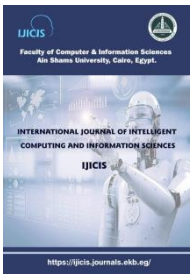
IJICIS, Vol.21, No.2, 50-64
DOI: 10.21608/ijicis.2021.73581.1086

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



Data Augmentation for Arabic Speech Recognition Based on End-to-End Deep Learning

Hamzah A. Alsayadi* Abdelaziz A. Abdelhamid Islam Hegazy Zaki T. Fayed

Computer Science Dept., Faculty of Computer & Information Sciences,Ain Shams University, Cairo, Egypt
hamzah.sayadi@cis.asu.edu.eg

Computer Science Dept., Faculty of Computer & Information Sciences,Ain Shams University, Cairo, Egypt
abdelaziz@cis.asu.edu.eg

Computer Science Dept., Faculty of Computer & Information Sciences,Ain Shams University, Cairo, Egypt
islheg@cis.asu.edu.eg

Computer Science Dept., Faculty of Computer & Information Sciences,Ain Shams University, Cairo, Egypt
ztfayed@hotmail.com

Received 2021- 4-22; Revised 2021-7-13; Accepted 2021-7-13

Abstract: *End-to-end deep learning approach has greatly enhanced the performance of speech recognition systems. With deep learning techniques, the overfitting stills the main problem with a little data. Data augmentation is a suitable solution for the overfitting problem, which is adopted to improve the quantity of training data and enhance robustness of the models. In this paper, we investigate data augmentation method for enhancing Arabic automatic speech recognition (ASR) based on end-to-end deep learning. Data augmentation is applied on original corpus for increasing training data by applying noise adaptation, pitch-shifting, and speed transformation. An CNN-LSTM and attention-based encoder-decoder method are included in building the acoustic model and decoding phase. This method is considered as state-of-art in end-to-end deep learning, and to the best of our knowledge, there is no prior research employed data augmentation for CNN-LSTM and attention-based model in Arabic ASR systems. In addition, the language model is built using RNN-LM and LSTM-LM methods. The Standard Arabic Single Speaker Corpus (SASSC) without diacritics is used as an original corpus. Experimental results show that applying data augmentation improved word error rate (WER) when compared with the same approach without data augmentation. The achieved average reduction in WER is 4.55%.*

Keywords: *Arabic Speech Recognition, Data Augmentation, CNN-LSTM, RNN-LM, Attention-based Model.*

1. Introduction

* Corresponding author: Hamzah A. Alsayadi
Computer Science Dept., Faculty of Computer & Information Sciences,Ain Shams University, Cairo, Egypt
E-mail address: hamzah.sayadi@cis.asu.edu.eg

Automatic Speech Recognition (ASR) is a task used to convert speech waves or signals to its mapping sequence of words or units using a determined algorithm [1]. These sequences are represented like the human transcription. The observations of speech vectors are used for representing input of speech audios [2]. ASR has a wide area of information technology (IT) applications; it employs in many IT-solutions and applications for civil and industrial areas such as; *Human-Computer Interaction* (HCI), voice applications, automatic language translation, and many via-voice systems [3]. ASR plays an important role to help persons to understand each other from a different society. In addition, it is a technology that makes disabled people communicate with society. It can able to make life easier and very promising [4].

End-to-end ASR is an approach which is used to build ASR system in one package. It maps the acoustic features with transcriptions directly [5]. The main purpose of end-to-end ASR is to simplify building ASR systems using one network hierarchy within new techniques of deep learning. The models of end-to-end ASR used only acoustic and language data and do not need the linguistic knowledge. These models will be trained using a single algorithm [5]. Therefore, end-to-end approach allows us to build ASR systems with little knowledge of speech processing. The end-to-end ASR architecture has various types such as recurrent neural network (RNN), convolutional neural network (CNN), long short-term memory network (LSTM), attention-based model, and hybrid models [6].

CNN is an appropriate technique for ASR in order to decrease word error rate (WER) [7]. Local correlations and spectral variations properties in speech signal make CNN is good choice for ASR. A CNN has new locality and weight sharing properties that are used for processing the noise and shifting the frequency in features [8]. LSTM have been successfully applied for ASR to improve the results and performance [8]. LSTM includes a memory block for handling the problem of time dependencies learning that is obtained from vanishing and exploding gradient methods [7, 8]. Thus, the hybrid CNN-LSTM is state-of-the-art for Arabic ASR systems.

There are ASR systems are built for Arabic language. Arabic language is one of the largest Semitic languages which still used widely. There are about 300 million that speak it as a native language. Arabic language is the fourth language in the world due to the number of Arabic speakers [9]. There are three types in Arabic language: a. Classical Arabic; b. Modern Standard Arabic (MSA); c. Dialectal Arabic [10]. Arabic ASR is not easy task according to the challenges of Arabic language such as; the language data sparseness, lexical diversity, diversity of spoken dialects language, non-diacritized text, and complex morphology [10].

Arabic language lacks of enough training and corpora for ASR tasks. So, data augmentation technique is a good solution for presenting additional training data by adding small perturbations on the initial training set [11, 12].

In this paper, we propose CNN-LSTM as state-of-art in end-to-end ASR for building acoustic, and attention-based models for speech decoding. Data augmentation is applied on the original corpus for enhancing the performance and accuracy. Data augmentation process produced additional training data by adding noise, pitch-shifting, and speed transformation to the original speech samples.

