Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033457
Article

Tech Science Press

check for updates

An End-to-End Transformer-Based Automatic Speech Recognition for Qur'an Reciters

Mohammed Hadwan[1,2,*], Hamzah A. Alsayadi[3,4] and Salah AL-Hagree[5]

[1]Department of Information Technology, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia
[2]Department of Computer Science, College of Applied Sciences, Taiz University, Taiz, 6803, Yemen
[3]Computer Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, 11566, Egypt
[4]Computer Science Department, Faculty of Sciences, Ibb University, Yemen
[5]Department of Computer Sciences & Information, Ibb University, Yemen
*Corresponding Author: Mohammed Hadwan. Email: m.hadwan@qu.edu.sa

Received: 17 June 2022; Accepted: 19 August 2022

Abstract: The attention-based encoder-decoder technique, known as the trans-former, is used to enhance the performance of end-to-end automatic speech recognition (ASR). This research focuses on applying ASR end-to-end transformer-based models for the Arabic language, as the researchers' community pays little attention to it. The Muslims Holy Qur'an book is written using Arabic diacritized text. In this paper, an end-to-end transformer model to building a robust Qur'an *vs.* recognition is proposed. The acoustic model was built using the transformer-based model as deep learning by the PyTorch framework. A multi-head attention mechanism is utilized to represent the encoder and decoder in the acoustic model. A Mel filter bank is used for feature extraction. To build a language model (LM), the Recurrent Neural Network (RNN) and Long short-term memory (LSTM) were used to train an n-gram word-based LM. As a part of this research, a new dataset of Qur'an verses and their associated transcripts were collected and processed for training and evaluating the proposed model, consisting of 10 h of .wav recitations performed by 60 reciters. The experimental results showed that the proposed end-to-end transformer-based model achieved a significant low character error rate (CER) of 1.98% and a word error rate (WER) of 6.16%. We have achieved state-of-the-art end-to-end transformer-based recognition for Qur'an reciters.

Keywords: Attention-based encoder-decoder; recurrent neural network; long short-term memory; qur'an reciters recognition; diacritized arabic text

1 Introduction

Nearly 300 million people are native speakers of the Arabic language, which is a member of the Semitic language family that includes other well-known languages such as Hebrew and Aramaic [1].

The Arabic alphabet consists of 28 letters, all of which represent consonants and are written from right to left. The three-letter root system is the most distinctive feature of Semitic languages. The pattern system is another important aspect of the Arabic language when a pattern is imposed on the fundamental root. For example, “wrote” and “writer” in the Arabic language is Katab “كتب” and Kateb "ك ت ب" which have the same three root letters that are "كاتب". For more details about Semitic languages refer to [1]. The Arabic script is a modified abjad [2] in which letters represent short consonants and long vowels, but short vowels and consonant length are not commonly shown in writing. Diacritics “Tashkeel” is an optional character that can be used to denote missing vowels and consonant length. The Arabic script contains various diacritics, which are critical in achieving typographic text usability standards such as homogeneity, clarity, and readability. Any modification to the letter’s diacritic of the word can radically alter its meaning [3].

The Holy Qur’an, Islam’s fundamental religious book, is divided into 114 chapters, each of which consists of verses. Besides its religious importance for Muslims, it is largely recognized as the best work in Arabic literature and has had a tremendous influence on the Arabic language [4,5]. Qur’an recitation “Qira’at” is a daily practice of Muslims as a part of their faith. The Qur’an was passed down from generation to generation through recitation. The Qur’an has seven canonical qira’at [6]. The correct recitation of the Holy Qur’an depends mainly on another discipline known as Tajweed. Tajweed [7] specifies the correct way of reciting the Qur’an, how to correctly pronounce each individual syllable, the pause places in Qur’an verses, in addition to elisions, where long or short pronunciation is needed, where letters should be kept separately, and where they should be sounded together, and so on. All Muslims around the world, be they Arabic native speaker or non-Arabic speakers, recite and listen to Qur’an in the Arabic language.

Automatic Speech Recognition (ASR) is the use of computers to recognize and process a person’s speech. In research and industry, there has been a substantial growth in interest in ASR mostly directed toward the English language. For the Arabic language, little attention is paid to exploring ASR where several attempts can be found in the literature such as in [8–11]. Published review papers of ASR for the Arabic language can be found in [10,12–14]. Because of the various drawbacks of traditional ASR models, the end-to-end model is an important research path in speech recognition. There are very few attempts to use ASR end-to-end transformer-based, as we only found two studies [15,16] published recently and none of them tackle the problem of Arabic diacritized texts. An attempt is made in [17] to explore the effect on recognition accuracy of ASR for Arabic diacritized and non-diacritized texts using convolution neural network (CNN) and long short-term memory (LSTM). The same Arabic diacritized and non-diacritized texts were used for the conducted experiments. The results showed that the word error rate (WER) for non-diacritized texts was 14.96% compared to 28.48% for the diacritized texts, the lower is the better. This proved that the diacritized Arabic text is challenging and needs to be explored deeply. As mentioned before, Qur’an is written using diacritics, which encouraged us to explore and propose the end-to-end transformer-based to solve this challenging problem.

To the best of researchers’ knowledge, the end-to-end transformer-based model was never proposed for ASR Qur’an reciters. To fill this gap, this research work proposes a novel deep learning model for the Qur’an *vs.* recognition with an end-to-end model.

The main contributions of this work are summarized as follows:

- A new deep learning approach of the Qur’an *vs.* recognition with an end-to-end model. The main idea of this model is to recognize the Arabic diacritized text.
- The end-to-end transformer architecture and multi-head attention are proposed to recognize the voice of Qur’an reciters.

- The look-ahead model is used to build word-based and character-based language models. These models are trained based on Recurrent Neural Network (RNN) and LSTM.
- A new dataset was collected and processed to be used for training and evaluating the proposed model and will be available publicly for further research.
- The presented model represents a state-of-the-art of ASR for the Holy Qur'an recognition.

