Life Science Journal

Websites:
http://www.lifesciencesite.com
http://www.sciencepub.net

Emails:
editor@sciencepub.net
sciencepub@gmail.com

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

Enhancing Arabic Reading Proficiency through Artificial Intelligent Application

Ashwag Maghraby
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
aomaghragy@uqu.edu.sa

Azzah Mahjoob
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
s438006566@st.uqu.edu.sa

Waad Alsaedi
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
s438001556@st.uqu.edu.sa

Atheer Alsaedi
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
s438005893@st.uqu.edu.sa

Manar Alsaedi
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
s438000262@st.uqu.edu.sa

Amirah Alotaibi
dept. Computers, Umm Al-Qura University
Mecca,24382 , Saudi Arabia.
s438000578@st.uqu.edu.sa

Abstract: *According to research, learning to read in classical Arabic is more difficult and time-consuming than learning to read in other languages. In addition to the complexity of the language's grammar and syntax, the difficulty of reading Arabic is further compounded by the fact that it is a non-Latin alphabet. This means that the language is not based on the same phonetic principles as English or other European languages. In fact, many people fall into mistakes when reading classical Arabic, such as misreading Harkat. This research proposes a solution to correct reading mistakes in classical Arabic by using a machine learning and natural language processing approach. In this research, the sound of the Arabic reader was recorded and converted into Arabic text to recognize the characters of the text using the Google API. The dataset was collected utilizing Google Forms and Arabic audio websites and was then labeled manually. The natural language processing approach was employed to detect prevalent patterns in misreadings of classical Arabic texts. Subsequently, a collection of machine learning algorithms was trained to recognize and correct the errors committed by readers. The effectiveness of these algorithms was then evaluated through severe testing, revealing that the random forest algorithm attained the highest level of accuracy, with an impressive accuracy rate of 81%. The outcomes of this investigation provide compelling evidence that the proposed machine learning approach holds immense potential for significantly enhancing the accuracy of reading classical Arabic texts.*

[Ashwag Maghraby, Azzah Mahjoob, Waad Alsaedi,Atheer Alsaedi, Manar Alsaedi, Amirah Alotaibi. **Enhancing Arabic Reading Proficiency through Artificial Intelligent Application**. *Life Sci J* 2024;21(5):39-49]. ISSN 1097-8135 (print); ISSN 2372-613X (online). http://www.lifesciencesite.com. 05. doi:10.7537/marslsj210524.05.

Keywords: *Machine learning (ML), Natural language processing (NLP), Classical Arabic error detection, Self-learning app*

1. Introduction

One of the primary difficulties in reading classical Arabic is the script itself. Arabic is written from right to left, which can be disorienting for those used to reading from left to right. Additionally, Arabic script contains many letters that are similar in appearance, making it easy to confuse them when reading quickly or without proper training.

Another challenge in reading Classical Arabic is the extensive vocabulary used in texts. Many words have multiple meanings and connotations, and understanding their intended meaning requires deep knowledge of the language. In addition, the grammar of Classical Arabic is also challenging for non-native speakers. Language has a complex system of verb conjugation and noun declension, and the rules for sentence structures can be difficult to master.

Furthermore, Classical Arabic includes many archaic and poetic terms that are not commonly used in modern Arabic, thus adding another layer of complexity to the text. In fact, the written form of Arabic is very different from the spoken form, making it a challenge to learn.

Despite the difficulty of learning Arabic, there are many benefits to doing so. Not only is it a beautiful language, but it is also a gateway to understanding Middle Eastern culture and history. Knowing the language can open up a world of opportunities for travel, business, and education.

Indeed, the occurrence of mistakes during the reading of Classical Arabic is not limited to non-Arabic speakers alone. Even some native Arabic speakers may encounter challenges due to the influence of their regional dialectal Arabic. Consequently, this can result in mispronunciations and misunderstandings, particularly when engaging with formal or religious texts. It is crucial for all Arabic speakers to actively pursue a comprehensive comprehension and accurate pronunciation of classical Arabic.

Three common types of mistakes in reading Classical Arabic are Confusing Similar Letter Sounds (Replacement); pronunciation and phonetics mistakes; and harakat mistakes. Replacement mistakes occurs when learners mix up or replace similar letter sounds. Pronunciation mistakes occur when sounds are mispronounced, while harakat mistakes involve the incorrect placement or omission of diacritical marks(such as fatha, kasra, and damma) that indicate short vowels. All types of mistakes have an impact on the accuracy and clarity of Arabic pronunciation. An examples of these mistakes that can occur in Arabic[3] are:

1- Confusing Similar Letter Sounds (Replacement): Mixing up Arabic letters that have similar sounds but differ in their specific points of articulation. As an illustration:

- Incorrect: سمك (samak) pronounced as "thamak" - meaning "fish"
  Correct: سمك (samak) - meaning "fish"
- Incorrect: "قُلْ هُوَ اللَّهُ أَخَدٌ". where the letter "ح" was replaced with "خ" which is incorrect, and the correct form is " قُلْ هُوَ اللَّهُ أَحَدٌ".

2- Harkat: the replacement of Damma "ُ " with Sukoon "ْ " in the word "كُفُوًا".

