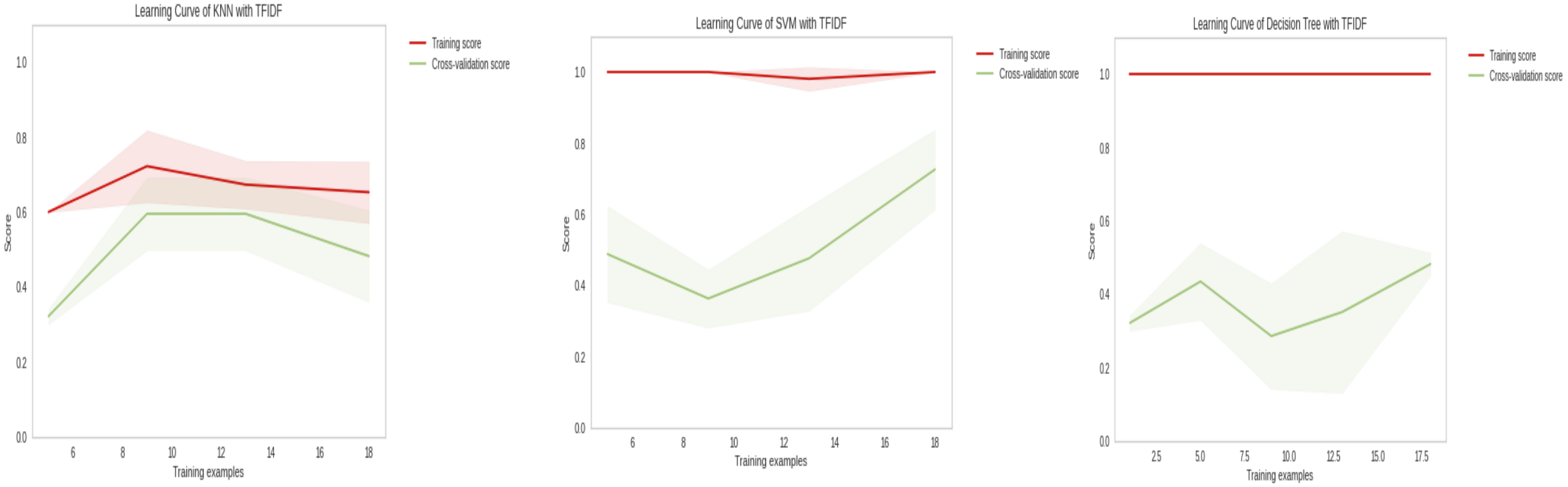


Fig 4. Word cloud for Clusters

3.3 Classification

By applying different machine learning algorithms as SVM, Decision Tree and KNN to find that the best model to classify question to return the cluster that contain the candidate documents , Decision Tree gives a **50 %** while KNN and SVM achieve the same accuracy of **75 %** , as SVM tends to overfit as the difference between training accuracy and validation accuracy is high and KNN gives the highest performance in classify documents to its cluster so we can decide which cluster the question belong we chose KNN as our champion model which its output will be considered in the model (first BM25 model) .

Training and Validation Loss (Fig 5)


KNN

SVM

Decision Tree

Fig 5. Training and Validation Loss Curves for The Models

3.4 First BM25 Model

BM25 (Best Match) is a ranking function that sorts a compartment of documents considering the number of query terms present in each document and ignoring the harmony between the question terms inside a document. It is used by many browsers to rank similar documents based on significance to the given search query [2].

After retrieval of the cluster contains the documents returned from classification that may contain the answer, the next step is to select a document from the selected cluster using the **BM25** model to return the high-ranking document for the answer.

3.5 Second BM25 Model

The second BM25 model take the selected document is divided into passages and each passage is considered as a document to extract the candidate passage for answering the question, then, feed the question and the selected document from the second BM25 to find the sentences from each paragraph that contain the answer for the user query, The Matched answer is extracted and outputs the one according to the scores and provides an exact answer to a user. The selected paragraphs and user queries are passed through the passage retrieval model.

3.5 Language model (BERT)

The last step in question answering task is how to extract the exact and concise answer for the selected passage using **BERT** (Bidirectional Encoder Representations from Transformers) For the Question Answering task, BERT takes the input question and passage as a single packed sequence. The input embeddings are the sum of the token embeddings and the segment embeddings. The input is processed in the following way before entering the model [3].:

1. **Token embeddings:** A [CLS] token is added to the input word tokens at the beginning of the question and a [SEP] token is inserted at the end of both the question and the paragraph (Fig 6).
2. **Segment embeddings:** A marker indicating Sentence A or Sentence B is added to each token. This allows the model to distinguish between sentences. For example, all tokens marked as A belong to the question, and those marked as B belong to the paragraph (Fig 7).

