A Semantic Search and Recitation IOS Application for Holy Quran

Ahmed Khattab[1], Ahmed Sayed[1], Esraa Mohee El-Deen[1], Eyad Shokry[1], Hoda Ali[1], Youssef Hanafy[1], and Ensaf H. Mohamed[1,*]
[1]Faculty of Computers & Artificial Intelligence, Helwan University
Email: (Ensaf_hussein@fci.helwan.edu.eg)

ABSTRACT

The Holy Quran is undoubtedly one of the most important books for all the Arabic/Islamic People, covering too many concepts and topics which guide all the Muslims in their life, behaviors, and acts of devotion.
This Project is developed to help all Muslims to deal with the Holy Quran easier and faster. as this Project allow them to search the Quran for specific Keyword or Verse, and for a Concrete Topic or Conceptual Topic which is a challenging task. It also helps them in its Memorization and Recitation.
This project consists of two parts; the first is a concept-based Search Engine for Holy Quran based on a Deep Learning Model called 'word2vec' used to search in the Quran using topic or concept with accuracy about 70%.
This search engine is built through four phases. First one is to build a new Quran dataset in which we annotated each verse with it's related topic using Mushaf Al-Takweed, Second, we collect large classic Arabic corpus which is about 40 million words, then we trained our word2vec model using this corpus, third, we get the vector of each topic in our dataset and query's vector using our word2vec model. Finally, we get the most relevant topic by calculating the cosine similarity between query's vector and topics' vectors and retrieve its verses from our dataset. We calculated our system's accuracy by collecting about 300 queries using google form and tried them on our system which could satisfied about 70% of these queries. The second part is an iOS Mobile Application which used to introduce the search tool and, also, to help users memorize and recite the Quran using voice and help them know their mistakes. This system has high accuracy in evaluating users' sayings in comparison of other applications.

1. INTRODUCTION

This Project is developed to serve all Muslims everywhere with new features that help them discover and know more about their Religion through the Holy Quran Book which considered the first and most important Book which answers all our questions about Legislations, correct Behaviors, what we should do and what we shouldn't, and all our dealings in the Society.
It is important to all of us to know as much as we can about this Book. And to commit to memory a lot of its verses. and our Project is basically developed to make these points easier, faster, more interesting and interactive.

2. RELATED WORK

Ahmad T. Al-Taani et. Al.[1] proposed algorithm that is used for searching the Quran about keywords and concepts, In this Research they handled words not sentences. but not a Query which consists of more than one word, Their Semantic Searching is based on Microsoft Word Thesaurus tool. they toke each query's synonyms from it and run their algorithm. So if there are words not in this thesaurus it will never able to get it's synonym.

Hikmat Ullah Khan et. Al. [2] used SPARQL query language to develop a QA System which focused on the domain of Animals and Birds in Quran. Their work focused only on the domain of animals and birds, but extensive research is required to create domain ontologies for all main domains mentioned in Holy Quran.

Mohammad Alhawarat [3] build up the first stage in a framework that will allow possible semantic search in the holy Quran. This is done by applying LDA topic modeling and TF-IDF Techniques to chapter Joseph of the holy Quran as a case study. This chapter has been chosen because it includes relative topics regarding story of the prophet Joseph (PBUH). The LDA topic modeling has been applied to words, roots and stems of that chapter. This paper is working on only one chapter (Joseph -) not the whole Quran.They are dealing only with one-word query. There is no consideration to Synonyms of words. no consideration to Semantic or meanings.

Hammad Afzal et. al. [4] proposed a new Arabic question answering system in the domain of Al Quran. The system prompts users to enter an Arabic question about Al Quran. Then, this system retrieves relevant Quranic verses with their Arabic descriptions from Ibn Kathir's book. This system uses 1,217 Quranic concepts integrated from the Quranic Arabic Corpus Ontology and Quranic Topic Ontology. It is claimed that retrieved results' accuracy can reach 65% using the top result. This system has three phases for answering a question: question analysis using the 'Morphological Analysis and Disambiguation of Arabic' tool (MADA), IR using 'explicit analysis approach' and answer extraction. The proposed system does not recommend a solution for if the question terms do not match any concepts from the Quranic Ontology.