The rest of this paper is organized as follows: The works related to Arabic ASR are introduced in section 2. Section 3 contains the background for techniques that are used in this work. In Section 4, we illustrate the research methodology of the proposed approach. Section 5 explains the experiments and the conducted results. Finally, section 6 presents the conclusion and future perspectives.

2. Related Work

Al-Anzi et al. [13] presented work for evaluating Arabic ASR based on some of phonological rules. These rules include three types, namely Shadda, Tanween, and special letters that affect the enhancement of Arabic ASR. The Carnegie Mellon University PocketSphinx speech toolkit was utilizes for building acoustic model. They used "in-house" modern standard Arabic speech corpus is used for training for testing.

Alsharhan et al. [14] introduced the work for investigating the effect of characteristic of data on the quality of Arabic ASR systems. They presented an analysis the interaction between some methods of MSA data (feature selection, data selection, and gender-dependent acoustic models) and dialect Arabic data. There are three conditions were applied to enhance Word Error Rate (WER), these conditions are MFCCs with 25- dimension, deleting the lowest quality audio, and building acoustic model using a grapheme. Then registering variation in dialect and gender are used as features for decreasing WER. The acoustic model is built using deep neural network (DNN) in HTK toolkit. The GALE (phase 3) corpus used for training and testing the employed model. All experiments decreased WER between 3.24% and 5.35%.

Alsharhan et al. [15] investigated the effects of the complex morphology of Arabic on Arabic ASR. The different acoustic and language model were built based several transcription that are non-diacriticised based grapheme transcription, phoneme-based transcriptions, and dictionary. The SAMA and MADAMIRA toolkits were used to generate phoneme-based transcriptions. The HTK toolkit with artificial neural network (ANN) was used for building acoustic and language modeling. The GALE (phase 3) corpus was used for training and testing the acoustic model. The WER results were obtained 14.71%, 27.24%, and 21% for Broadcast Conversations (BC), Broadcast Reports (BR), and Combined, respectively.

Khatatneh et al. [16] developed Arabic ASR system for phoneme using neural network (NN) method. Pre-processing step is conducted by NN algorithm for improving the performance. They used sampling, catching a signal, setting energy, and quantization techniques for enhancing the results. In addition, Gaussian Low Pass filtering algorithm is used for handling the noisy signal. They reported that their system achieved better results.

Najafian et al. [17] presented a description for Arabic ASR based MGB-3 and MGB-2 data. Some techniques are used to enhance the accuracy and performance such as data preprocessing, data augmentation, language model adaptation, and accent determined re-training. Deep learning based acoustic modeling topologies was used for building the acoustic model. The 4-gram language model was built. The best WER obtained on MGB-3 using a 4-gram re-scoring strategy is 42.25% for a BLSTM system, compared to 65.44% for a DNN system.

Ahmed et al. [18] used different acoustic models for building Arabic ASR system. The decision tree is proposed for constructing the pronunciation variant generation. Acoustic model is constructed using another native acoustic model. The acoustic model achieved 10.7% WER.

Ahmed et al. [11] developed and described an Arabic ASR based MGB-5 in Arabic. They applied speech augmentation using speed and volume perturbation, data reverberation and music-noise-speech injection transformation. CNN with TDNN and TDNN-f were used for building the acoustic model. The x-vector and i-vector are combined and used as new features in this system. In addition, language model

interpolation, semi-supervised learning, genre adaptation, and lattice-based MBR are proposed and combined. The proposed system achieved an average WER of 59.4%.
Zerari et al. [19] proposed an end-to-end ASR for identifying isolated Arabic words. Recurrent neural network (RNN) are utilized to extract the features as fixed-length vectors. In addition, multilayer perceptron is suggested to use the feature vectors for classifying words. Network training and encoding are conducted using RNN-LSTM/GRU method. They used Spoken Arabic Digit dataset[1] for training and testing. F-measure was reported and achieved 98.77% as accuracy.

3. Methodology

3.1. Convolutional Neural Network

Convolutional neural network (CNN) is a type of artificial neural networks for processing data that has been represented in pattern form. CNN is a suitable method for building deep learning model that has been used for object recognition tasks. CNN was presented for decreasing the input data that are required for traditional artificial neural network [20]. Convolution, pooling, and fully connected layers are considered the components of traditional CNN [21]. Convolution layer is an operation for feature extraction, whereas pooling layer is used for decreasing the number values in feature maps. The optimal feature maps are transferred into the classifier using fully connected layer [21, 22]. In general, CNN is scalable and needs less time for the training process [20]. CNN is considered as an optimal technique for enhancing the results when used in computer vision or object detection tasks [20, 23, 24]. CNN has properties make it optimal method for other tasks such as speech recognition. These properties are locality and weights sharing that are used to decrease the noises and shift the frequencies in signals. In addition, the first property reduces the number of weights in the learning network; whereas the translational variance is reduced by weights sharing property [7]. Figure 1 illustrates the layers of CNN.