3- Pronunciation and phonetics of Classical Arabic:

- Error in pronouncing of Emphatic Letters: Incorrectly articulating the emphatic consonants (ق ,ض ,ص ,ظ ,ط, and ع). These letters necessitate a distinct constriction in the throat during pronunciation. As an illustration:
  - Incorrect: قلب (qalb) pronounced as "kalb" - meaning "heart"
    Correct: قلب (qalb) - meaning "heart"
- Misjudging Vowel Length: Making errors in determining the duration or length of short vowels (َ,ِ,ُ) and long vowels (ا, و ,ي). This can affect word meaning and the overall flow of a sentence. As an illustration:
  - Incorrect: بيت (beit) pronounced as "bayt" - meaning "house"
  - Correct: بيت (beit) - meaning "house"
  - Incorrect: الصَّمَدُ (Al samad) pronounced as "Al sammad" where the letter "م" pronounced as a heavy letter
    Correct: الصَّمَدُ (Al samad) pronounced the letter "م" as a light letter.
- Inconsistent Hamza Pronunciation: Having inconsistencies in pronouncing the Arabic letter hamza (ء) accurately, which can occur at the beginning, middle, or end of words. It can be pronounced as a glottal stop or as a vowel sound. As an illustration:
  - Incorrect: أكل (akal) pronounced as "akel" - meaning "ate"
    Correct: أكل (akal) - meaning "ate"
- Incorrect Articulation Points: Failing to produce sounds from the appropriate points of articulation (makhraj) in the mouth or throat, leading to pronunciation distortions. As an illustration:
  - Incorrect: خارج (kharaj) pronounced as "sharaj" - meaning "outside"
    Correct: خارج (kharaj) - meaning "outside"

Classical Arabic reading is typically learned through reading sessions (face-to-face or online sessions through Microsoft Teams, Zoom, and Webex) with a teacher who listens to the student's reading, identifies reading errors, and provides appropriate corrections to the student. This process is repeated until the student can read the Classical Arabic in its all-proper rules. This is an effective and good learning method, but it requires a dedicated teacher for each learner or group of learners. It also requires extensive training time so that learners can correctly read Classical Arabic. The availability of Classical/Quranic Arabic sessions in countries where Islam is not the dominant religion can be quite scarce.

Recently, advanced technology has assisted in the development of applications for beginners by listening to the reading of the Classical Arabic while following the corresponding text on screen. This has made learning the Classical Arabic and listening to real Arabic speakers much easier. Some applications provide the Arabic speakers reading for specific text, and then allow the learner to repeat it. However, this method is not effective for beginners to learn the Classical Arabic without a tutor because it does not show whether the learner's reading is correct or incorrect.

Thus, some researchers have tried to develop applications to detect Classical Arabic reading errors using both artificial intelligence (AI) and natural language processing (NLP) approaches. A researcher in "Al-Moallim" [4] used both an automatic speech recognition (ASR) technique for recognizing the reader voice, and a deep learning approach to detect reading errors. The application is based on the open-source CMU Sphinx-4, which is a Hidden Markov Model (HMM) based and was trained using Arabic characters.

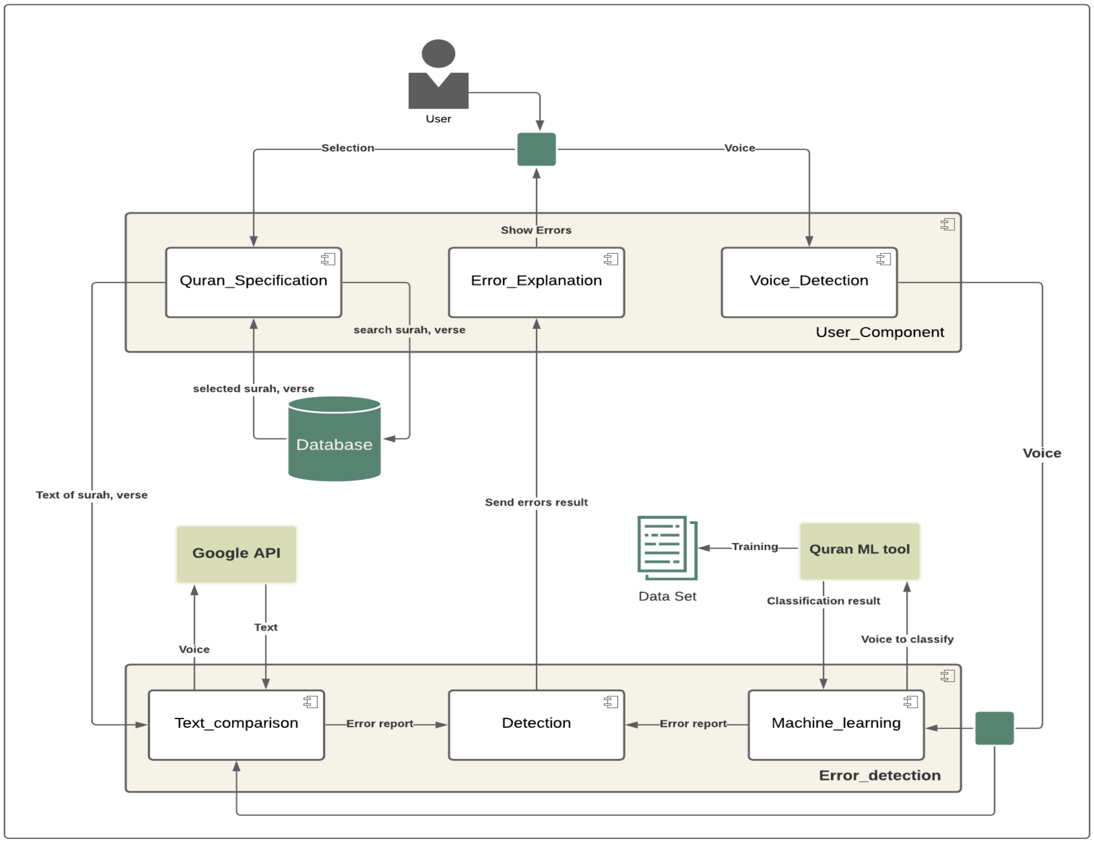


Figure 1.

