Investigating Speech Attribute Features for Anti-Phone based Pronunciation Verification Approach

Ayat Hafzalla Ahmed
College of Computer Science and Information Technology
Sudan University of Science and Technology

Hager Morsy
Information Technology, Faculty of Computers and Information, Cairo University, Giza, Egypt

Sherif Mahdi Abdo
Faculty of Computers at Cairo University Egypt
The Research and Development International (RDI)

ABSTRACT

With increased computing power, there has been a renewed interest in computer-assisted pronunciation learning (CAPL) applications in recent years;

Automatic accurate pronunciation verification method plays an important role in automating the learning process and increasing its quality.

Pronunciation errors can be divided into phonemic and prosodic error types. In this paper we propose a phoneme-level pronunciation verification method for Quranic Arabic based on anti-phone model. For each phoneme a binary support vector machine (SVM) classifier is trained to distinguish each phoneme from other phonemes. The (SVM) classifier is trained using speech attribute features derived from a bank of speech attribute detectors, namely manners and places of articulation. The feed forward deep neural network (DNN) architecture is utilized for the speech attribute detectors. The system is evaluated against speech corpora collected from fluent Quran reciters and achieved phoneme-level false-acceptance and false-rejection rates ranging from 2% to 25%.

General Terms

Artificial intelligent, Speech Recognition

Keywords

Speech attributes, pronunciation verification, anti-phone model, Quranic Arabic

1. INTRODUCTION

Computer Assisted Language Learning (CALL) applications, and, more specifically, Computer Assisted Pronunciation Training (CAPT) , computer aided speech and language therapy (CASLT) ,a computer aided pronunciation learning (CAPL) and language proficiency test applications that make use of automatic speech recognition (ASR) have received considerable attention in recent years. Automatic detection of pronunciation errors is a fundamental feature in that applications because using such a method will fully automate the learning application; make it more interactive and available all times. Automated the learning process using computer aided applications can dramatically speed up the learning curve compared to the traditional learning methods as shown in [1, 2]. However, the perfection of the automatic pronunciation error detection method is of crucial importance for the reliability of the learning tool. The pronunciation verification can be done on speaker-level, sentence-level, word-level or phoneme-level. Pronunciation error detection at the phoneme level is a much harder than measuring pronunciation fluency across multiple sentences. Kim et al. [3] ,but the most informative one.

Various approaches to phoneme pronunciation error detection can be found in the literature. The early work was relying on the confidence score approach where a certain score is estimated to measure the pronunciation quality of each phoneme. In [4] the authors showed that the posterior probability score outperform other scores based on the log-likelihood and the segment duration

The best known example of phoneme level error detection is relied on the Goodness Of Pronunciation (GOP) algorithm developed by Witt [5, 6], where a certain score is estimated to measure the pronunciation quality of each phoneme by estimating the posterior probability of each phoneme as a likelihood ratio between the forced alignment likelihood and the maximum likelihood obtained from free phone recognition and achieved highest correlation with the human scoring. The GOP is then becoming the most popular and commonly used confidence-score based pronunciation verification algorithm and successfully applied in different problems [7, 8].

A deep neural network (DNN) version of the GOP method proposed in [9] and achieved around 34% improvement over the convention GOP method.

The phoneme-level error detection can also be considered as a binary classification problem by classifying each phoneme as “correctly pronounced” or “mispronounced”. This approach was adopted in several research using different off-the-self binary classifiers such as support vector machines (SVM) [10,11], classification and regression tree (CART) [12] and artificial neural network (ANN) [13]. In Truong et al. [14] LDA classifiers manage to discriminate between voiceless fricatives and plosives in non-native Dutch and achieve 87-95% classification accuracy. A comparison between linear discriminative analysis (LDA) classifier and GOP had been performed on one Dutch phoneme [15] and showed that LDA classifier outperforms the GOP method.

Other phoneme error detection method using Support Vector Machine (SVM) with structural features compared to two baseline methods of Goodness of Pronunciation (GOP) and Likelihood Ratio (LR) under the task of Experiments in [16]. That verification method show that the (SVM) with structural features performs much better than both of the two baseline methods. For example, the false rejection rate is reduced by as much as 82%.

Another successful approach for phoneme-level pronunciation error detection is the extended recognition networks (ERNs) where a search network is constructed based on the common pronunciation error made by the learners. This approach is proved to be effective in custom problems whenever the expected pronunciation errors are known. In [17] the author construct an ERN containing the most common mispronunciations of Cantonese learners of English and the system was able to correctly recognize 54.8% of the mispronunciations. Similar approach was adopted in [18] for the diagnosis of speech disorder in children. The ERN designed to cover different pronunciation errors made by children with apraxia of speech disorder. The advantage of this method is that it does not only detect the location of the mispronounced error, but also it provides the type of the error, which is very important feature in the design of feedback messages. However, the performance of this method is affected significantly if the learner produces an unexpected error which is not covered by the constructed network.

Recently, speech attribute features which are derived from speech attribute detectors, namely the manners and places of articulation have achieved promising results in tackling the pronunciation verification problem [19]. Yet, the use of these features in mispronunciation detection is still limited.

In this paper, we are going to investigate the efficacy of the speech attribute features in pronunciation verification and then apply it on learning Quranic Arabic recitation. The attribute features are derived from a bank of DNN-based speech attribute detectors trained to recognize the existence and absence of each attribute. These features are then feeding a binary SVM anti-phone model for each phoneme to discriminate between frames belonging to this specific phoneme and frames from other phonemes. This paper also considers the first intensive study of speech attribute detection of Arabic language.