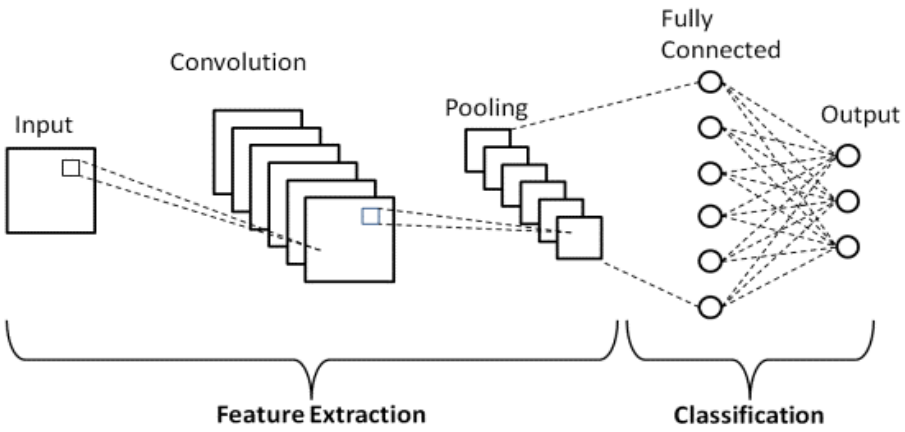


Figure 1: CNN Layers [25]

[1] https://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit

3.2. Long Short Term Memory Networks

Recurrent Neural Network (RNN) uses shared hidden state for processing sequential inputs and represents them in the activation function. RNN may be able to fetch the previous information to the current state for prediction the new information. However, the long term dependency and the vanishing gradient represent problems for RNN. So, there are different techniques can handle these problems such as Long Short Term Memory Networks (LSTMs) [26]. LSTMs are one kind of RNN that uses four interacting layers with a unique communication link [10, 27] as shown in Figure 2.

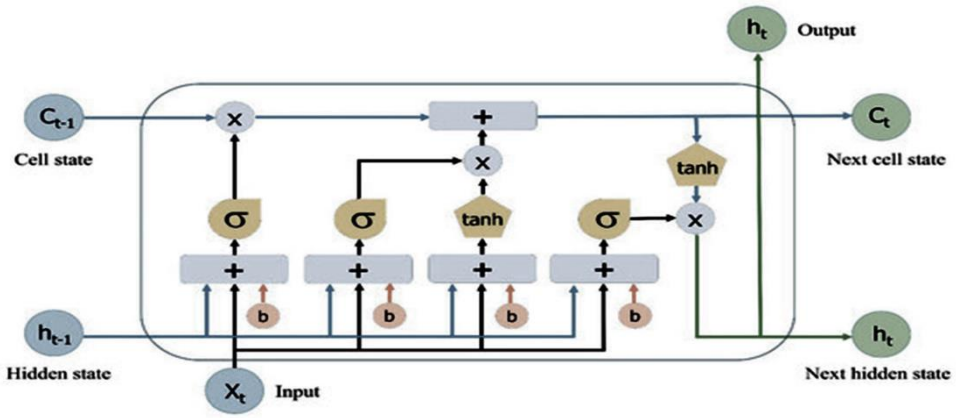


Figure *2*: The Structure of LSTM [27]

An LSTM consists of a set of cells used as memory units. It send cell and hidden states into the next cell. The first step is used to receive the input data. While the second step in LSTM is to determine unrelated information to be ignored from the current cell. The next step is updating the cell state by determining the important information resulted from the new inputs. In the final step, the output of the cell state is filtered to decide the output values [6, 12].

3.3. Attention Model

Attention is a technique which is used for calculating soft alignment of information between input data and corresponding labels directly [28]. End-to-end Seq2Seq system is built based on encoder-decoder using traditional RNN method. The encoder in this system represents input data by a fixed length vector, whereas the decoder is used to encode the output label depends on output of encoder. This model has a problem due to machine translation [29, 10]. All inputs are represented by fixed length vectors. In addition, each input (word or sentence) will be assigned by the same weight that leads to degrade the model performance [29]. Thus, the purpose of Attention mechanism is to solve the encoder-decoder problems. Using attention method, the encoder encodes each input data into a sequence of vectors, whereas the decoder assigns different weights to each input [10, 28]. Speech recognition task is a sequence-to-sequence process, so attention method is a suitable method for ASR. Attention-based end-to-end model can also be divided into three parts: encoder, aligner, and decoder [28]. Figure 3 shows the structure of attention model.

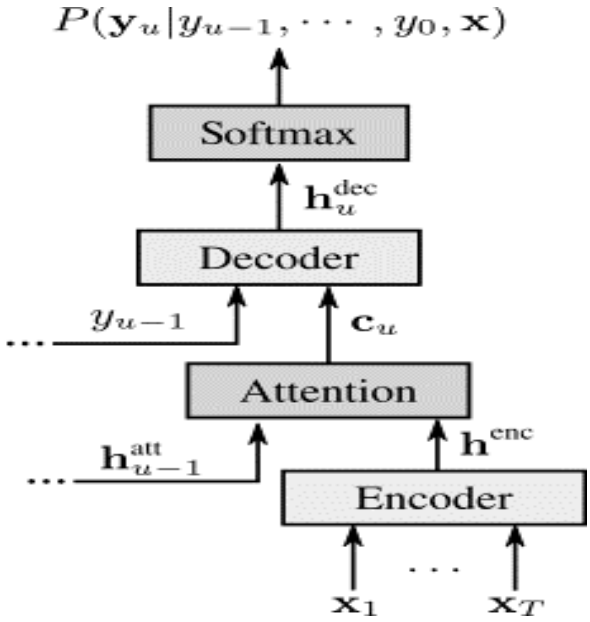


Figure 3: Attention Model Structure [28]