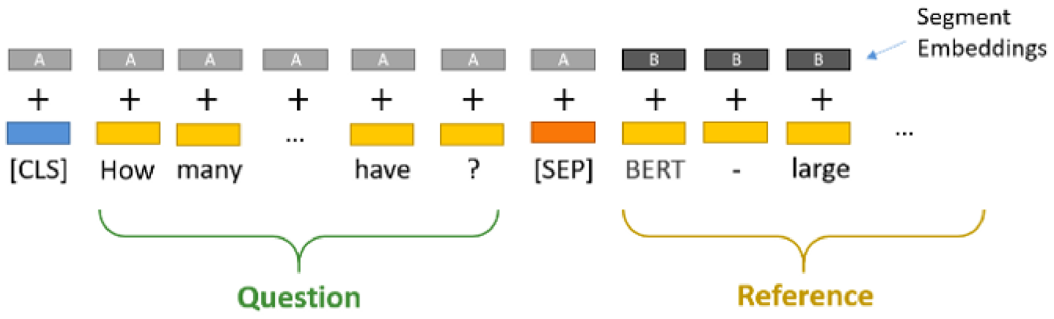


Fig 6. Token Embeddings on Bert Algorithm

To fine-tune BERT for a Question-Answering system, it introduces a start vector and an end vector. The probability of each word being the start-word is calculated by taking a dot product between the final embedding of the word and the start vector, followed by a SoftMax over all the words. The word with the highest probability value is considered. A similar process is followed to find the end-word [3].

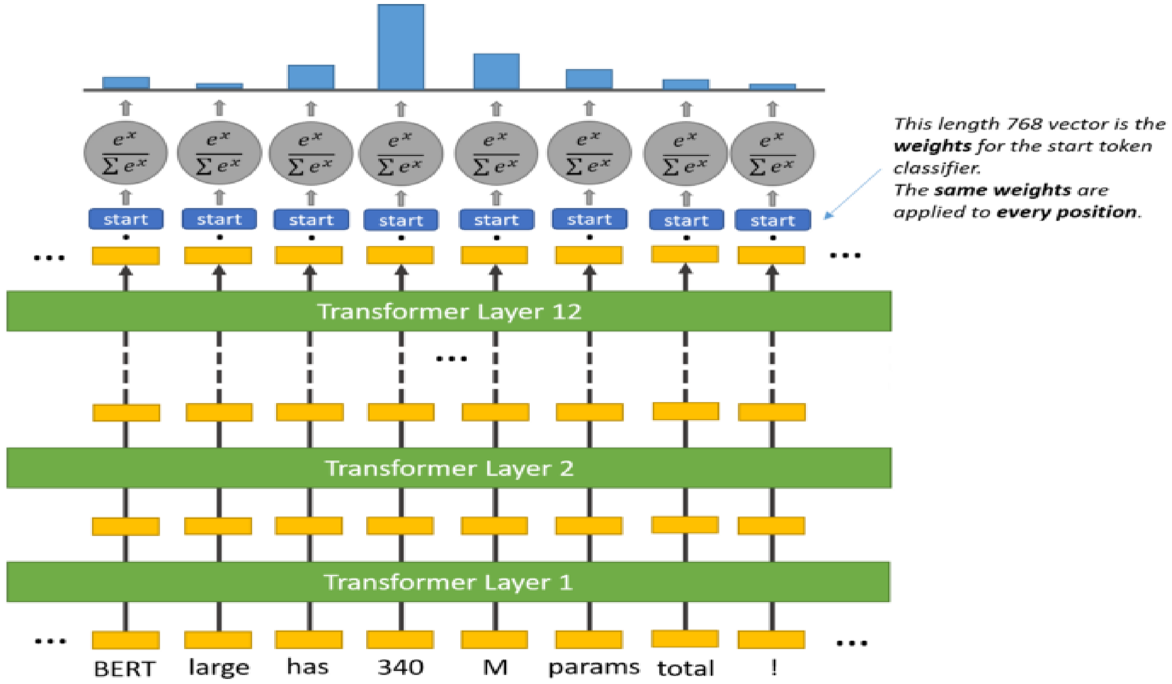

$\frac{e^x}{\sum e^x}$
start
Transformer Layer 12
Transformer Layer 2
Transformer Layer 1
BERT
large
has
340
M
params
total
!
This length 768 vector is the **weights** for the start token classifier.
The **same weights** are applied to **every position**.