This paper is organized as follows. Section 2 offers background information about the current research and Section 3 includes a description of the proposed model. In Section 4, the experimental setup and the used dataset are introduced. Then, Section 5 is devoted to the results and discussion. Finally, the conclusion of this research is in Section 6.

2 Related Work

In this section, the literature related to the proposed model is presented. The attention paid to the ASR methods focused to recognized the voice of Qur'an reciters.

Alkhateeb [18] used two classifiers, (i) K-Nearest Neighbors (KNN), and (ii) Artificial Neural Network (ANN) to recognize the Holy Qur'an reciters. MFCC is used to analyze the audio dataset. Pitch was used as a feature to train KNN and ANN. The results showed that ANN reached an accuracy rate up to 97.7% while KNN showed an accuracy rate of 97.03%.

Nahar et al. in [19] have used a recognition model to identify the "Qira'ah" (type of reading) from the related Holy Qur'an audio wave. The proposed model was created in 3 stages: (i) the extraction and labeling of MFCC features from an acoustic signal, (ii) training the SVM learning model with the identified features, and (iii) detecting "Qira'ah". With a success rate of 96 percent, the experimental findings demonstrated the power of the introduced SVM-based recognition model.

Lataifeh et al. [20] compared the performance of classical *vs.* deep-based classifiers. The study offers a comparison between the accuracy of the automatically proposed method in contradiction of human expert listeners' in recognizing reliable reciters from imitators. Results showed that the accuracy of selected classical and deep-based classifiers reached 98.6% compared to 61% of human experts. Arabic diversified dataset is introduced lately by Lataifeh et al. [21] to have a unified dataset that can be used to assess the introduced method and models for Qur'anic research.

Mohammed et al. in [22] provided a technique for Qur'an reciters rules recognition to detect the Medd rule and Ghunnah using phoneme duration. The used dataset was gathered from 10 Qur'anic reciters in order to compute the Medd and Ghunnah durations in the right recitation. The developed approach was then utilized to identify the Medd and Ghunnah as Tajweed norms in Quran recitation.

In Gunawan et al. [23], for Qur'anic reciter identification, the features of Mel Frequency Cepstral Coefficients (MFCC) were extracted from the recorded audio, and after training a Gaussian Mixture Model (GMM), Gaussian Supervectors (GSVs) were formed using model parameters such as the mean vector and the main diagonal of the covariance matrix. This model can be applied to protocol classification, feature learning, anomalous protocol identification, and unknown protocol classification. The researchers in [24] used a Support Vector Machine (SVM) and threshold scoring system to recognize different Tajweed rules automatically. 70- dimensional filter banks were used for feature extraction. A new dataset collected by the authors was used for the experiments and very promising results were obtained.

In [25], the authors used features such as MFCC and Pitch for learning process to recognize the Qur'an reciters. Several machine learning algorithms such as Random Forest, Naïve Bayes and J48 were used. The obtained results show the ability of proposed model to detect Qur'an reciter based on the used dataset. The best recognition accuracy was 88% when using Naïve Bayes.

Asda et al. in [26] proposed a reciters recognition system based on feature extraction from the MFCC and an ANN. a small dataset of five reciters was used for training and testing. They obtained 91.2% recognition accuracy for reciters. Bezoui et al. in [27] extracted characteristics from Quranic verse recitation using the MFCC approach. Preprocessing, framing, windowing, DFT, Mel Filter-bank, Logarithm, and Discrete Cosine Transform DCT are all part of MFCCs. An autonomous reciter recognition system based on text-independent speaker recognition approach employing MFCC is introduced by [28]. The dataset utilized a sample of 200 recordings made by 20 reciters. For clean samples, they achieved an 86.5% identification rate.

In [29], the Hidden Markov Model (HMM) algorithm approach is used in this work to develop a model utilizing the MFCC feature extraction. With conventional sentence percentages, HMM is utilized in reciters voice recognition. The dataset utilized in this study is voice data extracted from the voice of a known and related Quran reciter. The test results on the created model have an average percentage of test data correctness of 80%.

Based on Quranic recitations, researchers in [30] reported the construction of a recognizer for the allophonic sounds of Classical Arabic. The Cambridge HTK tools were used to create this recognizer. An acoustic HMM represented with three emission states is used to model each allophonic sound. For each emitting state, a continuous probability distribution based on 16 Gaussian mixed distributions is employed. The results demonstrate that without employing any special language model, identification rates attained a high degree of accuracy (88% on average), which is highly promising and encouraging.

We can conclude that, based on this comprehensive review of the ASR literature on Qur'anic reciters recognition, it is clear that no attempts have been made to use an end-to-end transformer-based model for recognizing and identifying Qur'an reciters. Therefore, we are trying to fill this gap in the literature by exploring the end-to-end transformer-based model for Qur'an reciters.

3 Theoretical Background

A convolution neural network (CNN) was established as a biologically inspired visual perception model. It can be used to feed ASR tasks to improve the recognition accuracy. Even though CNN leverages structural locality from the feature space to decrease spectral variation in acoustic features through pooling adaptation at a local frequency region, the CNN may identify the local structure in the input data. CNN can take advantage of long-term dependencies between speech frames by leveraging previous knowledge of speech signals. Furthermore, CNN has new properties above DNN, such as localization, weight sharing, and pooling. In the convolution unit, the locality is employed to handle noise where is used [31,32]. Additionally, locality minimizes the network weights that must be learned. Weight sharing with the locality is used to decrease translational variance. The same feature values computed at separate places are pooled together and represented by a single value in pooling. This is quite useful for dealing with tiny frequency shifts that are frequently available in speech signals. Other models, such as GMMs and DNNs, find it harder to manipulate this shifting process. As a result, ASR researchers have recently employed localization in both frequency and time axes in speech signals [31,32].