3. METHODOLOGY

In this project we proposed a system that searches on Quran verses by concept and allows users to memorize and recite Quran verses by sound. As shown in the Block Diagram, our Application has four main Components:

- **Quran Search Engine (Keywords & Topics) Block:** The System interact with this component by sending keywords or topics to get all the verses that contain these keywords or related to the desired topic
- **Quran Reading Block:** It takes the desired Sura and verses range from the main system and returns with the verses themselves for memorizing/revising before recitation.
- **Quran Recitation Block:** it takes Sura name, range of verses and user's recitation. Finally, it sends the evaluation and highlights for user's recitation mistakes.

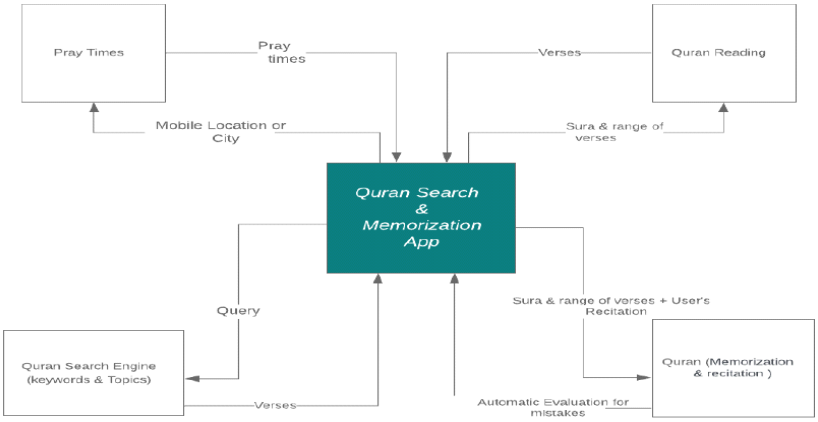

Figure 1: Quran search & memorization app block diagram

• **Pray Times Block:** When user wants to know pray times, he enters his city and if user enter invalid city then the system gets user's phone location, then retrieve pray times according to it.

3.1 Quran Search Engine

As shown in fig. 2, our Application has four main Components:

• **Search Quran by Keyword Block:** The System interact with this component by sending keyword or a series of keywords to get all the verses that contain these keywords from the dataset.

• **Search Quran by Topic:** The System interact with this component by sending a topic to it then this component interacts with its sub-components as following:

- The topic is sent to **the word2vec model** to get a vector for each word in the topic, then these vectors are sent to the **compute average vector** sub-component to get only one vector for the topic which consists of more than one word.

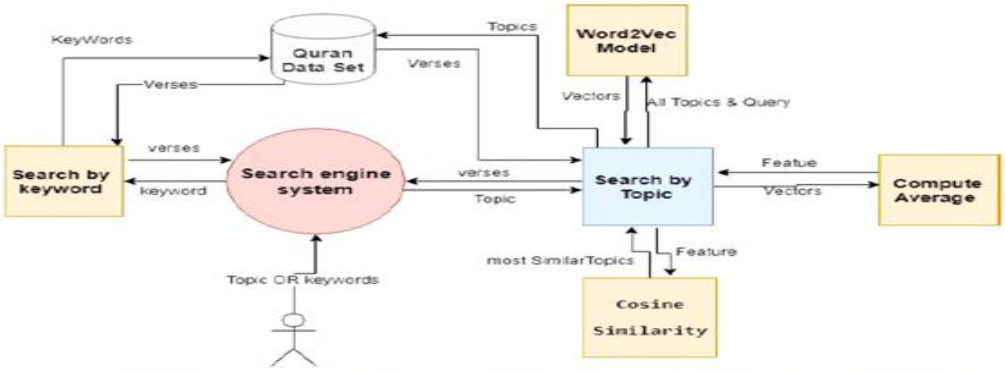

Figure 2: Search Engine block diagram

- The same process is done for all topics we have in the dataset
- All these vectors go to the last sub-component which is **calculate cosine similarity** between user's topic vector and all topics vectors we have and return to the system with the most related topic and its verses.