The rest of the paper is organized as follows. A background about the Quranic Arabic and the speech attribute detection is presented in section II. Detailed description of the speech corpora is demonstrated in

section III. In section IV, we explain the details of the proposed system. The results are presented in section V. Finally, the conclusion is drawn in section VI

2. BACKGROUND

3. Quranic Arabic

Quran, The Holly Book of Muslims which contains standard Arabic text. Millions of Muslims worldwide are seeking to learn the correct pronunciation of Quran of which most of them are not native Arabic speakers. In fact, even Muslims who are from countries where Arabic is the official language are not speaking standard Arabic but a dialectic version of it. Therefore, the native Muslim Arabic speaker have to learn how to correctly pronounce Quran. Moreover, Quranic Arabic has more sophisticated pronunciation rules than the standard Arabic and even additional sounds.

Reciting Quran in front of a professional teacher who listens to each individual reading and gives corrective feedback is yet the most popular way to learn the correct pronunciation of Quran. However, due to the shortage in Quran teachers, specifically in non-Arab countries, and their limited availability, this one-to-one traditional learning method is inconvenient and not available for the majority of Muslims.

The CAPL, with speech recognition capabilities, provide an effective and interactive alternative to the traditional teacher-led learning way. The key advantage of the CAPL is its availability where the learner can use the application at his convenient, in addition to the corrective feedback which controlled by the output of the speech recognition.

In spite of the success of this approach in learning domains such as second language learning, it achieved limited success in the Quranic Arabic pronunciation learning for two main reasons. Firstly, unlike the second language acquisition the fluency in Quran recitation is more important than the intelligibility where the correct pronunciation of each phoneme is crucial. Secondly, the work done in CAPL for Arabic as a second language in general, and for Quranic Arabic in specific, is still very limited compared to other languages such as English, Chinese, etc.

A speaker independent HMM-based speech recognition system for Quranic Arabic was presented in [20] with a word-level accuracy ranging from 68% to 85%. In [21] the author investigated the use of word-level ASR method for automatic Quran memorization system applied on a small dataset consists of 20 words produced by only one speaker.

The most successful attempt for developing a CAPL application for Quran recitation was HAFSS© [22], a system designed to automatically detect the pronunciation errors of reciting a short examples of Quran covering all recitation rules. The system based on a predefined phoneme-level ERN for each example containing all expected pronunciation errors which used to align the user recitation and produce a set of corrective feedback messages to the user. The system correctly identified 63% of the error made and the false acceptance rate was 15%. The system is further improved by using DNN-based acoustic model [23].

3.1 Speech attributes

The speech attributes are mainly the places and manners of articulation such as labial, dental, stop, fricative, etc.

In [24] Lee et al. proposed the automatic speech attribute transcription system (ASAT) where a bank of speech attribute detectors were trained to measure the existence or absence of each attribute and the output features is then merged and used for performing ASR. This bottom-up approach is known as knowledge-based speech recognition [25].

The powerful of the speech attribute features is that they are shared among languages and therefore speech corpora from multiple languages can be used in designing a universal speech attribute detectors [26].

In addition to the ASR system, the speech attributes features were used for other speech processing problems. Zhang et al. [27] investigated the effectiveness of such features in achieving speaker verification and the results showed that the proposed system outperform all other speaker verification methods. Furthermore, the attribute features were successfully utilized for foreign accented characterization [26] and spoken language recognition [28].

The speech attribute features are very helpful in the pronunciation verification problem. In fact, the phoneme is considered mispronounced when one or more of its attributes are changed. Several attempts for using the speech attribute features in tackling pronunciation verification problem in the literature. The speech attribute features were utilized in [19] to improve the mispronunciation detection and provide diagnostic feedback for Mandarin learners. In [29] the author introduced the so called articulatory goodness of pronunciation (aGOP) score where the articulation features is used for estimating the phoneme posterior probability.

Most recently, Shahin et al. [30] proposed an anomaly detection based system for phoneme-level pronunciation verification. The authors built a one-class SVM model for each phoneme trained using speech attribute features derived from a bank of DNN-based speech attribute detectors. The system tested against different English speech corpora collected from native, foreign-accented and disordered speakers. The system compared to the GOP algorithm and reduced the false-acceptance and false-rejection rates by 26% and 39% respectively.

However, the Arabic speech attribute features received very little attention in the literature. Hammady et al. proposed a hidden Markov model (HMM) for the detection of Arabic speech attributes [31]. While in [32] Ziedan et al. use the speech attribute features to discriminate among different Arabic dialect and accent.

In this work we study the automatic detection of Quranic Arabic speech attributes which contains all standard Arabic manners and places of articulation in addition to the special speech attributes for Quran.

4. SPEECH CORPORA

The advantage of the Quran text is that it is closed vocabulary (around 14716 unique words) with massive amount of speech data available from hundreds of different reciters. The Quran text consists of 114 chapters vary in their size from 12316 words to 25 words and each chapter consists of multiple verses. The duration of recording full Quran text from one speaker of average recitation speed is around 30 hours. Most of the available speech recordings are available on chapter-level, where each complete chapter saved as one continuous speech file. While few recordings are manually segmented into verse-level, where each verse saved as one continuous speech file. Although the availability of speech data, the quality of the recordings widely differ based on factors such as, the environmental noise, the type of recording devise, the existence of reverberation, etc.

In this work we use two speech corpora, one of which is segmented in verse-level (VER) which consists of 30 speakers reciting the last 56 chapters of Quran with total duration of around 90 hours. This corpus segmented from chapter-level to verse-level manually by every Ayah project [33] .The data released in mp3 format with different bit rate. This dataset divided into 3 subsets, namely training, validation and testing which contains 22, 4 and 4 speakers respectively.