However, there are two main issues in this application: (1) the correction is determined only for Harkat mistakes, and (2) it only shows an alert without determining the type of mistake. Therefore, users should have good knowledge of Arabic rules of pronunciation and phonetics to be able to correct their own mistakes.

Alobaylani, Parvez, and Alsuhibany [5] designed, implemented, and tested a system called “E-Hafiz,” with the goals of facilitating the learning of the Classical Arabic by minimizing errors or mistakes of all kinds and systematizing the reading process. The system has been trained in several reading, including women, men, and children. The system was based on the Mel-Frequency Cepstral Coefficient (MFCC) approach for feature extraction and the HMM approach for training and testing the model. The E-Hafiz system only detects pronunciation, phonetics and missing word errors. It achieves recognition rates of 92%, 90%, and 89%. However, the system has some disadvantages: (1) it contains a small dataset of experts’ voices; (2) the system does not work in real time and, therefore, it cannot point out the user’s mistakes during reading; (3) the system serves only those who have a good knowledge of pronunciation and phonetics rules; and (4) the system works on the word level, which may make the system unable to recognize mistakes on the character level.

In “Tasmee” research [6], developers developed a real-time application called Tasmee that uses speech recognition technology to detecting missing word errors. Tasmee faces the problem of slowing in response, only detecting missing word errors, and prolonging silences. If a mistake is detected, the mistake’s location will be marked with red without any explanation for the type of mistake.

Many efforts have been made in previous applications, but it is clear that there are some problems with these applications, such as slow response, not detecting the common mistakes of Classical Arabic reading, and failure to clarify the type of mistakes.

The primary goal of this research was to address the aforementioned challenges and assist individuals, both native and non-native Arabic speakers, in enhancing their proficiency in reading Classical Arabic. By identifying common mistakes that occur during Classical Arabic reading and providing real-time corrections, this research aims to reduce reliance on experts and facilitate independent learning. Ultimately, the objective is to empower individuals to improve and correct their reading skills in Classical Arabic.

The rest of the paper is organized as follows. Section 2 describes the methodology of this work. In Section 3, the results are discussed. Finally, this paper concludes with a summary and future work in Section 4.

2. Methods

In this project, the Quran played a central role as the primary reference for the classical Arabic book. Because of the Quran's remarkable linguistic and literary attributes, it is a priceless tool for identifying and correcting reader errors. By carefully studying the Quranic text, this project aimed to identify common mistakes occur during when reading Classical Arabic and offer suggestions for enhancing reading proficiency. The Quran's significance as a linguistic and literary masterpiece laid a solid groundwork for this research, allowing for a thorough examination of reader mistakes and the formulation of effective strategies for correction and improvement.

This project used two different types of error detection: (1) detection by machine learning (ML) to detect pronunciation, phonetics, and Harakat mistakes, and (2) detection by text comparison, which is dependent on natural language processing (NLP) to detect missing, replacement, addition, and repetition mistakes.

Figure 1 illustrates the architectural components of the system, comprising two main elements: (1) the user component and (2) the error detection component. The user component consists of three sub-processes. Firstly, the Quran specification component enables users to select the chapter (sura) they wish to read, with Chapter Al-Ikhlas currently available, and additional chapters planned for future implementation. Secondly, the voice detection component captures the user's voice input. Lastly, the error explanation component provides users with feedback on their reading errors.

The error detection component encompasses three sub-processes. Firstly, the text comparison component performs a comprehensive analysis by comparing the user's reading of the chapter text with the corresponding text in the database. Its objective is to detect any occurrences of replacement or missing words within the user's reading. This comparison helps detect discrepancies in the user's reading. Secondly, the machine learning component operates in real-time to classify the user's reading accuracy based on Harakat (vowel marks) and pronunciation and phonetics rules. It determines whether the reading adheres to the correct pronunciation and rules. Lastly, the detection component forwards the identified user mistakes to the error explanation component for further analysis and presentation to the user.

To detect mistakes, the system uses both the user component and error detection component, and the user's reading goes through four stages:

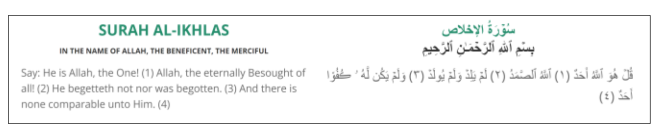
1-The user component extracts the features of the user's voice and transmits them to both the machine learning (ML) component and the text comparison component. This transmission process is facilitated by the user component.
2- The ML component receives the extracted voice features and employs machine learning models to classify the user's reading accuracy.
3- Simultaneously, the text comparison component receives the user's voice and initiates a comparison between the user's reading and the correct chapter stored within the application. This comparison aims to identify any instances of replacement or missing words.
4- Subsequently, the detection component transmits the identified user mistakes to the error explanation component. The error explanation component presents the user with a comprehensive overview of their errors, highlighting them in red for Harakat (vowel marks), replacement, or missing words, and in yellow for pronunciation and phonetics related errors.