Chiu et al. [33] reported that CNN achieves 30% enhancement over GMMs and 12% enhancement over DNNs, using 700 h of speech data. Rao et al. [34] showed that the CNN-based end-to-end ASR approach has given promising results. Thus, we used a CNN-based end-to-end ASR approach for building the employed model. Researchers in [35] proposed a sequential CNN to test a collected dataset that includes 312 phonemes of the short vowel Arabic Alphabet/a/ "Alif". The obtained results revealed that CNN gave high accuracy of 100% and 0.27 of loss.

LSTMs are a form of RNN that is utilized for RNN evolution. The LSTM method can save information over a long period using long-term dependencies to find and exploit long-range contexts. The conventional RNN includes a single neural network, whereas the LSTM has four cooperating layers, each with its own communication link [12,36]. Researchers in [37] discussed the Siamese LSTM network used to authenticate the Qur'an recitation for testing Qur'an memorization. They contrasted the MaLSTM with the Siamese Classifier model. Several feature extraction approaches, such as MFCC, MFSC, and Delta, were investigated. The model was developed and tested using four readers' data who recited 48 verses from the Qur'an's final ten suras (chapters). The best model used the MaLSTM with an additional fully connected layer with MFCC and delta features and received an F1-score of 77.35%.

In ASR, we use the coming context as well if the transcription for all utterances is obtained at the training time. An LSTM calculates an input sequence X = x1, x3, ..., xT and the corresponding output sequence Y = y1, y2, ..., yL using the activation of network units is calculated given the T-length of the speech feature sequence ot−1, an LSTM is employed in the training stage with subsampling. It is utilized to create the following high-level feature h1:T0 as presented in Eq. (1).

$$h_t = LSTM\ \left(x_t, o_{t-1}\right) \tag{1}$$

where the subsampling is denoted by h. X is the input feature that will be handled to create the hidden states h_t based on the processes frame-wise. To reduce the computational cost, LSTM presents the outputs. Therefore, in ASR, the input length is not equivalent to the output length [31].

LSTM models are considered state-of-the-art ASR systems [32]. Moreover, deep LSTM networks achieve better performance and accuracy in ASR tasks [33,34]. For building the employed model, the LSTM-based end-to-end ASR method is used.

In encoder-decoder, all features are represented using a context vector. The context vector is used to generate the word sequence during the time. The attention-based model [35,36] is utilized to adjust the encoder-decoder model to become more accurate. It uses the attention layer to obtain a time-variant and dynamic context vector *ci*. This layer is located between the decoder and encoder [37]. The dynamic context vector is formulated as follows in Eq. (2).

$$c_i = \sum\nolimits_t \alpha_{it} v_t \tag{2}$$

where αi1, αi, ..., αit denote to the weights. These weights will be calculated in a dynamic process as follows in Eq. (3),

$$\alpha_{it} = \frac{\exp\ \left(s\left(v_t, u_{i-2}\right)\right)}{\sum_{k=1}^{T} \exp\left(s\left(v_k, u_{i-2}\right)\right)} \tag{3}$$

The attention score is expressed by $s(v_t; u_{(i-2)})$ notation, t denotes the input position, and *i* denotes the output position. The scoring function is implemented using the dot product or bi-linear form as in Eqs. (4) and (5), respectively.

$$s\left(v_t, u_{i-2}\right) = v_t^T u_{i-2} \tag{4}$$

$$s\left(v_t, u_{i-2}\right) = v_t^T M u_{i-2} \tag{5}$$

where M is the corresponding parameter.

The conditional probability of a word sequence is formulated as follows in Eq. (6):

$$P\left(w_{1:T}|x_{1:T}\right) \simeq \prod_{i=1}^{M} P\left(w_i|w_{i-1},\ u_{i-2}, c\right) \simeq \prod_{i=1}^{M} P\left(w_i|\ u_{i-1}\right) \tag{6}$$

The attention layer uses the context vector c_t to find the frames dynamically.

4 System Description

The Espresso toolkit is used to build the end-to-end model, which is based on the end-to-end transformer. Its underlying deep learning engine is called PyTorch. The stages for the employed model are depicted in Fig. 1, beginning with the preprocessing step of the corpus data, lexicon, and language modeling. The following step is the extraction of the features, followed by training and language modeling construction. Finally, multi-head attention is used for the encoder and decoder. The data preprocessing, features ex-traction, and language modeling are performed as presented in [17,38].

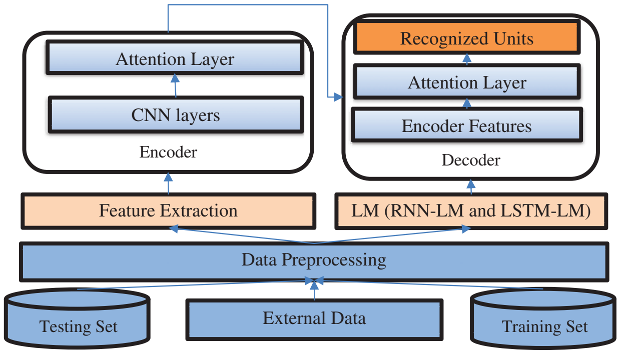

Figure 1: The steps of the proposed model

4.1 Feature Extraction

Features extraction is very important aspect for machine learning systems [39–44]. We utilize the Mel frequency filter bank (MFF Bank) technique to build the acoustic features. An MFF Bank is a method to simulate the human logarithmic audio perception which is considered a suitable method for ASR. The frequency is converted into Mel Scale for calculating the Mel frequency filter bank. Then, the Mel frequency cepstrum (MFC) is utilized to represent the short-term power spectrum of a phone using the log power spectrum transformation [38]. The filter bank is used to derive the MFC feature on top of the FFT. In general, the signal is a set of short frames of around 20 milliseconds. These frames will be split into values as frequency bands that represent the weights. In this work, the pitch features and the 40-dimensional Log Mel-filter bank (a total of 43 dimensions) from the raw speech data were generated. It is utilized for training the 10-hours dataset that is used in this research.