The second corpora is available in chapter-level speech recordings (CHAP). Here only the last 36 chapters of the whole Quran are selected from 30 speakers. The corpus consists of 1080 files with duration vary from 1 to 6 minutes and the average duration per speaker of all files is around one hour. The two corpora are from fluent reciters with no pronunciation errors. Table 1 summarizes the characteristics of the speech corpora.

Table 1 Quran speech corpora Corpus

corpus	Fluency	N# Speakers	Duration
VER-train	Fluent	22	~66 hours
VER-valid	Fluent	4	~12 hours

5. METHOD

5.1 System Description

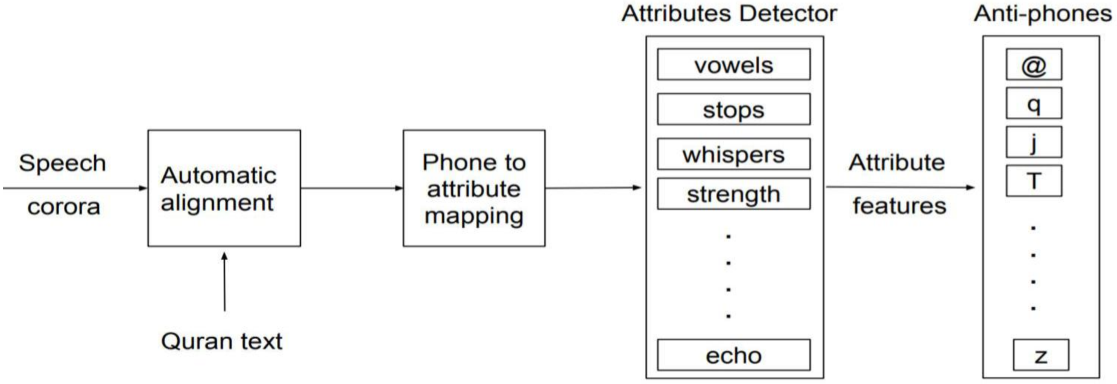

Figure 1 system flow diagram

Figure 1 shows the system flow diagram. First, the VER and CHAP Quran speech corpora along with the Quran text pass through a segmentation and alignment module. This module consists of an intensity-based voice activity detection (VAD) method and ASR method based on HMM acoustic models and n-gram Language Model (LM). The VAD used for segmenting the long speech files into short segments according to the silence position. The phoneme alignment is then performed using the ASR method.

Each phoneme is then mapped to its corresponding attribute according to a predefined Quran mapping rules. For each attribute we trained a binary DNN-classifier to classify each speech frame as positive, when the attribute is exist, or negative, when the attribute is missing. The samples from all phonemes belongs to a specific attribute is used as a positive samples while samples from all other phonemes are forming the negative ones.

The speech frame passed through the bank of pre-trained speech attribute detectors to extract the speech attribute feature vector. From each binary attribute detector, only the positive neuron contributed in the attribute feature vector.Using the attribute features we built a bank of anti-phone models, where a binary classifier is trained to discriminate each phoneme from all other phonemes

5.2 Pronunciation dictionary

First step is to create a Quran pronunciation dictionary to map each word to its corresponding phoneme sequence. The Quran is a special type of Arabic language which contains all standard Arabic phonemes in addition to extended set of phoneme describing its special pronunciation rules. For instance, the standard Arabic contains two nasal consonants („م‟ /m/ and „ن‟ /n/), however Quran has additional nasal sound called (:غن Ghunnah) which produced in several cases such as in geminated /m/ or /n/. Moreover, some phonemes are converted to another phonemes in some context such as /b/ which is pronounced as /m/ when comes after /n/ sound with No vowel. In other pronunciation rules, some phonemes dropped completely such as /n/ sound with no vowel when followed by /r/ sound, only the /r/ is pronounced.

Table 2 Quranic Arabic phoneme set Phoneme Description Phoneme Description Phoneme Description

Phoneme	Description	Phoneme	Description	Phoneme	Description
@	ء	S	ص	w	و
b	ب	D	ض	y	ي
t	ت	T	ط	m1	غنة م
t_h	ث	Z	ظ	n1	غنة ن
j	ج	~@	ع	m3	اقلاب
~h	ح	g_h	غ	a	فتحة
x	خ	f	ف	u	ضمة
d	د	q	ق	i	كسرة
~z	ذ	k	ك	a:	مد فتحة
r	ر	l	ل	u:	مد ضمة
z	ز	m	م	i:	مد كسرة
s	س	n	ن		
s_h	ش	h	ه		

The word pronunciation dictionary is then created by scanning the Quran text and applying the Quran pronunciation rules on each word to produce the correct phoneme sequence. As some of the pronunciation rules are cross-words, the pronunciation of the word may change based on the succeeded word. Therefore, each word in the pronunciation dictionary may contains more than one possible phoneme sequence.

In [34] the author proposed a phonological study of the Quran pronunciation. Following this study and other previous work on CAPL for Quran [35], we adopted the phoneme set as summarized in Table 2.

5.3 Voice Activity Detector

Applying this module on each speech file that contains either one verse or one complete chapter from one reciter in order to detect the position of pauses. Most of the materials used in this work were recorded in a noise clean environment such as studios, hence a simple intensity-based algorithm is used. The adopted method is controlled by three parameters, the silence threshold, the minimum speech duration and the minimum silence duration. The minimum silence duration is used to eliminate short silence segments that occurs during production of some phonemes, e.g. closure duration in plosive phonemes such as (kalkala قلقلة).The minimum speech duration used to cope with the short noise burst during silence intervals (e.g. microphone noise). Finally, the discrimination between the speech and silence segments performed based on the value of the silence threshold. Because of the variations in the voice level of the reciters and the recording environment, we computed the silence threshold for each speech file based on the values of the 5th and 95th percentile of the intensity within the current speech file.