For a more comprehensive understanding, the following subsections provide detailed explanations of the specific types of error detection employed in the system.

2.1. Error Detection by Machine Learning

ML models were utilized to detect pronunciation, phonetics, and Harakat mistakes. A dataset was meticulously curated, comprising recordings of people's readings. Each reading was manually evaluated to determine its correctness. The dataset was subsequently utilized to train the ML model, enabling it to learn and classify various types of mistakes accurately.

Throughout this project, six ML algorithms (Decision tree classifier, Random forest classifier , Support vector machine, Logistic regression , K-neighbors classifier, and Gradient boosting classifier) were rigorously tested and evaluated. The accuracy of each algorithm was meticulously recorded. After thorough analysis, it was determined that the random forest algorithm exhibited the highest level of accuracy among all the tested algorithms. Its exceptional performance and accuracy made it the preferred choice for the project's objectives.


SURAH AL-IKHLAS
IN THE NAME OF ALLAH, THE BENEFICENT, THE MERCIFUL
Say: He is Allah, the One! (1) Allah, the eternally Besought of all! (2) He begetteth not nor was begotten. (3) And there is none comparable unto Him. (4)

سُوْرَةُ الإخلاص
بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
قُلْ هُوَ ٱللَّهُ أَحَدٌ (١) ٱللَّهُ ٱلصَّمَدُ (٢) لَمْ يَلِدْ وَلَمْ يُولَدْ (٣) وَلَمْ يَكُن لَّهُۥ كُفُوًا أَحَدٌۢ (٤)

Figure 2. Chapter Al-Ikhlas in Englsih and Arabic

ID	Surah name	Verse number	Verse location	True or false	Error type	Error location	Error explanation	Er nu
8	Surah Alklas	3	ID8V3F.wav	1	نطق Pronunciation	يُولَدْ Begtten	خطأ في قلقلة حرف الدال Qalqalah error in Dall Letter	
8	Surah Alklas	1	ID86V1F.wav	1	نطق, نطق Pronunciation , Pronunciation	اللهُ, أَحَدٌ Allah , the one	نطق غير سليم في مخرج حرف الهاء، خطأ في قلقلة حرف الدال Incorrect pronouncing in Makhraj Haa' letter, Qalqalah error in Daal letter	1

Figure 3. One row sample of the final dataset

2.1.1 Dataset Description

A total of 1506 sounds of different reading of chapter Al-Ikhlas were collected and labeled in this work to facilitate the enhancement and rectification of individuals' proficiency in reading Classical Arabic. By automating the detection of prevalent errors encountered during the reading of Classical Arabic, this project aims to reduce reliance on experts and foster a more independent approach to learning.

The data were collected from August 1, 2021 to September 1, 2021, using Google Forms and Quranic audio websites, and then labeled manually.

Using Google Forms, 1230 were collected (for sentence 1(verse 1), 309 records were collected; for sentence 2, 311 records; for sentence 3, 315 records; and for sentence 4, 294 records were collected).

The remaining 276 records (69 for each sentence) were extracted using custom Python code from a Quranic audio website.

To facilitate the systematic analysis and organization of the data, dedicated table was created. This table was meticulously prepared and classified by a team of five highly knowledgeable experts in Classical Arabic. The purpose was to ensure accuracy and precision in the arrangement of the data records.

Figure 2 visually represents the four sentences that constitute chapter Al-Ikhlas, providing a clear and concise overview of their structure and content. This visual representation aids in understanding the context and relationships within the chapter.

The dataset itself consists of nine attributes, each playing a crucial role in characterizing the data. These attributes are thoughtfully illustrated in Figure 3, serving as a helpful reference for further analysis and interpretation.

The first attribute, known as "verse ID" assigns a unique identification number ranging from 0 to 1505 to each sentence(verse). This numbering system ensures the distinction and individuality of each sentence(verse) within the dataset.

The second attribute is dedicated to capturing the "Surah name" (chapter name), which, in this case, exclusively holds the value "Al-Ikhlas." This attribute provides valuable information about the chapter associated with each sentence(verse), facilitating easy identification and categorization.

The third attribute is designed to store the "verse number" (sentence number), which can take on one of four possible values: 1, 2, 3, or 4. This attribute plays a crucial role in differentiating between the sentences(verses) within the chapter, enabling precise referencing and analysis.

The fourth attribute, known as "verse location," stores the link to the specific sentence(verse) within the local machine. This attribute enables easy access and retrieval of the verse for further analysis or reference.

The fifth attribute indicates whether an instance is correct (with a value of 0) or contains an error (with a value of 1). This attribute serves as a binary indicator, allowing for straightforward identification of instances that require attention or correction.

The sixth attribute determines the type of error present in the instance, specifically categorizing it as either a pronunciation and phonetics error or a harakat error. In the absence of an error, the attribute value remains zero (0), signifying the absence of any detected errors.