4.2 Language Modeling

ASR requires a Language Model (LM) to compute a priori probability of a word sequence. The acquired text data is utilized to train and create the proposed model's external LM. An external neural

LM, named the Look-ahead model [45], is utilized to generate the trained word-based and character-based language models. This external LM outperforms multilayer LMs [45]. Using prefix trees, a word-based LM is turned into a character-based model. The RNNLM and LSTM-LM are used to construct the word-based LM, with the RNN-LM calculating the probability of the following character based on all preceding words and the word's prefix. The LSTM-LM, on the other hand, is used to forecast word probability [46].

In the decoding step, the trained word-based and character LMs are combined with the estimated probabilities of the look-ahead word's prefix. In this research, the look-ahead word-based LM allows batches to make training faster than existing LMs. The look-ahead word-based LM probabilities are determined in each recognition phase according to the decoding of the word prefixes. The prefix trees are used to transform a word-based LM into a character-based LM [17,31]. We used a completely parallelized version of the decoding technique for GPUs that was presented in Espresso for Parallelization LM. The character-based LM in [45] is created from a word-based LM using the prefix tree technique.

4.3 End-to-end Transformer-based Architecture

The transformer is a new end-to-end model, which uses the encoder-decoder based to build ASR [47]. The transformer-based model uses the mechanism of self-attention to transform the sequences by applying attention matrices to the acoustic feature. In this work, we proposed a transformer-based model as a state-of-the-art model for Qur'an reciters recognition, as shown in Fig. 2. The employed transformer-based model comprises two parts: an encoder with a set of blocks and a decoder with a set of blocks.

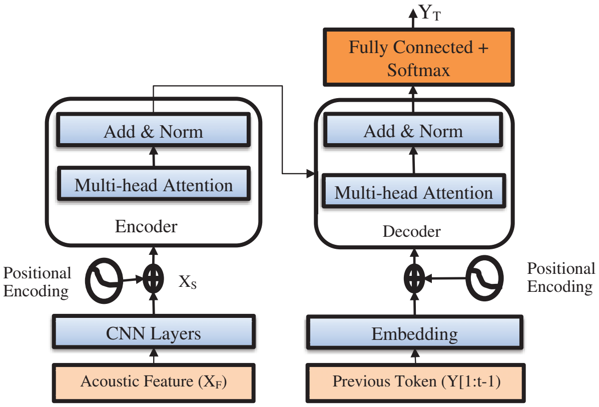

Figure 2: Transformer-based architecture

The encoder model converts the feature vector to an intermediate of encoded features. This feature vector represents an input of the encoder with 43-dimensional feature frames. Based on the encoded features and previous predictions, the decoder model is used to generate a new prediction. Both models include attention and feedforward network techniques. In addition, our ASR includes encoder frontend, positional encoding, ASR training, and ASR inference.

4.3.1 Encoder Frontend

In this step, we used CNN at the input layer to subsample the acoustic feature (X-fbank) into (X-sub). CNN consists of four layers, kernel size (3; 3), and convolution of 2-dimensions are used for each layer for the feature and time frame axis.

4.3.2 Encoder

The encoder contains a set of blocks with different layers: a multi-head self-attention mechanism and a position-wise feed-forward network. A layer normalization [35] is applied after each layer using residual connections. The subsampled sequences (X-sub), that are generated by the previous step, represent the input to the encoder blocks. The encoder transforms (X-sub) to (Q, K, V) using a self-attention layer with a Softmax as follows in Eq. (7):

$$Self\ Attention\left(Q,\ K,\ V\right) = softmax\left(\frac{Q * \ K^{T}}{\sqrt{d^{k}}}\right) * V \tag{7}$$

$\boldsymbol{Q}\ \in\ \mathbb{R}^{n^{q} * d^{q}}$, $\boldsymbol{K}\ \in\ \mathbb{R}^{n^{k} * d^{k}}$, ***and*** $\boldsymbol{Q}\ \in\ \mathbb{R}^{n^{v} * d^{v}}$ denote queries, keys, and values respectively. $\boldsymbol{d}^{*}$ is the dimensions of values, keys, and queries, and $\boldsymbol{n}^{*}$ is sequence lengths.

We used multi-head attention (MHA) to perform multiple attention networks. MHA yielded from all concatenated self-attention heads as follows in Eqs. (8) and (9):

$$MHA\left(Q,\ K,\ V\right) = \left[H_{1},\ H_{2}, \ldots H_{h}\right] W^{h} \tag{8}$$

$$H_{i} = \ \ Self\ Attention\left(Q_{i},\ K_{i},\ V_{i}\right) \tag{9}$$

where h denotes the attention heads number in a single layer and i is the i^{th} head in the layer. The MHA output is normalized before being sent into the Feed Forward (FF) sub-layer linked network, which is implemented for every point individually as in Eq. (10).

$$FF\left(h\left[t\right]\right) = \ \max\left(0,\ h\left[t\right] * \ W_{1} + b_{1}\right)\ W_{2} + b_{2} \tag{10}$$

where h[t] denotes the t^{th} position of the input H to the FF sublayer.

4.3.3 Decoder

The decoder is processed as the encoder. Besides, it obtains the probability of the next unit's sequence (YL) from the previous unit's sequence (YL-1) and the output of the encoder (Hi), like LM model. The decoder has multiple self-attention to transform the encoder features and previous units (YL-1) into prediction (YL) at each time. In the decoder, the attention between the encoder output sequence (feature) and the previous sequence is computed using MHA. The residual connections are used in each sublayer. The encoder and the decoder are trained efficiently as an end-to-end model.

4.3.4 Positional Encoding

The encoder features, the output of the encoder, are updated by positional encoding to address the positional context of units. Transformers utilize the sinusoidal positional encoding with time location as follows in Eqs. (11) and (12).

$$PE_{(n,2i)} = \sin\left(\frac{n}{1000^{\frac{2i}{d_{model}}}}\right) \tag{11}$$

$$PE_{(n,2i+1)} = \cos\left(\frac{n}{1000^{\frac{2i}{d_{model}}}}\right) \tag{12}$$

where n denotes a word's position in the text and i denotes the position with the dimension of the embedding vector.