Fig 7. Segment embeddings on Bert Algorithm

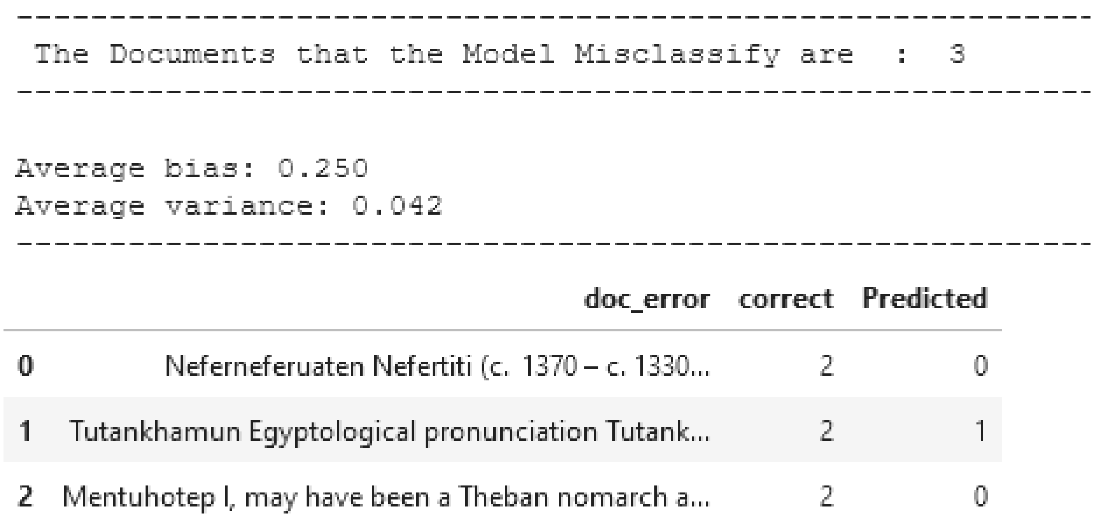
4. Error analysis

By following the previous workflow, we found that the system can being deceived in three different ways:

- First in classifying the question to return the clusters which the question belongs as this confusion may occur due to the wrong clustering of documents and this error in clustering due to the similarity of words that used to describe some king’s behaviors and lives as they rule the same country and their relationship with their people and building new cites as the very frequent words are found in both clustering, silhouette score of k-means with tfidf: 0.5042

- Second and third errors are in classification of question to retrieve the right cluster that contain the candidate documents and the retrieve the right document that contain the candidate paragraph , as classification model tries to predict the right cluster of the question to retrieve the candidate document that will be passed to the top ranking modules which will return the wrong paragraph if the question is misclassified either in classifier or in the first BM25 model this misclassify can be raised due to the incomplete, short and unbiased paragraphs which can’t indicate the right class of the document to retrieve the text from it .. for the KNN classification algorithm the classifier only misclassifies to paragraphs.

The next figure (Fig 8) shows the document that the model misclassifies and its variance and bias, here the model has high bias as more document need to be added (the trained data consists of 37 documents) to enhance the model performance.



```
----------------------------------------------------------
 The Documents that the Model Misclassify are  :  3
----------------------------------------------------------

Average bias: 0.250
Average variance: 0.042
----------------------------------------------------------
```

	doc_error	correct	Predicted
0	Neferneferuaten Nefertiti (c. 1370 – c. 1330...	2	0
1	Tutankhamun Egyptological pronunciation Tutank...	2	1
2	Mentuhotep I, may have been a Theban nomarch a...	2	0

5. Deployment

As for the deployment process , we Serialize our model parameters into PKL files, so in the production we only need to load the trained models, using Flask framework to create API to get the result of the models and Ngork to test the connection between Dailogflow and our APIs, but here we faced a problem of Timeout, after investigation we got that the response must occur within 10 seconds for Google Assistant application or 5 seconds for all other application, otherwise the request will timeout [4].

As our model exceeds the time limit because we have a sequence of pipelines that must be executed to get the final result. So, we make an alternative solution by building a simple webpage which takes the question from the user and triggers our pipelines to get the answer back to the user

6. Future Work

As next step in our NMEC Guide question answering system we need to optimize our models to get response in less than 5 seconds and connect it with the Dialogflow, hence we can launch our chatbot which can be available for the users that tends to visit the museum in the future, and adding more documents to enhance our model performance.

7- References

[1] Malik et al., 2013, Bhoir and Potey, 2014, Neves and Leser, 2015

[2] Z. Zheng, A. Arbor, Answer Bus Question Answering System, Proc. Second Int.Conf. Hum. Lang. Technol. Res. (2002) 399–404.

[3] https://medium.com/saarthi-ai/build-a-smart-question-answering-system-with-fine-tuned-ber b586e4cfa5f5

[4] https://cloud.google.com/dialogflow/es/docs/fulfillment-webhook