Silence threshold value (ST) calculated as following:

$$ST = P(05) + 0.2\,[\,P(95) - P(05)\,]$$

The 5th percentile intensity value P(0.5) and the 95th percentile intensity value P (95) were used as alternative to the minimum and maximum intensity values in order to reduce sensitivity to outliers.

5.4 Automatic alignment of speech corpora

In this section we describe in details the automatic alignment method which used to obtain the time boundary of each phoneme in the speech corpora VER and CHAP. The VER corpus was first aligned and used for building initial HMM acoustic model as it is segmented into verses and hence gives more accurate alignment. The resultant HMM acoustic model is then used for aligning the CHAP corpus.

Since each speaker has around 3 hours of speech data, we built a speaker dependent acoustic model that is used for aligning speech data of each speaker. Although each speech file in the VER corpus contains exactly one verse, a common behaviour by reciters that part of the verse is repeated once or more. Therefore, simple forced alignment method will lead to an inaccurate phoneme alignment. To cope with this issue, the speech file is first passed through a VAD module which detects the existence of pauses and their positions. If no pauses are detected, it is most likely that this speech file has no repetition and contains exactly the verse phoneme sequence. This process repeated for all speech files for a specific speaker and filtering out all files with pauses and using the rest of the data to build a flat-start speaker-dependent HMM acoustic model.

The speech files that contain pauses are first segmented into short segments and then decoded using these initial speaker-independent acoustic model along with a bi-gram language model created for each verse. Furthermore, we used all the speech data of specific speaker and trained a final speaker-dependent HMM acoustic model and then re-aligned its speech files to produce more accurate phoneme time boundaries.

This process was repeated for all VER speakers and finally we trained one speaker-independent HMM acoustic model using the speech files from all speakers. The VER-valid dataset is transcribed manually in word-level and then used for tuning the decoding parameters, namely the language model scale and insertion penalty, using bigram LM created for each chapter.

On the other hand, the CHAP speech corpus, which contains recording of one complete chapter in one speech file, is first segmented into short segments using the VAD and then each segment was decoded using the VER speaker-independent acoustic model along with a bi-gram LM created for each chapter. To reduce the chance of using incorrect alignments data, we accept only the chapters that more than 95% of its words appeared in the recognition output of its segments.

All HMM acoustic models are tied-states context-dependent with 32 mixtures per state. The models trained using 13 MFCC features extracted from 25 msec window sampled every 10 msec. The delta and acceleration are further computed to form an input feature vector of size 39.

5.5 Speech attribute detection

The speech attributes of the Arabic language are a controversial issue and some of them are not agreed among all linguistics. In this work we adopted 38 attributes as listed in Table 3 following mainly the study in [36].

Table 3 Speech attributes for Quranic Arabic and the corresponding phoneme

	Features	phoneme	Features	Phoneme
Places of articulations	Oral cavity	a:, u:, i:	Nasal cavity	m1, m3, n1
	Pharynx	@, h, ~@, ~h, g_h, x	Interdental	Z, ~z, t_h
	Deep tongue	q, k	Alveolar	t, d, s, n, z, T, D, S, r, l
	Middle tongue	j, s_h, y	Post-alveolar	s_h, j
	Tongue tip	T, d, t, Z, ~z, t_h, S, z, s, n, r	Palatal	Y
	Tongue border	D, l	Velar	x, g_h, k
	Labial	f, m, w, b	Uvular	Q
	Bilabial	b, m, w	Pharyngeal	~h, ~@
	Labiodental	F	Glottal	@, h
Manners of articulations	Whisper	f, ~h, t_h, h, s_h, x, S, s, k, t	Deviate	l, r
	Strength	@, j, d, q, T, b, k, t	Hiding	h, a:, u: , i:
	Moderate	l, n, ~@, m, r	Echo	q, T, b, j, d
	Softness	D, f, g_h, h, ~h, s, S, s_h, t_h, w, x, y, z, ~z, Z, a, a:, i, i:, m1, m3, n1, u, u:	Stops	b, t, T, d, D, k, q, @
	Silence	Sil	Fricatives	f, s, S, z, t_h, ~z, Z, s_h, x, g_h, ~h, ~@, h
	Elevation	x, S, D, g_h, T, q, Z	Affricates	j
	Adhesion	T, Z, S, D	Glides	y, w
	Whistle	S, z, s	Lateral	L
	Prolongation	D	Vowels	a:, u:, i:, u , a, i

For each attribute we built a binary DNN-based classifier to discriminate between frames where this specific attribute exists (positive samples) and other frames where the attribute is absent (negative samples).

The DNN classifier consists of 6 fully connected hidden layers with typically 2048 neuron in each layer. The output layer is a soft max layer consisting of 2 neurons, one of which is fired in case of positive sample while the other one is fired in case of negative one. The rectifier linear units (RELU) activation function was adopted for all hidden neurons. The RELU function is proved to speed up the training of the DNN and avoid the vanishing gradient problem. Therefore, the time and resource consuming pre-training step becomes less effective and hence we did not perform it. The binary cross entropy was used as an objective function.Unlike the HMM acoustic model, the DNN was trained using filter bank features which are used commonly with DNN speech models and achieved better performance over the traditional MFCC features [36]. We extracted 21 filter banks from each 25 msec window and then computing the delta and acceleration. We further concatenated each 11 frames (5 frames preceded and 5 frames succeeded the current frame) to form an input feature vector of size 693 per sample.

The mini-batch stochastic gradient descent (SGD) method was utilized for the fine tuning of the DNN model with batch s newbob method where the learning rate starts with 0.1 and remains constant for the following epochs as long as the improvement of the classification accuracy of the validation size of 200 samples. The learning rate was controlled by the set is greater than 0.05.