4.3.5 Training of ASR Transformer

In the training step, the prediction of all unit frames is predicted by the acoustic model as P¬t(Y|X), where Y is the transcription of the units and X is the acoustic features. The training loss is calculated using the multi-objective function as follows in Eq. (13).

$$L_{asr} = -\ \log p_t\,(Y|X) \tag{13}$$

where Pt is the probability predicted by the transformer decoder.

MHA does not calculate the values of the lower triangular in the attention matrix. Thus, the output sequence does not send into positions in the query sequence.

4.3.6 Transformer ASR Decoding

In the decoding step, we integrate the end-to-end model with the employed language model (LM) to enhance the predicted units. The external LM is utilized in this paper based on RNN and LSTM. The end-to-end transformer and shallow fusion are combined and computed based on the two posterior distributions summed over units as follows in Eq. (14):

$$Y_{final} = \log P_{asr} + \lambda \log P_{lm} \tag{14}$$

For every timestamp t, the prediction, $\mathrm{Y}_{\mathrm{final}}$, is created by interpolating the distribution over vocabulary (V) given by log PASR and the same distribution given by log P_{lm}. In addition, we apply for the coverage and end-of-sentence threshold technique used in inference to improve the performance as presented in [17,38].

5 Experimental Setup and Dataset

A novel end-to-end transformer model is proposed for Qur'an *vs.* recognition. The proposed model is trained and evaluated using a collected dataset of Qur'an verses described in Section 4.1. LM is trained and evaluated on a dataset from several websites. For evaluation purposes, the traditional character error rate (CER) and word error rate (WER) are used to report the accuracy of recognition, while perplexity and out of vocabulary (OOV) are reported for LM. A Python code for data preprocessing is written. In addition, the Espresso toolkit is used to write the recipe for implementing the entire models. Experiments are conducted using a laptop with GeForce GTX 1060 6 GB as GPU, Intel i7-8750H as CPU, 16 GB as RAM, and CUDA version 10.0.

5.1 End-to-end Transformer-based Architecture

For the Qur'an dataset used in this research, 10 h of mp3 Qur'an verses recited by 60 reciters were collected from the A2Youth.com website and its associated transcripts. The collected dataset is used for training and evaluating the proposed models. Tab. 1 shows more details about the dataset, including the suras names, the average number of sounds, #wav files, and the total minutes. The dataset is distributed into three parts: (i) 70% for the training set, (ii) 10% for the development set, and (iii) 20 for the testing set.

Extensive preprocessing occurs to the collected dataset as follows:

1- The original audio files are converted from .mp3 format into .wav format.
2- All audio file are resampled into 16 kHz.
3- All transcript files (the text that corresponds to the audio file) are converted to Buckwalter format.

where Buckwalter transliteration may be thought of as the binary code for English, making it simple for machines to parse.

Table 1: The dataset details

Suras name	Average of seconds	#wav files	Total of minutes
Al-Zalzala	55	60	55
Al-Adiyat	58	60	58
Al-Qaria	59	60	59
Al-Takathur	47	60	47
Al-Asr	25	60	25
Al-Humaza	48	60	48
Al-Fil	36	60	36
Quraish	31	60	31
Al-Maun	37	60	37
Al-Kauther	22	60	22
Al-Kafiroon	39	60	39
An-Nasr	28	60	28
Al-Masadd	34	60	34
Al-Ikhlas	20	60	20
Al-Falaq	30	60	30
An-Nas	33	60	33
Total	**602**	**960**	**602**

5.2 End-to-End Transformer-Based Model Configuration

In this section, the proposed model configuration is presented, which includes different steps as follows. In the preprocessing step, a Python code is written to convert audio files from mp3 into wav type and unify the sample rate of all audio files. Then, we wrote Python code to convert corpus, lexicon, LM data to Buckwalter format, and link each transcript in our corpus with its audio file to make them suitable for the Espresso recipe. The Espresso toolkit uses Kaldi-format to do the training, testing, wav.scp, and utt2spk files. All data are cleaned by deleting numbers, extra empty spaces, newlines, and single-character words.

For generating the acoustic features step, the Kaldi-format for feature extraction is used. 40-dimensional log Mel-frequency filter bank features and pitch features (3) are calculated every 20 milliseconds (ms) from the raw speech data. The output of this stage will be sent into the data conversion stage to store all data in one JSON file. This file represents the input of the acoustic model. Then, the n-gram RNN-LM and LSTM-LM were trained using the training and collected data. This data consists of 250 k words and 50 k sentences. The LM is trained based on the Pytorch library using 3-layer LSTM with 1200 units for every layer. Moreover, the LM is trained using 35 epochs with 2000 as batch-size and 40 as the max length of the string. In addition, we use the stochastic gradient descent (SGD) optimizer for training purposes.

In the pre-encoder step, the encoder receives the feature vectors with 43 as the dimension. Then, the CNN is used to prepare the input of the encoder. CNN includes 4 convolutional layers with (3; 3) kernels for each layer. In end-to-end transformer training step, the acoustic model is employed by 6 encoders, where 12 encoder blocks are set for each employed encoder. The model dimension is configured by d-model = 512 and multi-head attention blocks use 4 attention heads. In addition, the acoustic model includes 6 decoders and multi-head attention blocks that use 4 attention heads. The scheduled sampling is implemented by $p = 0.5$ to adapt epochs starting from epoch 6. The temporal smoothing schema with $p = 0.05$ was applied to help the model ignore a sub-unit in the transcript depending on the errors of beam search. The Adam optimizer has been used in this model as an optimization method.