3.2 Building Qur'an's Dataset

we needed to get a documented and trusted representation of verses of the whole Quran and their according topics, because this is something religious which we cannot make it ourselves to be trusted for the users of the application. So, we searched till we get a book which contains exactly what we need.

- **Book's Name:** Mushaf Al Tajweed Quran book
- **Book's Compiled by** Dr. Mohammed Fayez Kamel. Under Supervision of Dr. Ali Abu Al-Kheir
- **Book's Publisher:** published by Dar Al-Maarifa in Syria and **authenticated by Al Azhar Islamic Research Academy in Egypt.**

3.2.1 Dataset Columns

The basic dataset is downloaded from qurandatabase.org. Its Columns are:

- **SuraID:** id for each verse's Sura. It's values from 1 to 114
- **VerseID:** id for each verse's number.
- **Ayah:** the text of each verse with its Tajweed and word signs

We added the following Columns:

- **ChapterID:** id for each verse's Chapter. It's from 1 to 30.
- **AyahText:** the text of each verse but without word's formation to be able to process the word.
- **SuraName:** each sura's name to be aware with the whole information of the verse.
- **ManualKeyword:** the topics that verse represent with its hierarchical sub and main topics separated by dash "-".

To represent the tree structure of the topics we made a separated file consists of two columns:

- **Topic:** each individual topic is represented in this column
- **code:** a string represents it to let us know if it's a main topic or sub-topic. It's idea like the content page which is consists of sections and sub-sections. For more Illustration, this is a screen shot from the file:

So now we have a dataset contains more than 800 topics. each of them is assigned to specific verses. The number of verses which have at least one topic is about 5100 verses from 6263 (without Basmala). the remaining verses are not covered by the Islamic Expert who wrote the mentioned Book.

topic	code
اركان الاسلام	1
التوحيد	1.1
توحيد الله تعالى	1.1.1
وجدانية الله تعالى	1.1.1.1
وحدانية الله تعالى	1.1.1.2
الالوهية	1.1.1.3
لا شريك لله تعالى	1.1.1.3.1
لله الاسماء الحسنى	1.1.1.3.2

Figure 3: tree file

3.3 Quran Memorization/Recitation

As shown in the fig. 3, we have three main components:

- **Quran Reading:** user uses his voice to record an audio for his recitation and the record is immediately passed to the speech recognition API by Apple's Speech framework [25] while it is still recording, which allow users to see their converted speech-to-text while reciting.

- **Recitation Editing by Typing:** to give users the best experience we allow them to edit the converted text by typing before evaluating it. As we know, some errors may happen while converting speech to text because of noise or any other reason.

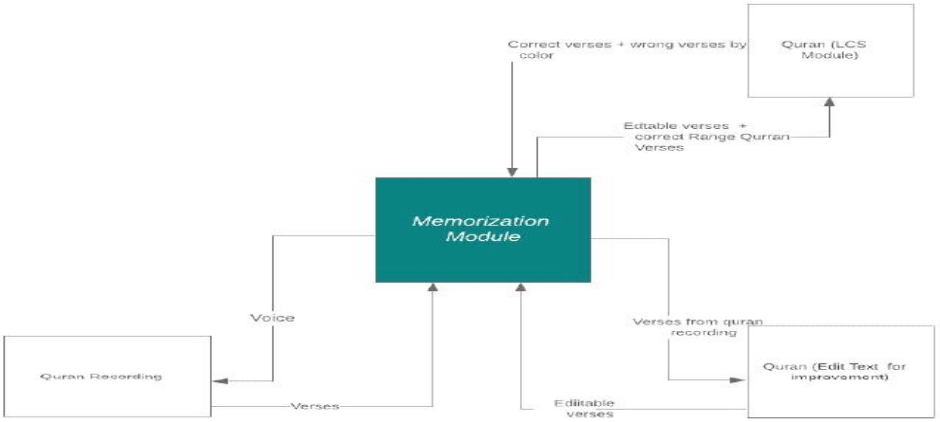

Figure 4: proposed model for Quran Recitation module (Component Level Diagram)

• **Evaluation (Longest Common Subsequence Algorithm):**

The Longest Common Subsequence (LCS) of two strings is the longest sequence of Words/characters that appear in the same order in both strings.