Once the improvement in the classification accuracy of the validation set fell under the 0.05, the learning rate scaled by 0.5 during each of the remaining epochs. The training is terminated when the learning rate reaches a minimum value of 0.00001.

Furthermore, we adopted the dropout regularization technique to alleviate the effect of the overfitting over the training data [37]. The idea is to dropout part of the neurons in each hidden layer in the training phase by removing their connections to the neurons in the next and previous layers and not updating their weights during the dropout epoch. This performed by ignoring each neuron with probability and keep it with probability ( ) in each training epoch. On the other hand, all neurons will be fully connected during test with weights multiplied by p.

The samples from all phonemes belongs to a specific attribute were used as the positive samples in the training of the binary classifier while the negative samples are chosen from the frames of the others phonemes. To imbalance training of the classifier, we choose equal number of positive and negative samples for each attribute. Moreover, the training samples distributed equally over all phonemes.

The samples from all phonemes belongs to a specific attribute were used as the positive samples in the training of the binary classifier while the negative samples are chosen from the frames of the others phonemes. To imbalance training of the classifier, we choose equal number of positive and negative samples for each attribute. Moreover, the training samples distributed equally over all phonemes.

4.6. Anti-phone modelling

The frames of each phoneme is first passed through all the speech attribute detectors and only the positive output is taken to form a 38 dimensions speech attribute feature vector for each frame. These features are then used to feed a phoneme-specific binary SVM classifier to discriminate between each phoneme and all other phonemes. Each SVM classifier is tuned separately to obtain the optimum parameter per phoneme model. The tuning parameters are the kernel (rbf, sigmoid), C and gamma parameters. The best parameters are chosen to maximize the frame-level accuracy on a separate validation set. Both training and validation sets contain equal number of frames from the current phoneme and other phonemes.

To perform phoneme-level evaluation, for any specific segment if the majorety of its frames classified as belonging to the current phoneme, the whole segment is considered from this phoneme and conversely. We used two metrics to evaluate the performance of the system, the false-acceptance rate ( ) and false-rejection rate ( ) which are calculated as follow:

$$\mathbf{FAR} = \frac{\mathbf{FA}}{\mathbf{TR} + \mathbf{FA}} \tag{2}$$

$$\mathbf{FAR} = \frac{\mathbf{FR}}{\mathbf{TA} + \mathbf{FR}} \tag{3}$$

where TA, TR, FA and FR are the true-acceptance, true-rejection, false-acceptance and false-rejection samples respectively.

Furthermore, instead of train the anti-phone model as a binary classification between each phoneme and all other phonemes, we limited the anti-phone list to the most common mistaken phonemes in reciting Quran as explained in Table 4.

Table 4 Quran common pronunciation errors

Phone	D	S	T	Z	d	g_h	j	n1
Errors	z, d	z, s	d, t, D	d, ~z, z	t	X	s_h	n

Phone	Q	~@	~z	Z	a	U	i
Error	K	~h	t_h, d, z	S	a:	u:	i:

6. EXPERIMENTAL RESULTS

6.1. The speech attributes detection

In this experiment we trained one binary DNN classifier for each attribute to discriminate between frames belongs to this attribute and frames where the attribute is absent. We use two data sets the CHAP and VER-train for the training and the VER-valid and VER-test for validation and testing of the attribute detectors respectively. The VER-valid was used only to control the learning rate scheduling and early stopping of the training process while the training set was used for computing the gradients and updating the weights in the back propagation mechanism. The final accuracy reported with the VER-test dataset. As aforementioned, the number of positive samples, where the attribute exists, and negative samples, where the attribute is absent, in both the training, validation and test datasets is balanced and hence we used the frame level accuracy as our performance measure. Table 5 summarizes the overall accuracy and the number of samples of each attribute in the training, validation and testing datasets.

Overall, the manners of articulation behave better than the places of articulation with average test accuracy of 84% and stander deviation of 4.7% compared to 83% and stander deviation of 4.4% respectively. The “spreading” attribute achieved the best test performance of 94% followed by “affricates” and “Post-alveolar” of 91% each.

We further adopted the dropout as a regularization technique to cope with the overfitting problem and improve the model generalization. The dropout value is fixed to 0.3 for the input layer and 0.2 for all hidden layers. The effect of using dropout is summarized in Figure 2. As shown in the figure, the dropout improved the performance of almost all the attribute detectors by 14% to 1% reduction in the error rate.