In the recognition step, we combined the external LM and shallow fusion to enhance the performance and accuracy. In addition, we added the end-of-sentence threshold and coverage methods for improving the quality of the Arabic ASR approach. Each model is with 10 k decoded as the optimal batch size. For enhancing the accuracy and performance, The LM fusion weight is allotted by 0.45 as an optimal parameter and combined with the look-ahead word-based LM. Moreover, assigned the end-of-sentence (EOS) threshold by 1.5 and beam size by 20 as the optimal size.

5.3 Evaluation Metrics

The accuracy performance metric is used for evaluating the performance of ASR approaches. Moreover, the perplexity metric is used for evaluating the performance of LM. This section describes these evaluation metrics in detail.

5.3.1 ASR Evaluation

The performance evaluation of ASR is usually presented in terms of two criteria: (1) Character Error Rate (CER), which represents the percentage of the character-level errors of the recognized units, and (2) Word Error Rate (WER), which represents the percentage of the word-level errors of the recognized units. These criteria are defined as follows in Eqs. (15) and (16):

$$CER = \frac{S + D + I}{N} \times 100 \tag{15}$$

$$WER = \frac{S + D + I}{N} \times 100 \tag{16}$$

where N represents all words in the set of evaluation utterances, substitutions (S) denote the number of misrecognized words, deletions (D) denote the number of deleted words in the recognition result, and I is the number of inserted words in the recognition result.

5.3.2 Language Model Evaluation

The performance evaluation of LM uses the perplexity and OOV measures. If set M contains all of the tokens in LM data and set T contains all tokens in the test data, then OVV is the number of tokens in T's complement divided by the number of tokens in T according to [17,31], as shown below in Eqs. (17) and (18).

$$OOV = \frac{\#token\ in\ \text{complement of T}}{\#token\ in\ T} \tag{17}$$

$$Perplexity = \left(\prod_{i=1}^{K} P\left(token_i|\ token_{j<i}\right)\right)^{-\frac{1}{k}} \tag{18}$$

where P (token i| token j < I) denotes the probability of i^{th} throw of the training of LM based on the first i − 1 tokens.

6 Results and Discussion

The employed end-to-end transformer is a state-of-the-art model for Qur'an *vs.* recognition. ASR in this research recognizes the reciters' voice-through two main parts-the Encoder and the decoder. In the encoder, the features were extracted and passed to the CNN layers and then to the attention layers before sending them to the decoder. While in the decoder the RNN-LM and LSTM-LM were used to encode the features and pass them to the attention layer to reach the units that recognized the reciters' voices. Experimental results of this model on a collected verses dataset shows the outstanding performance of the proposed model. The used dataset comprises 60 reciters with 16 verses for each reciter. To achieve the best result, we trained the model many times with different sizes of encoder and decoder, size of layers, learning weights, epochs, and smoothing parameters. In addition, the inference is performed with different LM weights, EOS, and the size of the beam. Experimental results are conducted twice: i) completely, for the development set and testing set; and ii) partially, for each verse in the development set and testing set. In addition, we evaluated LM based on testing data.

The developed LM is evaluated based on the test data. Tab. 2 shows the perplexity and OOV results for the language model.

Table 2: Perplexity and OOV results

Perplexity	OOV
1.36	1.07

The transformer-based model with/without LM is evaluated for a complete development set and testing set. Tab. 3 shows CER and WER results on the complete development set and testing set using a transformer-based model without LM.

Table 3: The CER and WER results of the transformer-based model without LM

Model	Development set		Testing set	
Transformer-based without LM	CER	WER	CER	WER
	3.64	6.72	3.83	12.52

According to Tab. 3, the transformer-based model without LM yields results of 3.64% and 3.83% of CER for the development set and testing set, respectively. Moreover, the model achieves results of 6.72% and 12.52% of WER for the development set and testing set, respectively. Tab. 4 shows CER and WER results on the complete development set and testing set using the transformer-based model with LM.

Table 4: The CER and WER results of the transformer-based model with LM

Model	Development set		Testing set	
Transformer-based with LM	CER	WER	CER	WER
	1.58	3.83	1.98	6.16

According to Tab. 4, the transformer-based model with LM yields results of 1.58% and 1.98% of CER for the development set and testing set, respectively. The model with LM achieves results of 3.83% and 6.16% of WER for the development set and testing set, respectively. Thus, the model with LM obtains the best results. Based on the results in Tab. 4, it is clearly that the CER of the model with LM is reduced by 2.06% for the development set and 1.85% for the testing set when compared with the same model but without LM. Moreover, the WER for the model with LM is 2.89% as for the development set and 6.36% as WER for testing set gain on top of the model without LM. The transformer-based model with/without LM for the complete and partial development set and testing set is shown in Tab. 5 for all tested verses.

Table 5: The CER and WER results of the transformer-based model with/without LM

Verse name	Model without LM		Model with LM	
	CER	WER	CER	WER
Al-Zalzala	1.78	4.50	0.84	1.88
Al-Adiyat	5.72	14.23	2.95	4.67
Al-Qaria	5.91	15.88	3.36	5.29
Al-Takathur	0.30	1.78	0.0	0.0
Al-Asr	1.64	4.63	0.42	1.47
Al-Humaza	2.76	5.41	1.18	2.53
Al-Fil	2.07	5.09	0.95	2.32
Quraish	2.89	6.03	1.49	3.18
Al-Maun	3.09	9.07	1.81	4.73
Al-Kauther	0.28	1.02	0.0	0.0
Al-Kafiroon	8.06	26.61	4.74	11.55
An-Nasr	6.90	33.91	2.93	16.52
Al-Masadd	4.04	12.82	2.21	6.23
Al-Ikhlas	3.69	10.23	1.94	6.56
Al-Falaq	7.74	38.75	4.83	25.66
An-Nas	4.43	10.42	2.03	5.97

Based on Tab. 5, it is noticed that the Al-Takathur and Al-Kauther verses have the best CER and WER which got 0.0%, while Al-Asr and Al-Zalzala verses have 1.47% and 1.88% for WER, respectively. In addition, Al-Falaq verse has the worst WER result 25.66%. Also, the employed model achieved 25.66% for Al-Takathur and Al-Kauther gained top on the worst result.