3.4. Data Augmentation

The main purpose of deep learning techniques is building an efficient model for improving the results and performance, but this model requires large size of training data [22]. There is another technique used for solving the limitation of training data namely data augmentation [22]. Data augmentation is a method for increasing training data which has been shown to be effective for deep learning training to build robust predictions [30, 31, 32]. Data augmentation is a good process for speech recognition task because of the lack of enough training data and corpora [11]. In addition, data augmentation is used to improve the quality of the amount and variety of training data. This process will enhance the robustness of the models and avoid overfitting. There are some used effective methods to avoid overfitting such as [33, 34]. In speech recognition, data augmentation includes three types namely time stretching, pitch shifting, and Music-Noise-Speech injection [11, 12].

3.4.1 Pitch-Shifting-Speech Method

Speed method greatly enhances the performance of speech recognition tasks. It is utilized to resample the original speech signal. For man speech in the speech corpus, when the original speed is increased by rate, the new speech will be nearest to woman or child voice. We apply speed-speech method to generate new data using random length as a uniform distribution between 0.8 to 2 millisecond. This process creates one additional copy of the original corpus.

3.4.2 Pitch-Shifting-Speech Method

Pitch-Shifting-Speech method is used to produce new data from original data for representing other sample in population. It utilized to modify the original speech signal for making it nearest to woman and child voice. The Pitch-Shifting-Speech method is configured by adding a random length as a

uniform distribution between 1 and 10 Mel scale frequency. This process creates one additional copy of the original corpus.

3.4.3 Noise-Speech injection Method

Noise-Speech is proposed to produce new data by adding noise signal to original data. A random length of noise is selected to have a uniform distribution interval [0.01, 0.1] for Noise-Speech injection method. This process creates one additional copy of the original corpus.

3.5. Data

The Standard Arabic Single Speaker Corpus (SASSC) [35] is used in this work that contains 51K words in 7 hours of recording. The 4372 utterances (Audio + diacritized text) are recorded at 96kHz sampling rate. It has different classes such as news, date-time, names, number, customer-service, story, Miscellaneous, Financial, and traditional.

3.6. The Proposed System Architecture

End-to-end ASR is proposed to build an Arabic ASR based on non-diacritized form SASSC corpus, the architecture of the proposed system is shown in Figure 4. All details of this system will be presented in the next subsections.

3.6.1. **Preprocessing**

In this stage, we used three preprocessing methods to adapt and prepare the data: preparing the nondiacritized data, data augmentation, and the required files for training and testing processes.

A. Preparing the cleaned Data

The nondiacritized, external and transcriptions of data are prepared for data augmentation and training processes. The Python code is written for producing all data in this sub-step.