Table 5 The FAR and FRR of the anti-phone models

	Speech Attribute	**Number of samples**			**Overall Accuracy (%)**		
		Training	Validation	Testing	Training	Validation	Testing
Places of articulation	Oral cavity	2042210	449286	408442	95.8	85.1	85.1
	Pharynx	927860	204129	185572	88.1	78.4	77.9
	Deep tongue	334610	73614	66922	95.0	87.3	86.8
	Middle tongue	368060	80973	73612	93.9	85.0	84.9
	Tongue tip	1606020	353324	321204	91.1	77.9	77.7
	Tongue border	530960	116811	106192	96.2	82.6	82.2
	Labial	1150060	253013	230012	91.9	76.9	77.4
	Bilabial	1007320	221610	201464	90.2	77.9	78.4
	Labiodental	142740	31403	28548	96.6	84.7	85.2
	Nasal cavity	472690	103992	94538	94.3	88.0	87.7
	Interdental	149720	32938	29944	100.0	79.0	80.3
	Alveolar	1987260	437197	397452	90.3	76.8	76.6
	Post-alveolar	122590	26970	24518	100.0	91.4	91.5
	Palatal	245470	54003	49094	92.0	86.3	87.0
	velar	246640	54261	49328	97.9	85.2	84.0
	Uvular	165360	36379	33072	94.1	87.9	87.0
	Pharyngeal	233570	51385	46714	100.0	87.7	88.0
	Glottal	616900	135718	123380	92.8	78.6	77.8
manner of articulation	Whisper	1255000	276100	251000	93.8	86.6	86.1
	Strength	1317780	289912	263556	93.8	83.2	83.7
	Moderate	1999410	439870	399882	93.3	76.3	76.5
	Softness	3317380	729824	663476	95.7	74.7	75.0
	Silence	3502210	770486	700442	96.5	90.6	88.4
	Elevation	410360	90279	82072	96.5	86.8	86.7
	Adhesion	167610	36874	33522	98.9	85.5	88.0
	Whistle	241050	53031	48210	97.8	85.7	89.6
	Prolongation	25250	5555	5050	94.4	87.6	85.8
	Spreading	58810	12938	11762	100.0	95.1	94.8
	Deviate	856430	188415	171286	99.6	81.3	81.1
	Hiding	2366440	520617	473288	93.0	84.1	84.0
	Echo	657100	144562	131420	99.1	85.9	86.2
	stops	1279250	281435	95.0	95.0	82.5	82.8
	Fricatives	1227510	270052	245502	94.8	81.8	81.7
	Affricates	63780	14032	12756	99.8	90.0	91.4
	Glides	509770	112149	101954	92.8	80.8	80.5
	Affricates	63780	14032	12756	99.8	90.0	91.4
	Glides	509770	112149	101954	92.8	80.8	80.5
	Lateral	505710	111256	101142	95.5	84.2	83.5
	Vowels	4109510	904092	821902	95.3	79.2	79.1
	Repetition	350720	77158	70144	96.8	85.7	85.9

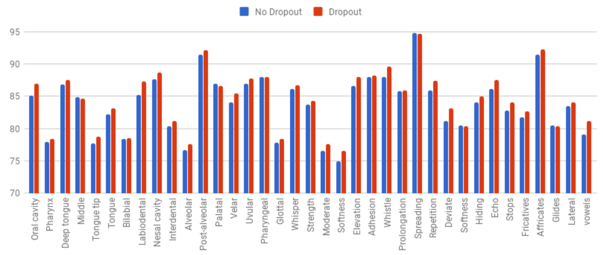

Figure 2 The effect of the dropout regularization method

In order to demonstrate the powerful of the attribute features in discriminating between phonemes, we draw a scatter plot for the speech attribute features of each pair of phonemes that are considered similar in articulation such as /m/ and /n/, /q/ and /k/, /t/ and /d/, etc. The t-SNE [38] is used to project the speech attribute feature vector from 38 to 2 dimensions. Figure 3 shows the 2D scatter plot of random samples selected from the validation set of 6 confusable phoneme pairs. It is obvious from the figures that each phoneme has clear separate region(s) with some minor overlaps.

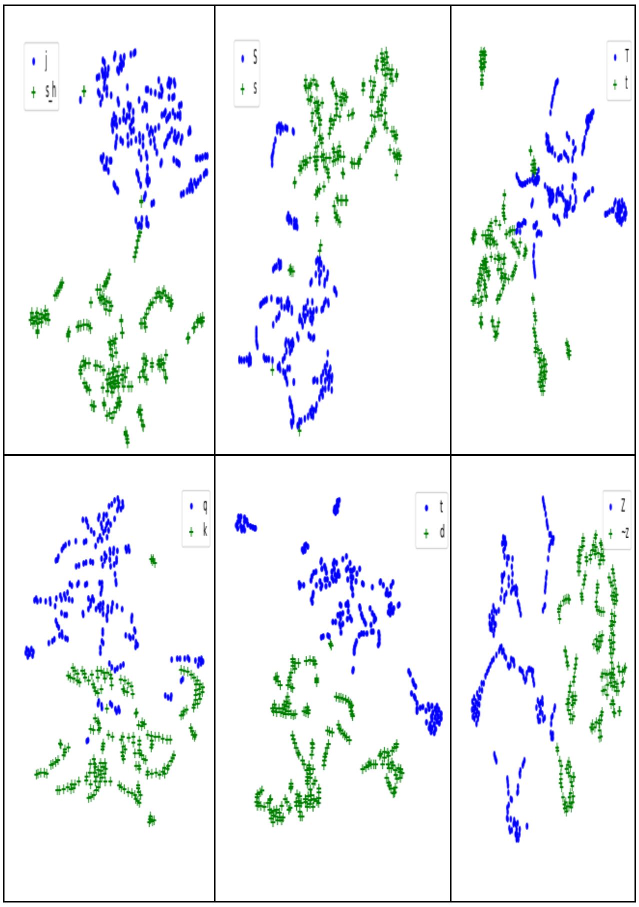

Figure 3. 2D scatter plot of the speech attribute features of 6 pairs of confusable phonemes

6.2 The anti-phone models

In this experiment, we trained a phoneme-specific binary SVM anti-phone model to discriminate between each phoneme and all other phonemes. First, the speech attribute features are extracted from each frame using the pre-trained speech attribute detectors and then fed the SVM classifiers. Here also the training was performed using the CHAP and VER-train corpora while the VER-valid and VER-test were used for the validation and testing of the SVM anti-phone models. The optimal model parameters were selected to achieve the best frame-level accuracy over the validation set. The final performance evaluation reported using the testing set and computed on phoneme-level. The whole phoneme segment is assigned to the class where the majority of its frames are classified as. The training and validation sets for all anti-phone models are balanced over the two classes which means that 50% of the samples are selected from the underlying phoneme and 50% are distributed equally over all other phonemes. While the evaluation was performed on all the phoneme samples. Due to the imbalance in the testing dataset, we adopted the FAR and FRR as our evaluation metrics. The results are summarized in Table 6 along with the number of samples of each phoneme in the testing dataset. Overall, the average FAR and FRR are 7.6% and 13.8% with standard deviations of 7.32% and 5.23% respectively. The fricatives /s_h/, /s/, /~h/ and /S/ achieved FAR lower than 2% and FRR of 8.5%, 21.7%, 12.5% and 12% respectively. The short vowels /a/, /u/ and /i/ obtained the worst FAR of around 26%, 30% and 27% respectively. Even though the long vowels, /a2/, /u2/ and /i2/, are acoustically similar to the short vowels, they are in fact the elongated version of the short vowels, their anti-phone models perform much better than the short vowels with FAR of 8.6%, 5.6% and 9.8% and FRR of 9.9%, 8.8% and 8.5% respectively.