The seventh attribute specifies the exact location of the identified error within the sentence(verse). This attribute provides valuable insight into the specific spot or segment where the error occurs. If no error is present, the attribute value remains zero (0), indicating the absence of any identified errors.

The eighth attribute offers a detailed explanation of the detected error. This attribute provides a comprehensive description or analysis of the specific error, offering insights into the nature and characteristics of the identified issue.

Lastly, the ninth attribute, which represents an error number ranging from 0 to 39. This attribute plays a crucial role in categorizing the data into distinct error types and locations. Specifically, the dataset is divided into two main categories: correct readings and incorrect readings. The incorrect readings are further classified into 39 subcategories based on the type of error and its specific location. These subcategories are numbered and documented in Table 1, providing a comprehensive reference for understanding and analyzing the errors.

The error number attribute ranges from 1 to 39, allowing for easy identification and classification of specific errors within the dataset. In instances where no error is present (representing a correct reading), the attribute is assigned a value of zero (0).

Table 1. Correct reading and incorrect reading categories.

#	**Type of Error**
0	Correct recitation
1	Add Madd to the Allah word, Qalqalah error in Daal letter
2	Add Madd to H’aa’ letter, Qalqalah error in Daal letter
3	Re-read the verse clearly
4	Lightness error in Laam letter, Qalqalah error in Daal letter
5	Qalqalah error in Daal letter
6	Error in pronouncing of Sukoon
7	Error in pronouncing of Sukoon, incorrect pronouncing in Makhraj Haa’ letter, Qalqalah error in Daal lette
8	Error in pronouncing of Qaaf letter, Qalqalah error in Daal letter
9	Incorrect pronouncing in Makhraj Laam letter, Qalqalah error in Daal letter
10	Incorrect pronouncing in Makhraj Haa’ letter, Qalqalah error in Daal letter
11	Error in pronouncing of Hamzah Al-Wasl
12	Error in pronouncing of Hamzah Al-Wasl and incorrect pronunciation in Makhaarj Daal letter, Qalqalah error in Daal letter
13	Error in pronouncing of Qaaf letter, Qalqalah error in Daal letter
14	Error in pronouncing of Sukoon in Daal letter
15	Error in pronouncing of Damma in Haa’ letter, Qalqalah error in Daal letter
16	Qalqalah error in Daal letter, Qalqalah error in Daal letter
17	Error in pronouncing of Al-Kasrah in Laam letter and Qalqalah error in Daal letter, Qalqalah error in Daal letter
18	Qalqalah error in Daal letter, error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter
19	Error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter, Qalqalah error in Daal letter
20	Error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter,error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter
21	Error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter,error in pronouncing of Sukoon in Miim letter, Qalqalah error in Daal letter
22	Error in pronouncing of Al-Kasrah in Laam letter and Qalqalah error in Daal letter, error in Madd Laam letter and Qalqalah error in Daal letter
23	Izhar error in Miim letter
24	Izhar error in Miim letter, Qalqalah error in Daal letter
25	Izhar error in Miim letter, error in pronouncing of Tanwin, Qalqalah error in Daal letter
26	Izhar error in Miim letter, error in pronouncing of Sukoon in Noon letter, error in pronouncing of Damma in Faa’ letter and error in pronouncing of Tanwin, Qalqalah error in Daal letter
27	Izhar error in Miim letter, error in pronouncing of Damma in Faa’ letter
28	Izhar error in Miim letter, error in pronouncing of Damma in Faa’ letter and error in pronouncing of Tanwin, Qalqalah error in Daal letter
29	Izhar error in Miim letter, error in pronouncing of Damma in Faa’ letter, Qalqalah error in Daal letter
30	Error in pronouncing of Tanwin
31	Error in pronouncing of Tanwin, Qalqalah error in Daal letter
32	Error in pronouncing of Damma in Faa’ letter
33	Error in pronouncing of Damma in Faa’ letter and error in pronouncing of Tanwin
34	Error in pronouncing of Damma in Faa’ letter and error in pronouncing of Tanwin, Qalqalah error in Daal letter
35	Error in pronouncing of Damma in Faa’ letter, add Madd to H’aa’ and Qalqalah error in Daal letter
36	Error in pronouncing of Al-Fathah in Laam letter, error in pronouncing of Damma in Faa’ letter, Qalqalah error in Daal letter
37	Error in pronouncing of Al-Fathah in Yaa’ letter, error in pronouncing of Damma in Faa’ letter, Qalqalah error in Daal letter
38	Error of Madd in Haa’ letter, error in pronouncing of Damma in Faa’ letter and error in pronouncing of Tanwin, Qalqalah error in Daal letter
39	Qalqalah error in Daal letter in Yalid word

For more detailed information about the dataset, including its composition and structure, a comprehensive explanation can be found in the provided reference [10]. This resource offers in-depth insights into the dataset, enabling researchers and analysts to gain a comprehensive understanding of the data and its implications.

2.1.2 ML Algorithm and Evaluation Matrices

To achieve the highest possible accuracy, a comprehensive approach was adopted, employing six different algorithms:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Support Vector Machine (SVM)
4. Logistic Regression
5. K-Neighbors Classifier
6. Gradient Boosting Classifier

For each algorithm, models were created and trained individually for every sentence(verse) in the dataset. To ensure robust evaluation, various test sizes were employed, including 2, 3, 4, 25, 0.2, 0.10, 0.25, 0.30, and 0.40. Additionally, different random states (specifically, 42, 3, 0, and 1) were utilized to account for any potential variations.