For Example: The Longest Common Subsequence of "ولم يلد لم أحد لله هو قل ولم يولد ولم يلد لم الصمد الله أحد الله هو قل" and "أحد كفوا يكن ولم يولد أحد كفوا له يكن ولم يولد ولم يلد لم أحد لله هو قل" should be: "أحد"

So, we used this algorithm to compare user's recitation with the correct verses of the Quran. When we pass both to the algorithm it will return only the correct statements of user's recitation.

After finishing text editing by user and submitting his recitation. This recitation and the
correct range of verses from Quran which the user selected to recite are sent to the evaluation module, which runs the longest common subsequence algorithm using both verses (user verses and the correct verses) to give the user an automatic feedback and evaluation about his mistakes through our Application User Interface.

4. TEST AND RESULTS

4.1 Evaluation Criteria

In this section, we will discuss how we evaluated our Word2Vec Model. After searching in how to evaluate a word2vec model created to make a search engine we found that there isn't a computerized method or evaluation function can do this task and it may be done manually by trying several queries
and evaluate the outputs by eye.
This is one of the field experts' opinion about evaluating word2vec models on stackoverflow.com [5]

So, we decided to make a quick Google Form to collect some queries to use them in the evaluation process to make it more fairly than if we tried our queries which may be biased for our model and we got 104 responses contains about 500 queries.

We needed to filter these responses because some of them was irrelevant and others contain typos. After filtering them the remaining queries number was about 300. We run our model on them to evaluate it as following:

We got the most relevant topics for each query and check them by eye. If there is at least only one topic relevant to the query, then we considered query's owner is satisfied.

After calculating the percentage of the satisfied queries, we got an accuracy about 70% which considered not bad for a problem like this

5. CONCLUSION & FUTURE WORK

In this document we proposed a Quran Search Engine by keywords and topics, we manually annotated Quran verses with concepts using Mushaf Al-Tajweed, then we trained our word2vec model on 40 million classic and MSA words, which is used later to generate vectors for both user query and all the collected topics, finally cosine similarity is computed between both vectors and the most relevant verses are retrieved. We collected about 300 queries and when we tried our system on them, it could satisfy about 70% of these queries with related verses which is considered very good results for a difficult problem like this.
We also developed the iOS Version of our application which includes the search engine and a recitation feature which allow users to check their memorization for Quran verses with a smooth and flexible user experience which solved all problems we recognized in the related and similar applications.

As a future work, we plan to do the following:
1- Improve Quran Dataset by reviewing the manual annotation by Islamic expert.
2- Improve word embedding by adding more classic Arabic, and colloquial Arabic corpus to handle users' quires that write in different dialects.
3- Find a standard metrics to evaluate Quranic searching tools.
4- Develop the Android version of the mobile application.
5- Allow searching Quran by English language.

REFRENCES

[1] A. T. Al-taani and A. M. Al-gharaibeh, "Searching Concepts and Keywords in the Holy Quran," *System*, no. November, pp. 1–3, 2010.

[2] H. Ullah Khan, S. Muhammad Saqlain, M. Shoaib, and M. Sher, "Ontology Based Semantic Search in Holy Quran," *Int. J. Futur. Comput. Commun.*, vol. 2, no. 6, pp. 570–575, 2013.

[3] M. Alhawarat, "Extracting Topics from the Holy Quran Using Generative Models," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 12, 2016.

[4] H. Afzal and T. Mukhtar, "Semantically Enhanced Concept Search of the Holy Quran : Qur ' anic English WordNet," *Arab. J. Sci. Eng.*, 2019.

[5] https://stackoverflow.com/questions/52645459/how-to-evaluate-word2vec-model