7 Conclusion

In this research, a high-performance approach for Qur'an verses recognition is presented. A transformer-based model is proposed to develop this approach using an end-to-end model. This model is a new deep learning model to recognize the Arabic diacritized speech. The Mel filter bank with 40-dimensional for building acoustic features was utilized. A CNN is used as an encoder frontend at the input layer for subsampling the acoustic feature. The acoustic model is built as an encoder-decoder model using multi-head attention. In addition, the look-ahead model is used to employ the word-based and character-based language models language modeling (LM). We have used RNN to train and build RNN-LM and LSTM-LM. The employed model represents the state-of-the-art on Qur'an verses recognition. We presented a new Qur'an verse dataset including 10 h of Qur'an verses recited by 60 reciters. The proposed model was trained and evaluated based on a collected dataset of Qur'an verses. The results obtained are 1.98% for CER and 6.16% for WER which show the power of the proposed model for verses recognition. For further improvements, the researchers have the following suggestions: (i) applying on-the-fly for feature extraction, (ii) applying transducer model training and decoding on the employed model and (iii) investigating the suitability of the proposed models on larger datasets.

Acknowledgement: The authors would like to thanks the Chair of Prince Faisal for Artificial intelligence research (CPFAI) for funding this research work through the project number QU-CPFAI-2-10-5. Also would like to extend their appreciation to the Deputyship for Research& Innovation, Ministry of Education and the Deanship of Scientific Research, Qassim University for their support for this research.

Funding Statement: The research work was supported by the Chair of Prince Faisal for Artificial Intelligent research (CPFIA), Qassim University through the Project Number QU-CPFAI-2-10-5.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Weninger, *The Semitic Languages an International Handbook*, Berlin, Germany: De Gruyter Mouton, 2011. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/9783110251586/html?lang=en.

[2] N. Alsunaidi, L. Alzeer, M. Alkatheiri, A. Habbabah, M. Alattas *et al.,* "Abjad: Towards interactive learning approach to Arabic reading based on speech recognition," *Procedia Computer Science*, vol. 142, no. 1, pp. 198–205, 2018.

[3] H. Mohamed and A. Lazrek, "Design of Arabic diacritical marks," *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 262–271, 2011.

[4] K. Y. Jung, "The linguistic impact of the Quran on Arabic," *Arabic Language&Literature*, vol. 17, no. 1, pp. 1–20, 2013.

[5] A. J. Arberry, "The Koran interpreted: A translation," *Journal of the American Oriental Society*, vol. 85, no. 2, pp. 289–298, 1965.

[6] M. A. S. Khalil and N. H. Yusof, “The difference in Qur’anic readings in the interpretation of Al-Tabari and its effect on jurisprundential rulings: An analytical study,” *Jurnal Islam dan Masyarakat Kontemporari*, vol. 16, no. 1, pp. 111–126, 2018.

[7] A. H. Ishaq and R. Nawawi, “Ilmu Tajwid dan implikasinya terhadap ilmu qira’ah,” *QOF*, vol. 1, no. 1, pp. 15–24, 2017.

[8] I. K. Tantawi, M. A. M. Abushariah and B. H. Hammo, “A deep learning approach for automatic speech recognition of the Holy Qur’ān recitations,” *International Journal of Speech Technology*, vol. 24, no. 4, pp. 1017–1032, 2021.

[9] H. Tabbal, W. El Falou and B. Monla, “Analysis and implementation of a Quranic verses delimitation system in audio files using speech recognition techniques,” in *Int. Conf. on Information and Communication Technologies: From Theory to Applications, ICTTA 2006*, Damascus, Syria, pp. 2979–2984, 2006.

[10] N. O. Balula, M. Rashwan and S. Abdou, “Automatic speech recognition (ASR) systems for learning Arabic language and Al-Quran recitation: A review,” *International Journal of Computer Science and Mobile Computing*, vol. 10, no. 7, pp. 91–100, 2021.

[11] F. Thirafi and D. P. Lestari, “Hybrid HMM-BLSTM-based acoustic modeling for automatic speech recognition on Quran recitation,” in *The 2018 Int. Conf. on Asian Language Processing, IALP 2018, Institute of Electrical and Electronics Engineers Inc.*, Bandung, Indonesia, pp. 203–208, 2019.

[12] A. A. Abdelhamid, H. Alsayadi, I. Hegazy and Z. T. Fayed, “End-to-end Arabic speech recognition: A review,” in *The 19th Conf. of Language Engineering (ESOLEC’19)*, Alexandria, Egypt, 2020.

[13] S. R. Shareef and Y. F. Irhayim, “A review: Isolated Arabic words recognition using artificial intelligent techniques,” *Journal of Physics: Conference Series*, vol. 1897, pp. 1–13, 2021.

[14] N. J. Ibrahim, M. Y. I. Idris, M. Y. Z. M. Yusoff and A. Anuar, “The problems, issues and future challenges of automatic speech recognition for Quranic verse recitation: A review,” *AlBayan*, vol. 13, no. 2, pp. 168–196, 2015.

[15] A. Hussein, S. Watanabe and A. Ali, “Arabic speech recognition by end-to-end, modular systems and human,” *Computer Speech and Language*, vol. 71, no. 1, pp. 1–39, 2022.

[16] W. Lin, M. Madhavi, R. K. Das and H. Li, “Transformer-based Arabic dialect identification,” in *2020 Int. Conf. on Asian Language Processing, IALP*, Kuala lumpur, Malaysia, pp. 203–208, 2020.

[17] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy and Z. T. Fayed, “Non-diacritized Arabic speech recognition based on CNN-LSTM and attention-based models,” *Journal of Intelligent and Fuzzy Systems*, vol. 41, no. 6, pp. 1–13, 2021.

[18] J. H. Alkhateeb, “A machine learning approach for recognizing the Holy Quran reciter,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 268–271, 2020.