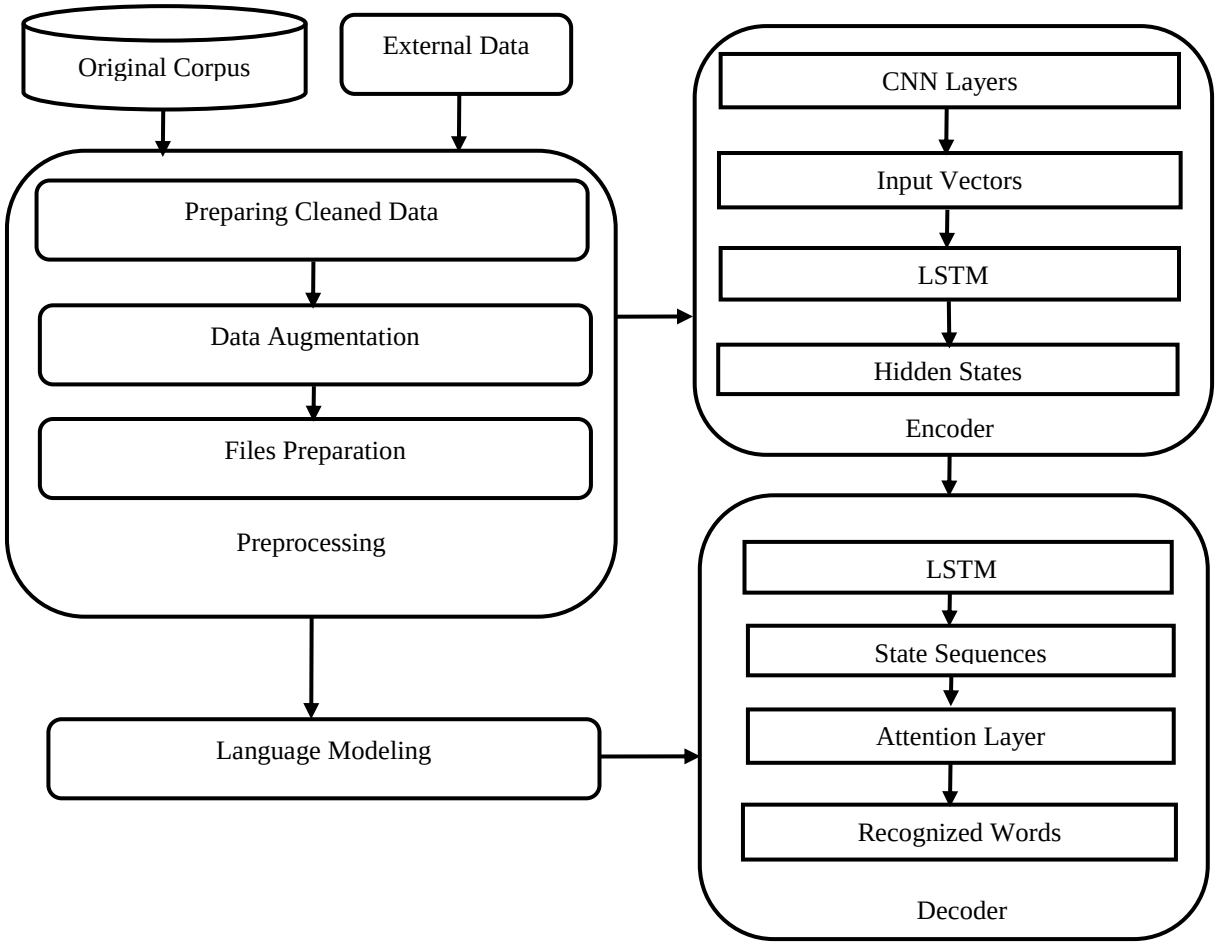


Figure. 4: The Proposed System Architecture

B. Data Augmentation

Several data augmentation methods are used to obtain additional training data based on SASSC dataset. We applied Speed-Speech, Noise-Speech injection, and Pitch-Shifting-Speech methods to produce three times from original data. A total of 16.8 hours were obtained from the original SASSC training and development data. Figure 5 shows the algorithm of the data augmentation processes. The total of the new training and development data is 22.4 hours after applying data augmentation. We write a Python code for producing the new training data. The performances is analyzed before and after applying the data augmentation methods on the testing dataset.

Algorithm 1: Applying data augmentation on original SASSC corpus

Input: A text **file** containing list of all audio files and their names and paths.
Output: All audio files after applying data augmentation methods.
begin

```
Speed-Target-Path ← specify a path for output audio file  // after applying Data Augmentation
Noise-Target-Path ← specify a path for output audio file  // after applying Data Augmentation
Pitch-Target-Path ← specify a path for output audio file  // after applying Data Augmentation

for each line ∈ file do
    data = read_audio_file(line)
    file_name ← get_file_name(line)
    // Speed augmentation method
    data_ stretch ← librosa.effects.time_stretch(data, 0.8)
    write_audio_file(Speed-Target-Path + '\' + file_name + '-speed', data_ stretch)
    // Noise augmentation method
    noise ← random(length(data))
    data_noise ← data + 0.09 * noise
    data_noise ← data_noise.astype(data)
    write_audio_file(Noise-Target-Path + '\' + file_name + '-noise', data_noise)
    // Pitch augmentation method
    data_pitch ← pitch_shift(data, 16000, 5)
    write_audio_file(Pitch-Target-Path + '\' + file_name + '-pitch', data_ pitch)
End For
```

End

Figure. 5: *Algorithm for Data Augmentation*

C. Files Preparation

Kaldi recipe is used to prepare the data of directories, lexicons, and language model data. According to Kaldi-defined, the training and testing data are stored in text files while the utterances are stored in SCP format which contains the utterances ID, Feature file. Each utterance has utterance reference that reads the binary information from acoustic features. The corpus and collected data are merged in one file to present training data for language modeling. In addition, Features are extracted as MFCC vectors in this step.

3.6.2. **Language Modeling**

Language model (LM) is trained to build external LM. This LM is called Look-ahead model and used to present word based and character-based LMs using RNN-LM and LSTM-LM methods. Whereas, RNN-LM is used to calculate the probability of the next character depend on all previous words and the prefix of the word. While the LSTM-LM is used for predicting the word probability. The prefix tree is used for building the character-based LM depending on word-based LM.

3.6.3. **Acoustic Model**

The deep CNN-LSTM network is used to build the acoustic model using Espresso toolkit [36]. CNNs are used to get better properties of spectral and temporal invariance and reduce translational variance using convolutional filters. Few parameters are utilized in the training network. The LSTM layer is a set of RNNs with memory blocks which is used for real-world sequence processing problems. LSTM with subsampling is utilized to forward each sentence into different networks and backward it into these networks. CNN with 2-dimensional convolution is built to represent the time frame and feature. The downsampling is used in the first and third layers. The final layer includes 128 dimensions and 21

downsampled frequency features. We stacked the convolution layers for producing the output channels. LSTM received the output from the output channels to process the time dependencies learning. In addition, this model used the scheduled sampling to determine the minimum probability. For enhancing the accuracy the temporal smoothing schema with p = 0.05 was used to make the model able to ignore a sub-unit in the transcript depending on beam search errors.

3.6.4. **Decoding (Recognition)**

LSTM with attention-based is used as end-to-end approach for decoding for speech decoding. A 3-layer decoder is presented above the hidden states in LSTM. Output vectors are obtained and sent into LSTM using attention method. 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 quality of Arabic ASR system.