Moreover, instead of the anti-phone models which trained to differentiate between each phoneme and all other phonemes, we investigated a binary SVM classifier that discriminate between each phoneme and the most commonly confusable phonemes in Quranic Arabic recitation as listed in Table 4. The best discrimination is between /s_h/ and /j/ with FAR and FRR of 5.2% and 3.8% respectively. While there are high confusion between /n/ and /n1/ with FAR and FRR of 22% and 30.7% respectively. Interestingly,the models of the short vowels (/a/, /u/, /i/) can effectively discriminate between them and their elongated versions (/a2/, /u2/, /i2/) with FAR of 12.7%, 12.5% and 17.3% and FRR rates of 10.6%, 8.3% and 10.7% respectively.

Table 6 The FAR and FRR of the anti-phone model

Phoneme	N# Samples	FAR (%)	FRR (%)	Phoneme	N# Samples	FAR (%)	FRR (%)
~@	1796	4.07	13.03	S	681	1.71	11.89
@	4254	14.81	17.87	s_h	614	1.61	8.47
b	3034	7.88	10.65	t	8000	4.41	11.20
d	1678	166.87	13.59	T	2455	3.98	13.10
D	274	4.65	12.41	t_h	420	6.82	17.92
f	2118	6.08	16.67	w	385	5.40	14.63
g_h	265	4.21	20.38	x	3370	2.24	17.22
h	3158	7.90	25.11	y	511	6.16	12.69
~h	940	1.69	12.55	z	2726	2.77	13.22
j	924	2.02	14.83	~z	348	4.24	13.59
k	2173	5.25	11.73	Z	1567	3.83	27.97
l	7252	5.12	17.50	a	118	26.17	6.32
m	5174	18.94	10.20	a:	20390	8.58	9.90
n	3761	11.11	18.93	i	6878	27.69	9.94
n1	1762	7.58	17.48	i:	6830	5.79	8.87
r	3346	6.6	15.2	u:	4586	9.87	8.57
q	1329	3.86	13.62	u	1804	30.55	10.29
r	3346	6.6	15.2	u:	4586	9.87	8.57
s	1704	1.68	21.77				

Table 7 The FAR and FRR of the binary SVM models discriminating between each phoneme and most commonly confusable phonemes

Phoneme	Common Errors	FAR (%)	FRR (%)	Phoneme	Common Errors	FAR (%)	FRR (%)
D	z, d	5.46	14.96	n	n1	22.07	30.70
S	z, s	10.28	7.64	q	k	5.75	9.71
T	d, t, D	3.79	12.86	~@	~h	9.26	2.34
Z	d, ~z, z	5.15	4.24	~z	t_h, d, z	16.80	16.53
d	t	10.14	3.69	a	a:	12.69	10.68
g_h	x	19.57	2.64	u	u:	12.58	8.37
j	s_h	5.21	3.79	i	i:	17.33	10.79

7. CONCLUSION

In this paper we explored the speech attribute features in anti-phone modeling for pronunciation verification of Quranic Arabic. Firstly, a bank of speech attribute detectors, namely the manners and places of articulation, were built for estimating the existence or absence of each specific attribute. These detectors are based on DNN architecture fed by filter bank features extracted from each speech frame.

The attribute detectors achieved average accuracies of 84%±4.7% and 83%±4.4% for the places and manners of articulations respectively.

For each phoneme we then built a binary SVM anti-phone model to classify each frame as belongs to the underlying phoneme or any other phoneme. The anti-phone models trained using speech attribute features derived from the pre-trained speech attribute detectors. The average phoneme-level FAR and FRR of the anti-phone models are 7.6%±7.3% and 13.8%±5.3% respectively.

Pronunciation verification method for computer aided pronunciation training system.

The future work is to evaluate the system using influent speech dataset to demonstrate its effectiveness in mispronunciation detection. Furthermore, the system will be extended to learning Arabic as a second language.

8. REFERENCES

[1] A. R. A. Mahmoud, "The Role of e-Learning Software in Teaching Quran Recitation," in Advances in Information Technology for the Holy Quran and Its Sciences (32519),

[2] A. Neri, O. Mich, M. Gerosa, and D. Giuliani, "The effectiveness of computer assisted pronunciation training for foreign language learning by children," Computer Assisted Language Learning, vol. 21, pp. 393-408, 2008.

[3] Olatunji, S. et al ," I dentification of Question and Non-Question Segments in Arabic Monology Based on Prosodic Features Using Type-2 Fuzzy Logic System,", Second International conference on Computational Intelligence, Modelling and Simulation, IEEE computer Society, pp: 149-153 (2010).

[4] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, 1997, pp. 1471-1474.

[5] Witt, S.M., Use of speech recognition in Computerassisted Language Learning, PhD thesis, Department of Engineering, University of Cambridge, 1999.

[6] Witt, S.M. and Young, S., "Phone-level Pronunciation Scoring and Assessment for Interactive Language Learning", Speech Communication 30, 95-108, 2000.