Cross-validation, a statistical method used to assess and compare learning algorithms, was incorporated during the training process. This technique involves dividing the data into two segments: one for training the model and the other for validating the model's performance. By employing cross-validation, the models were rigorously evaluated to identify the optimal accuracy.

The combination of these six algorithms, along with extensive testing and cross-validation, aimed to identify the most effective approach for achieving accurate results in the classification task.

The accuracy of a model is computed by taking into account the sum of the elements true positive (TP) and true negative (TN) at the numerator, as well as the entries of the confusing matrix at the denominator. The two elements are placed on the main diagonal of the matrix and are correct for classification by the model. On the other hand, the denominator takes into account all the elements that have been incorrectly labeled by the model (FP and FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

After conducting the evaluations, it was observed that the Random Forest algorithm, without cross-validation, achieved the highest accuracy compared to the other algorithms. The best performance was achieved with a test size of 20% and a random state of 1.

Table 2. Evaluation of ML models.

Verse Number	Accuracy	recall	Precision	F1-Score
Decision Tree Classifier	55	55	56	55
Random Forest Classifier	81	80	81	80
Support Vector Machine (SVM)	76	76	77	76
Logistic Regression	65	65	66	65
K-Neighbors Classifier	68	68	69	68
Gradient Boosting Classifier	73	73	74	73

The Random Forest algorithm proved to be highly effective in accurately classifying the verses within the dataset, achieving an impressive overall accuracy rate of 81% as shown in Table 2.

In addition to accuracy, the performance of the classifier models was evaluated using precision, recall, and F1-score metrics. These metrics provide a comprehensive understanding of the models' effectiveness in correctly classifying the data. Table 2 presents the evaluation results of the models using these metrics.

The recall of a model was calculated by dividing the true positive (TP) elements by the sum of TP and false negative (FN) elements. It represents the ability of the model to correctly identify positive instances out of all the actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

The precision of a model was computed by dividing the TP elements by the sum of TP and false positive (FP) elements. Precision measures the accuracy of the model in correctly identifying positive instances out of all the instances classified as positive.

$$\text{Precision}=\frac{TP}{TP+FP} \qquad (3)$$

Lastly, the F1-score of a model is the harmonic average of precision and recall. It provides a balanced measure of the model's overall performance, taking into account both precision and recall.

$$\text{F1-score}=\frac{precision * recall}{precision + recall} \qquad (4)$$

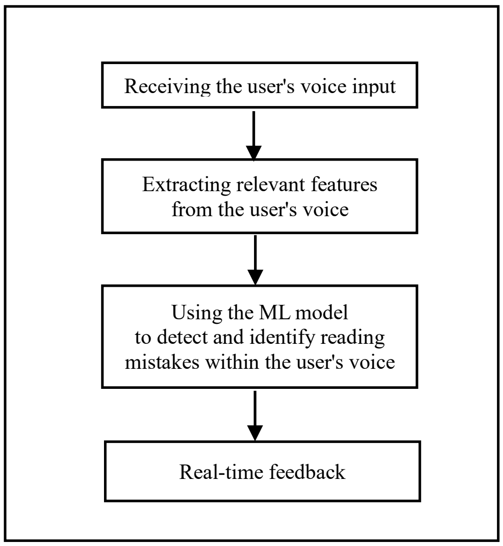


Figure 4. ML deduction process steps

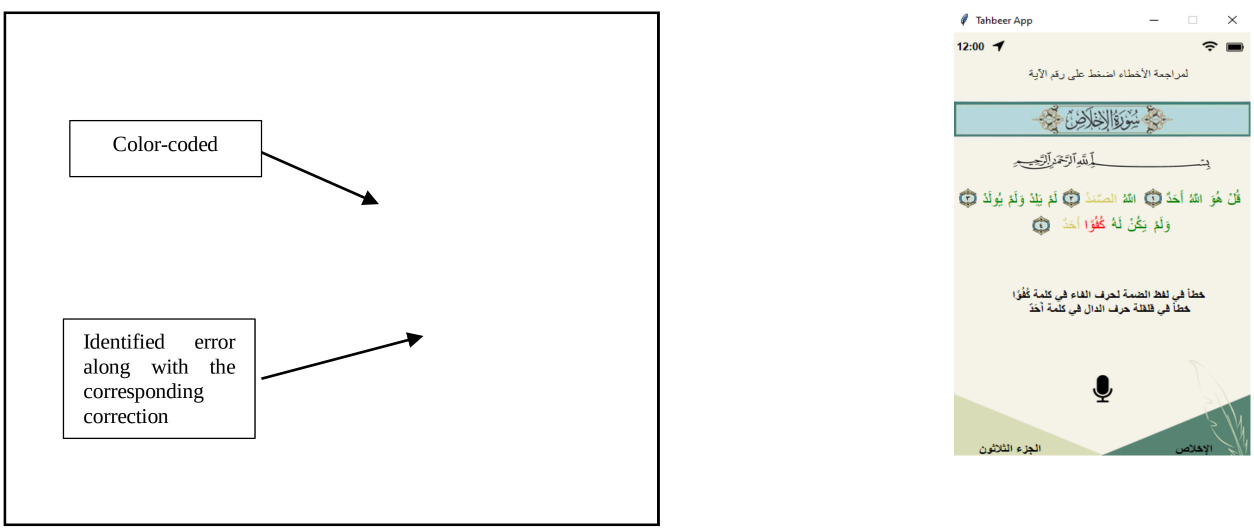


Figure 5. Detecting a mistake real-time feedback

2.1.3 ML Deduction Process Steps

Figure 4 provides an overview of the ML steps involved in the system. The process begins with the system receiving the user's voice input and detecting whether it contains any vocal content or is empty. If the input is determined to contain vocal content, it proceeds to save the user's voice as a (.wav) file format for further processing.