4. Experimental Results

4.1. Experimental Parameters

The results discussed in this section are expressed in terms of two experiments. Firstly, we train acoustic model on the original data (training and development). Secondly, we trained acoustic model on the augmented data. For original data, we used 4.6 hours as training data and one hour as development data. Based on original and augmented data, we used 18.4 hours for training and 4 hours for development. The same testing data is used for both experiments, which includes 1.4 hours. Experiments are conducted using a machine with GeForce GTX 1060 6GB as GPU, Intel i7-8750H as CPU, 16 GB as RAM, and CUDA version 10.0.

LM is trained using the training and collected data with different epochs 5k as batch size. The best evaluation of LM model is obtained in the fifth epoch for original data and tenth epoch for all data. We use attention dim= 320, d-model =512 model dimension and d-inner = 1024 inner-layer dimension to train CNN-LSTM network. After that, the pooling layer with 512 hidden node and two fully connected layers with 48 hidden nodes are added to the training network. In addition, CNN-LSTM is trained with different epochs for enhancing accuracy. The best evaluation of the trained model is obtained for original data and all data after 400 and 250 epochs, respectively. In recognition process, the fusion weight of 0.50 and end-of-sentence threshold of 1.5 are used for integrating with LM model in order to improve the accuracy and performance. Finally, the decoding process use beam size with 50.

4.2. Results Discussion

This work presented the evaluation results using the proposed models based on original data and all data after data augmentation. Word error rate (WER) and character error rate (CER) are employed for both employed models based on the evaluation metrics in equations (1) and (2).

$$WER = \frac{I_1 + D_1 + S_1}{N_1} \quad (1)$$

where I_1, D_1, S_1, N_1 represent insertion, deletion, substitution, and number of words, respectively.

$$CER = \frac{I_2 + D_2 + S_2}{N_2} \quad (2)$$

where I_2, D_2, S_2, N_2 represent insertion , deletion, substitution, and number of characters, respectively.

Table 1 shows the results of presented model based on original data before applying data augmentation using different epochs. While the results of the employed model based on the original and augmented data are resulted in Table 2.

Table 1 Results of the proposed model before applying data augmentation

#epochs	CER(%)	WER(%)
35	5.81	15.78
50	5.74	15.78
100	5.75	15.62
200	5.79	15.42
250	5.71	15.34
300	5.72	15.35
350	5.54	15.14
400	5.39	14.96
450	5.39	14.96
500	5.39	14.96

Table 2 Results of the proposed model after applying data augmentation

#epochs	CER(%)	WER(%)
35	5.13	11.63
50	4.98	11.31
100	4.94	11.07
150	4.80	10.87
200	4.67	10.63
250	4.43	10.41
300	4.43	10.41
350	4.43	10.41
400	4.43	10.41

From above results, we note that applying the data augmentation methods led to reduce CER and WER compared with the data before applying the data augmentation methods. The overall reduction in WER is 4.55% and CER is 0.96%. In addition, the best result of the trained model using all data is obtained in the epoch #250 while the best result of the trained model using original data is obtained in epoch #400. We conclude the CNN-LSTM with attention methods resulted the best result with fewer epoch with the large datasets. The best results of two trained models are summarized using a chart in Figure 6. This figure shows the effect of applying the data augmentation on the system's performance.

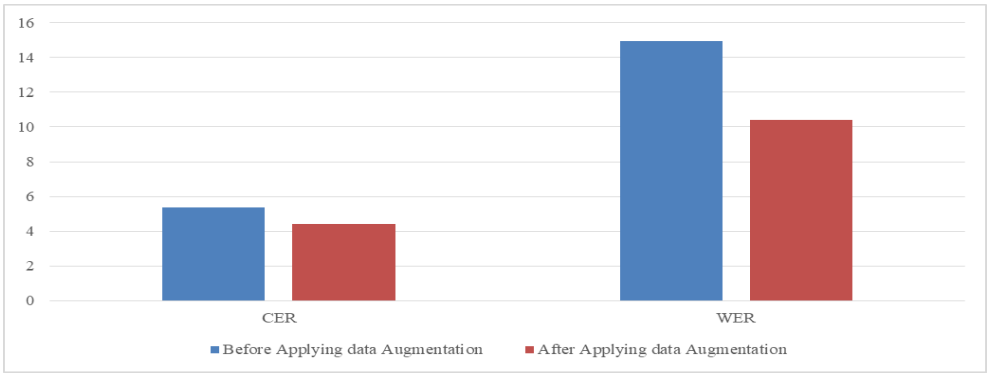


Figure. 6: The best results of two trained models

From the results in Table 1 and Table 2, we conclude the accuracy of proposed model after data augmentation better than accuracy before data augmentation in each epoch. In each epoch, the accuracy is reduced after applying data augmentation compared with before applying data augmentation. Figure 7 shows curves to state the gap between the accuracy before applying data augmentation and after applying data augmentation in each epoch.