[7] S. Kanters, C. Cucchiarini, and H. Strik, "The goodness of pronunciation algorithm: a detailed performance study," 2009.

[8] T. Pellegrini, L. Fontan, J. Mauclair, J. Farinas, and M. Robert, "The goodness of pronunciation algorithm applied to disordered speech," in Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[9] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," Speech Communication, vol. 67, pp. 154-166, 2015.

[10] H. Franco, L. Ferrer, and H. Bratt, "Adaptive and discriminative modeling for improved mispronunciation detection," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 7709-7713.

[11] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," Speech Communication, vol. 51, pp. 896-905, 2009.

[12] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, "Pronunciation error detection method based on error rule clustering using a decision tree," in Ninth European Conference on Speech Communication and Technology, 2005.

[13] H. Ryu and M. Chung, "Mispronunciation Diagnosis of L2 English at Articulatory Level Using Articulatory Goodness-Of-Pronunciation Features," in Proc. 7th ISCA Workshop on Speech and Language Technology in Education, pp. 65-70.

[14] Truong, K., Neri, A., De Wet, F., Cucchiarini, C., and Strik, H., "Automatic detection of frequent pronunciation errors made by L2-learners", Proceedings of Interspeech, 1345-1348, 2005.

[15] H. Strik, K. P. Truong, F. d. Wet, and C. Cucchiarini, "Comparing classifiers for pronunciation error detection," in Eighth Annual Conference of the International Speech Communication Association, 2007.

[16] T. Zhao, A. Hoshino, M. Suzuki, N. Minematsu and K. Hirose, "Automatic Chinese pronunciation error detection using SVM trained with structural features," 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 473-478, doi: 10.1109/SLT.2012.6424270.

[17] A. M. Harrison, W.-K. Lo, X.-j. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in International Workshop on Speech and Language Technology in Education, 2009.

[18] M. Shahin, B. Ahmed, A. Parnandi, V. Karappa, J. McKechnie, K. J. Ballard, et al., "Tabby talks: An automated tool for the assessment of childhood apraxia of speech," Speech Communication, vol. 70, pp. 49-64, 2015.

[19] W. Li, S. M. Siniscalchi, N. F. Chen, and C.-H. Lee, "Improving non-native mispronunciation detection and enriching diagnostic feedback with DNN-based speech attribute modeling," in Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, 2016, pp. 6135-6139.

[20] E. Mourtaga, A. Sharieh, and M. Abdallah, "Speaker independent Quranic recognizer based on maximum likelihood linear regression," in Proceedings of world academy of science, engineering and technology, 2007, pp. 61-67.

[21] B. Abro, A. B. Naqvi, and A. Hussain, "Qur'an recognition for the purpose of memorisation using Speech Recognition technique," in Multitopic Conference (INMIC), 2012 15th International, 2012, pp. 30-34.

[22] S. M. Abdou, S. E. Hamid, M. Rashwan, A. Samir, O. Abdel-Hamid, M. Shahin, et al., "Computer aided pronunciation learning system using speech recognition techniques," in Ninth International Conference on Spoken Language Processing, 2006.

[23] M. S. Elaraby, M. Abdallah, S. Abdou, and M. Rashwan, "A Deep Neural Networks (DNN) Based Models for a Computer Aided Pronunciation Learning System," in International Conference on Speech and Computer, 2016, pp. 51-58.

[24] C.-H. Lee, M. A. Clements, S. Dusan, E. Fosler-Lussier, K. Johnson, B.-H. Juang, et al., "An overview on automatic speech attribute transcription (ASAT)," in Eighth Annual Conference of the International Speech Communication Association, 2007.

[25] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in Proc. ICSLP, 2004.

[26] V. Hautamäki, S. M. Siniscalchi, H. Behravan, V. M. Salerno, and I. Kukanov, "Boosting universal speech attributes classification with deep neural network for foreign accent characterization," in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[27] S. Zhang, W. Guo, and G. Hu, "Exploring universal speech attributes for speaker verification," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, 2017, pp. 5355-5359.

[28] Y. Wang, J. Du, L. Dai, and C.-H. Lee, "A fusion approach to spoken language identification based on combining multiple phone recognizers and speech attribute detectors," in Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on, 2014, pp. 158-162.

[29] H. Ryu, H. Hong, S. Kim, and M. Chung, "Automatic pronunciation assessment of Korean spoken by L2 learners using best feature set selection," in Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific, 2016, pp. 1-6.

[30] M. Shahin, B. Ahmed, and J. Ji, "One-Class SVMs Based Pronunciation Verification Approach," presented at the the 24th International Conference on Pattern Recognition (ICPR 2018), 2018.

[31] H. Hammady, O. Badawy, S. Abdou, and M. Rashwan, "An HMM system for recognizing articulation features for Arabic phones," in Computer Engineering & Systems, 2008. ICCES 2008. International Conference on, 2008, pp. 125-130.

[32] R. Ziedan, M. Micheal, A. Alsammak, M. Mursi, and A. Elmaghraby, "A Unified Approach for Arabic Language Dialect Detection," in 29th International Conference on Computers Applications in Industry and Engineering (CAINE 2016), Denver, USA, 2016.

[33] Every Ayah. Available: http://everyayah.com/

[34] A. Ragheb, "Quran Phonology; Quran reciting rules based on modern acoustics," M.Sc Thesis Cairo University, 2004.

[35] S. Hamid, "Computer aided pronunciation learning system using statistical based automatic speech recognition," PhD thesis, Cairo University, Cairo, Egypt, 2005.

[36] A.-r. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, 2012, pp. 4273-4276.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[38] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of machine learning research, vol. 9, pp. 2579-2605, 2008.