Next, the system extracts relevant features from the user's voice. Using the ML model, the system then employs classification techniques to detect and identify reading mistakes within the user's voice. The ML model has been trained to recognize specific patterns or deviations that indicate errors in the reading.

Upon detecting a mistake (as shown in Figure 5), the system offers real-time feedback to the user by displaying the identified error along with the corresponding correction. To enhance clarity, the system highlights errors in distinct colors. Specifically, errors related to Harakat (vowel marks), replacement, or missing words are highlighted in red. Pronunciation and phonetics related errors, on the other hand, are highlighted in yellow.

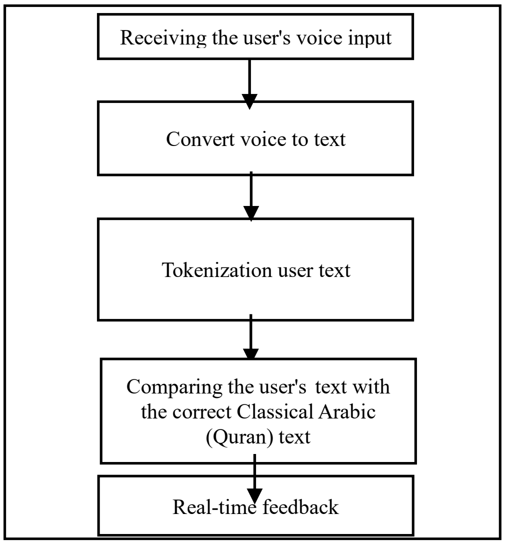


Figure 6. ML deduction process steps

This color-coded approach effectively draws the user's attention to the specific type of error they made, enabling them to easily identify and understand the nature of the mistake. This real-time feedback mechanism enhances the user's understanding and helps them develop accurate reading skills more effectively. It promotes a continuous learning process, enabling users to identify and rectify mistakes as they occur, leading to a more refined reading performance over time.

2.2. Error Detection by Natural Language Processing

The system employs natural language processing (NLP) techniques to address and resolve various types of mistakes, including word, or letter replacements and missing letters or words. These mistakes are resolved using a rule-based approach, leveraging the capabilities of NLP.

Through the integration of NLP techniques, the system offers a robust solution for detecting and correcting reading mistakes. By automating the error detection and correction process, users can receive accurate feedback and improve their reading skills accordingly.

Figure 6 provides an overview of the NLP steps involved in the system. The process begins with the system receiving the user's voice input and detecting whether it contains any vocal content or is empty. If the input is determined to contain vocal content, it proceeds utilizes the Google Speech API to convert the user's voice into text.

The converted text is then tokenized, which involves breaking it down into individual words or units. This tokenization process allows for a more detailed comparison and analysis of the user's text with the correct Classical Arabic (Quran) text.

By comparing the user's text with the correct Classical Arabic (Quran) text, the system can detect and identify mistakes made in reading. This comparison enables the system to pinpoint specific errors in the user's reading and apply appropriate corrections based on predefined rules.

When a mistake is detected (as shown in Figure 5), the system provides real-time feedback to the user by presenting the identified error and its corresponding correction and highlighted error in red.

3. Results and Discussion

The main objective of the project was to assist users in reading Classical Arabic accurately. To achieve this, a mobile-based application called "Tahbeer" was developed, which utilizes ML and NLP techniques in real time to correct the reader's Classical Arabic reading.

This project covers two types of mistake detection. The first type involves the use of ML to detect pronunciation, phonetics, and Harakat mistakes. The ML model is trained to identify errors in these areas.

The second type of mistake detection relies on NLP and text comparison techniques. NLP is utilized to detect missing, replacement, addition, and repetition mistakes in the text. This approach involves comparing the input text with a reference or correct version to identify discrepancies.

The Tahbeer system utilizes two different types of detection for specific reasons. Firstly, relying solely on NLP is not sufficient because there are no readily available tools that accurately convert Arabic speech to text while preserving the Harakat and pronunciation. Therefore, the system needs to incorporate Harakat and pronunciation detection, which can be effectively addressed using machine learning (ML) techniques.

On the other hand, relying solely on ML is also not ideal because the collected dataset might not encompass all possible mistakes, such as missing and replacement errors. Collecting a comprehensive dataset for these specific mistakes would require significant time and effort, as they are relatively rare occurrences.

By combining both NLP and ML approaches, the Tahbeer system can effectively cover a wider range of mistakes, including Harakat, pronunciation, missing, and replacement errors, providing a more comprehensive and accurate error detection capability for Arabic text.