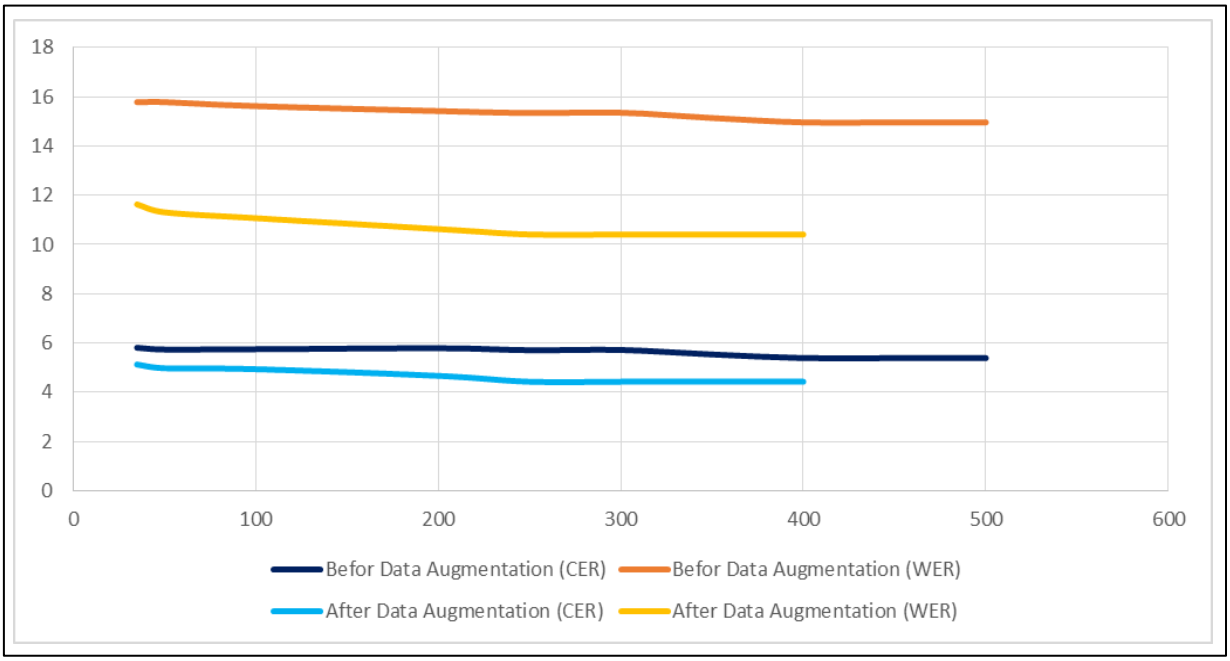


Figure 7: The accuracy of two models in each epoch

Our best results are compared with previous works are done by Hagen et al. [37], Ahmed et al. [38], Tuka et al. [39], and Eiman et al. [15] using WER. Our results achieved better progress than their best results as shown Table 3.

Table 3 Comparison with other Arabic ASR systems.

System	WER
Hagen et al. [37]	17.00%
Ahmed et al. [38]	15.81%
Tuka et al. [39]	18.30%
Eiman et al. [15]	21.00%
The proposed System	**10.41%**

5. Conclusion and Future perspectives

In this paper, the effect of data augmentation is investigated for Arabic ASR based on end-to-end deep learning. SASSC speech corpus is employed in this work. We applied data augmentation on original data using adding noise, pitch-shifting, and speed transformation. Acoustic model is built using CNN-LSTM with attention-based model in end-to-end ASR. The results show that the proposed approach after applying data augmentation achieved an accuracy better than original corpus by 4.55% in terms of WER criterion. In future work, we will apply various factors for noise, pitch-shifting, and speed transformation and other data augmentation techniques to generate more speech samples, and thus better recognition results can be achieved.

References

1. He, Xiaodong, and Li Deng. "Discriminative learning for speech recognition: theory and practice." Synthesis Lectures on Speech and Audio Processing 4.1: 1-112 (2008).

2. Ali, Ahmed, Mohamed Abdel Maksoud. "Multi-dialect Arabic broadcast speech recognition." (2018).
3. AbuZeina, Dia, Wasfi Al-Khatib, Moustafa Elshafei, and Husni Al-Muhtaseb. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3: 227-236 (2011).
4. Satori, Hassan, Hussein Hiyassat, Mostafa Harti, and Noureddine Chenfour. "Investigation Arabic Speech Recognition Using CMU Sphinx System." International Arab Journal of Information Technology (IAJIT) 6.2 (2009).
5. Belinkov, Yonatan, Ahmed Ali, and James Glass. "Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition." arXiv preprint arXiv:1907.04224 (2019).
6. Hori, Takaaki, Jaejin Cho, and Shinji Watanabe. "End-to-end speech recognition with word-based RNN language models." 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018.
7. Passricha, Vishal, and Rajesh Kumar Aggarwal. "A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition." Journal of Intelligent Systems 29.1: 1261-1274 (2020).
8. Abdel-Hamid, Ossama, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. "Convolutional neural networks for speech recognition." IEEE/ACM Transactions on audio, speech, and language processing 22.10: 1533-1545 (2014).
9. Abdou, Sherif Mahdy, and Abdullah M. Moussa. "Arabic speech recognition: Challenges and state of the art." Computational Linguistics, Speech and Image Processing for Arabic Language. 1-27. (2019)
10. Abdelhamid, Abdelaziz A., Hamzah A. Alsayadi, Islam Hegazy, and Zaki T. Fayed. "End-to-End Arabic Speech Recognition: A Review." in The 19th Conference of Language Engineering (ESOLEC'19) At Alexandria, Egypt, 2020.
11. Ahmed, Hany, Hazem Mamdouh, Salah Ashraf, Ali Ramadan, and Mohsen Rashwan. "RDI-CU SYSTEM FOR THE 2019 ARABIC MULTI-GENRE BROADCAST CHALLENGE." (2019).
12. Vachhani, Bhavik, Chitralekha Bhat, and Sunil Kumar Kopparapu. "Data Augmentation Using Healthy Speech for Dysarthric Speech Recognition." Interspeech. 2018.
13. Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology 20.3: 715-723 (2017).
14. Alsharhan, Eiman, and Allan Ramsay. "Investigating the effects of gender, dialect, and training size on the performance of Arabic speech recognition." Language Resources and Evaluation 54.4: 975-998 (2020).
15. Alsharhan, Eiman, Allan Ramsay, and Hanady Ahmed. "Evaluating the effect of using different transcription schemes in building a speech recognition system for Arabic." International Journal of Speech Technology: 1-14 (2020).
16. Khatatneh, Khalaf. "A novel Arabic Speech Recognition method using neural networks and Gaussian Filtering." International Journal of Electrical, Electronics & Computer Systems 19.1 (2014).
17. Najafian, Maryam, Wei-Ning Hsu, Ahmed Ali, James Glass. "Automatic speech recognition of Arabic multi-genre broadcast media." 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017.
18. Ahmed, Basem HA, and Ayman S. Ghabayen. "Arabic automatic speech recognition enhancement." 2017 Palestinian International Conference on Information and Communication Technology (PICICT). IEEE, 2017.