[19] K. M. O. Nahar, R. M. Al-Khatib, M. A. Al-Shannaq and M. M. Barhoush, “An efficient Holy Quran recitation recognizer based on SVM learning model,” *Jordanian Journal of Computers and Information Technology*, vol. 6, no. 4, pp. 392–414, 2020.

[20] M. Lataifeh, A. Elnagar, I. Shahin and A. B. Nassif, “Arabic audio clips: Identification and discrimination of authentic cantillations from imitations,” *Neurocomputing*, vol. 418, no. 2, pp. 1–48, 2020.

[21] M. Lataifeh and A. Elnagar, “Ar-DAD: Arabic diversified audio dataset,” *Data in Brief*, vol. 33, no. 1, pp. 162–177, 2020.

[22] A. Mohammed, M. S. B. Sunar and M. S. H. Salam, “Recognition of Holy Quran recitation rules using phoneme duration,” *Lecture Notes on Data Engineering and Communications Technologies*, vol. 5, no. 1, pp. 1–12, 2018.

[23] T. S. Gunawan, N. A. M. Saleh and M. Kartiwi, “Development of Quranic reciter identification system using MFCC and GMM classifier,” *International Journal of Electrical and Computer Engineering*, vol. 8, no. 1, pp. 372–378, 2018.

[24] A. M. Alagrami and M. M. Eljazzar, “SMARTAJWEED automatic recognition of Arabic Quranic recitation rules,” in *Int. Conf. on Computer Science, Engineering and Applications*, London, United Kingdom, pp. 145–152, 2020.

[25] R. U. Khan, A. M. Qamar and M. Hadwan, "Quranic reciter recognition: A machine learning approach," *Advances in Science, Technology and Engineering Systems*, vol. 4, no. 6, pp. 173–176, 2019.

[26] T. M. H. Asda, T. S. Gunawan, M. Kartiwi and H. Mansor, "Development of Quran reciter identification system using MFCC and neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 1, no. 1, pp. 168–175, 2016.

[27] M. Bezoui, A. Elmoutaouakkil and A. Beni-Hssane, "Feature extraction of some Quranic recitation using Mel-Frequency Cepstral Coeficients (MFCC)," in *Int. Conf. on Multimedia Computing and Systems*, Marrakech, Morocco, pp. 127–131, 2017.

[28] M. A. Hussaini and R. W. Aldhaheri, "An automatic qari recognition system," in *Int. Conf. on Advanced Computer Science Applications and Technologies, ACSAT 2012*, NW Washington, DC, United States, pp. 524–528, 2012.

[29] O. V. Putra, F. R. Pradana and J. I. Q. Adiba, "Mad reading law classification using Mel Frequency Cepstal Coefficient (MFCC) and Hidden Markov Model (HMM)," *Procedia of Engineering and Life Science*, vol. 2, no. 1, pp. 1–7, 2021.

[30] Y. O. M. Elhadj, M. Alghamdi and M. Alkanhal, "Approach for recognizing allophonic sounds of the classical Arabic based on Quran recitations," *Theory and Practice of Natural Computing*, vol. 8273, no. 1, pp. 57–67, 2013.

[31] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy and Z. T. Fayed, "Arabic speech recognition using end-to-end deep learning," *IET Signal Processing*, vol. 15, no. 8, pp. 521–534, 2021.

[32] H. Sak, A. Senior and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. of the Annual Conf. of the Int. Speech Communication Association, INTERSPEECH*, Brno, Czech Republic, pp. 1–5, 2014.

[33] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen *et al.,* "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP, IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, Alberta, Canada, pp. 1–5, 2018.

[34] K. Rao, H. Sak and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017*, Okinawa, Japan, pp. 193–199, 2017.

[35] W. Chan, N. Jaitly, Q. Le and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP, IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Shanghai, China, pp. 4960–4964, 2016.

[36] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, Montreal, Canada: MIT Press, pp. 1–9, 2015.

[37] C. Wu, "Structured deep neural networks for speech recognition," Ph.D. Dissertation, University of Cambridge, United Kingdom, 2018.

[38] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[39] E. -S. M. El-kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *International Journal of Innovative Computing, Information & Control*, vol. 16, no. 1, pp. 831–844, 2020.

[40] A. Takieldeen, E. El-kenawy, M. Hadwan and M. Zaki, "Dipper throated optimization algorithm forunconstrained function and feature selection," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1465–1481, 2022.

[41] M. M. Eid, E. -S. M. El-Kenawy and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," in *4th National Computing Colleges Conf. (NCCC 2021)*, Taif, Saudi Arabia, IEEE, pp. 1–6, 2021.

[42] S. S. M. Ghoneim, T. A. Farrag, A. A. Rashed, E. -S. M. El-Kenawy and A. Ibrahim, "Adaptive dynamic meta-heuristics for feature selection and classification in diagnostic accuracy of transformer faults," *IEEE Access*, vol. 9, pp. 78324–78340, 2021.

[43] D. S. Khafaga, A. A. Alhussan, E. M. El-kenawy, A. E. Takieldeen, T. M. Hassan *et al.,* "Meta-heuristics for feature selection and classification in diagnostic breast cancer," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 749–765, 2022.

[44] E. -S. M. El-Kenawy, S. Mirjalili, F. Alassery, Y. Zhang, M. Eid *et al.,* "Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems," *IEEE Access*, vol. 10, pp. 40536–40555, 2022.

[45] T. Hori, J. Cho and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," in *2018 IEEE Spoken Language Technology Workshop, SLT*, Athens, Greece, pp. 389–396, 2019.

[46] Y. Wang, T. Chen, H. Xu, S. Ding, H. Lv *et al.,* "Espresso: A fast End-to-end neural speech recognition toolkit," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019*, Sentosa, Singapor, pp. 1–8, 2019.

[47] L. Dong, S. Xu and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP, IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, pp. 5884–5888, 2018.