In order to evaluate the accuracy of mistake detection, the study utilized six different algorithms applied to ML models (for more detailed information, please refer to section 2.1.2). Among these algorithms, the random forest algorithm demonstrated the highest level of accuracy, achieving an impressive overall accuracy rate of 81%. This finding underscores the effectiveness of the random forest algorithm in accurately detecting mistakes.

4. Conclusion

A system was developed to detect and correct reading mistakes in Classical Arabic. The system utilizes AI techniques to analyze the user's voice and applies both ML and NLP methods to classify reading mistakes. Through evaluation, it was found that the random forest classification algorithm provided the best results. In future iterations, there are several areas for improvement. Firstly, the dataset used for training and evaluating the ML detection models consisted of 1506 records. However, in order to further enhance the accuracy of these models, it would be beneficial to gather a new set of records specifically from non-Arabic speakers and children. This additional dataset would provide a more diverse range of samples which would significantly enhance the accuracy of the ML detection models. Secondly, the current version of the app is limited in its coverage, as it only includes one book chapter. To enhance its comprehensiveness, future efforts should be directed towards expanding the coverage to include multiple chapters and books. By doing so, users would be able to leverage the system's capabilities across a wider range of texts, enabling them to benefit from its functionalities in a more comprehensive manner.

Dr. Ashwag Maghraby received the B.S. (2002) and M.S. (2007) degrees in computer science from King Abdulaziz University, Jeddah University and the Ph.D. degree in Intelligent software engineering, The Centre for Intelligent Systems and their Applications (CISA) University of Edinburgh , 2013. Since 2013, she has been an Assistant Professor with the Department of Computers, Umm Al-Qura University, Mecca, Saudi Arabia. For the last three years her student researches awarded the best college research project in Umm Al-Qura University. Her research interests lie in the field of software and intelligent systems engineering, with a focus on utilizing natural language processing (NLP) and machine learning to enhance healthcare and improve people's daily lives.

Ms. Manar Alsaedi, Ms. Azzah Mahjoob, Ms. Amirah Al-Otaibi, Ms. Waad Alsaedi, and Ms. Atheer Alsaedi are graduates of Umm Al-Qura University, Mecca, Saudi Arabia, receiving their B.S. degrees in computer science in 2021. They possess expertise in data analysis and machine learning, along with a strong background in Classical Arabic.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Alhawarat M., “Processing the Text of the Holy Qur’an: A Text Mining Study,” *International Journal of Advanced Computer Science and Applications*, vol.6, no. 2, pp. 262- 267, 2015. http://dx.doi.org/10.14569/IJACSA.2015.060237

[2] Agency C.I., “The World Factbook”, [Online], Available: https://www.cia.gov/the-world-factbook/countries/world/

[3] Waqar M. Muhammad R., Muhammad A. and Martinez A., “Voice Content Matching System for Qur’an Readers,” *Ninth Mexican International Conference on Artificial Intelligence*, 2010. http://dx.doi.org/10.1109/MICAI.2010.11

[4] Satori H., Hiyassat H., Harti M., and Chenfour N., “Arabic Speech Recognition Using CMU Sphinx System,” *The International Arab Journal of Information Technology*, vol. 6, no. 2, pp.186-190, 2009. https://www.researchgate.net/publication/220413611_Investigation_Arabic_Speech_Recognition_Using_CMU_Sphinx_System

[5] Alobaylani A., Parvez M. and Alsuhibany S., “A Database for Speech Processing Based Qur’anic Recitation Correction,” *2020 International Conference on Computing and Information*

Technology, University of Tabuk, Kingdom of Saudi Arabia, pp. 1-5, 2022. https://doi.org/10.1109/ICCIT144147971.2020.9213826
[6] Software development company, "Tasmee," [Online], Available: http://tasmee.eqratech.com/
[7] More A., "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv*, vol.1, 2016. https://arxiv.org/abs/1608.06048
[8] Boughorbel S., Jarray F., and El-Anbari M., "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS ONE*, vol.12, no.6, 2017. https://doi.org/10.1371/journal.pone.0177678
[9] Khalilia M., Chakraborty S., and Popescu M., "Predicting disease risks from highly imbalanced data using random forest," *BMC Med Inform Decis Mak*, 2011. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-11-51
[10] Maghraby A., Mahjoob A., Alsaedi W., Alsaedi A., Alsaed M., and Alotaibi A., "Surah Al-Ikhlas of the Holy Qur'an Error Detection Dataset," Mendeley Data repository,2022.[unpublished] https://data.mendeley.com/datasets/sxtmmr6mvk/draft?a=91535a08-ae01-4f71-9daf-0c6b79fbd394
[11] Refaeilzadeh P., Tang L., and Liu H., "Cross-Validation," *Encyclopedia of Database Systems, Springer, Boston, MA*, pp.532-538, 2009. https://doi.org/10.1007/978-0-387-39940-9_565
[12] Miyasato K., "Classification report: Precision, recall, F1-score, accuracy," [Online], Available: https://medium.com/@kennymiyasato/classification-reportprecision-recall-f1-score-accuracy-16a245a437a5
[13] Agarwal R., "The 5 classification evaluation metrics every data scientist must know," [Online], Available: https://towardsdatascience.com/the-5-classificationevaluation-metrics-you-must-know-aa97784ff226

5/12/2024