19. Zerari, Naima, Samir Abdelhamid, Hassen Bouzgou, and Christian Raymond. "Bi-directional recurrent end-to-end neural network classifier for spoken Arab digit recognition." 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP). IEEE, 2018.
20. Nazih, Waleed, Yasser Hifny, Wail S. Elkilani, and Tamer Abdelkader. "Fast Detection of Distributed Denial of Service Attacks in VoIP Networks Using Convolutional Neural Networks." International Journal of Intelligent Computing and Information Sciences 20.2: 125-138 (2020).
21. Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." Insights into imaging 9.4: 611-629 (2018).
22. Abeßer, Jakob. "A review of deep learning based methods for acoustic scene classification." Applied Sciences 10.6 (2020).
23. Nassif, Ali Bou, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. "Speech recognition using deep neural networks: A systematic review." IEEE access 7: 19143-19165 (2019).
24. Sadek, Esraa T., Noha A. Seada, and Said Ghoniemy. "Computer Vision Techniques for Autism Symptoms Detection and Recognition: A Survey." International Journal of Intelligent Computing and Information Sciences 20.2: 89-111 (2020).
25. Phung, Van Hiep, and Eun Joo Rhee. "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets." Applied Sciences 9.21(2019).
26. Rizk, Basem. Evaluation of state of art open-source ASR engines with local inferencing. Diss. Bachelor's Thesis, Institute of Information Systems, Hof University, 2019.
27. Le, Xuan-Hien, Hung Viet Ho,Giha Lee, and Sungho Jung. "Application of long short-term memory (LSTM) neural network for flood forecasting." Water 11.7: 1387 (2019).
28. Wang, Dong, Xiaodong Wang, and Shaohe Lv. "An overview of end-to-end automatic speech recognition." Symmetry 11.8: 1018 (2019).
29. Wang, Song, and Guanyu Li. "Overview of end-to-end speech recognition." Journal of Physics: Conference Series. Vol. 1187. No. 5. IOP Publishing, 2019.
30. Rebai, Ilyes, Yessine BenAyed, Walid Mahdi, and Jean-Pierre Lorré. "Improving speech recognition using data augmentation and acoustic model fusion." Procedia Computer Science 112: 316-322 (2017).
31. Cai, Weicheng, Haiwei Wu1, Danwei Cai, and Ming Li. "The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion." arXiv preprint arXiv:1907.02663 (2019).
32. Ko, Tom, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. "Audio augmentation for speech recognition." Sixteenth Annual Conference of the International Speech Communication Association. 2015.
33. Park, Daniel S., William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," Proc. of Interspeech 2019.
34. Nguyen, Thai-Son, Sebastian Stuker, Jan Niehues, and Alex Waibel. "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation." *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020.

35. Almosallam, Ibrahim, Atheer Alkhalifa, Mansour Alghamdi, Mohamed Alkanhal, and Ashraf Alkhairy. "SASSC: A standard Arabic single speaker corpus." Eighth ISCA Workshop on Speech Synthesis. 2013.
36. Wang, Yiming, Tongfei Chen, Hainan Xu, Shuoyang Ding, Hang Lv, Yiwen Shao, Nanyun Peng, Lei Xie, Shinji Watanabe, and Sanjeev Khudanpur. "Espresso: A fast end-to-end neural speech recognition toolkit." 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019.
37. Soltau, Hagen, George Saon, Daniel Povey, Lidia Mangu, Brian Kingsbury, Jeff Kuo, Mohamed Omar and Geoffrey Zweig. "The IBM 2006 Gale arabic ASR system." 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. Vol. 4. IEEE, 2007.
38. Ali, Ahmed, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. "A complete KALDI recipe for building Arabic speech recognition systems." 2014 IEEE spoken language technology workshop (SLT). IEEE, 2014.
39. AlHanai, Tuka, Wei-Ning Hsu, and James Glass. "Development of the MIT ASR system for the 2016 Arabic multi-genre broadcast challenge." 2016 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2016.