Research Article

Speech Attribute Detection to Recognize Arabic Broadcast Speech in Industrial Networks

M. Al Dabel

Department of Computer Science and Engineering, University of Hafr Al Batin, Hafar Al-Batin, Saudi Arabia

Correspondence should be addressed to M. Al Dabel; maldabel@uhs.edu.sa

Received 3 August 2022; Revised 5 September 2022; Accepted 16 September 2022; Published 5 October 2022

Academic Editor: Praveen Kumar Reddy Maddikunta

Copyright © 2022 M. Al Dabel. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work, we present an alternative model to Arabic speech recognitions to boost classification accuracy of basic speech units in Industrial Internet of Things. This approach integrates both top-down and bottom-up knowledge into the automatic speech attribute transcription (ASAT) framework. ASAT is a new family of lattice-based speech recognition systems grounded on accurate detection of speech attributes. Two state-of-the-art deep neural networks were used for attribute speech feature detection, convolutional neural networks (CNNs) and feed-forward neural network, with pretraining using a stack of restricted Boltzmann machines (DBN-DNN). These attribute detectors are trained using a data-driven approach for Arabic speech data using the Buckwalter code for Arabic transcription. The detectors work by detecting phonologically distinctive articulatory features of Arabic based on manner and place of articulation, which extract characteristics of the human speech production process from the speech dataset. The results of the work are shown in terms of average equal error rate inferring that the attribute model based on the CNN consistently outperforms the DBN-DNN model. Attribute detectors with CNN, versus DBN-DNN model, reduce the AEER producing 10.42% relative improvements for manner of articulation and producing 13.11% relative improvements for place of articulation.

1. Introduction

The revolution in information and communication technology has played an important role in the industrial operations and productions. The industries relay heavily on tools powered by machine learning models and algorithms to process the terabytes of data produced through sensors, actuators, industrial management systems, and other applications. These data have the features of volume (terabyte) and variety (image, audio, video, graphics), and therefore, designed models and algorithms are required for analysis and management [1–3]. Industrial Internet of Things (IIoT) can be defined as a set of machines, robotics, cognitive technologies, and computers for intelligent industrial operations with the help of data analytics and its applications such as speech recognition [1].

State-of-the-art automatic speech recognition (ASR) systems generally depend on a pattern matching scheme that represents spoken utterances as sequences of stochastic patterns [4]. There are two approaches in the literature, top-down and bottom-up. The top-down approaches are typically used to produce all constraints in a single, compact probabilistic finite state network (FSN), containing of acoustic hidden Markov model (HMM) states with emission probabilities generated by Gaussian mixture models (GMMs), phones, lexicon, grammar nodes, and their connecting arcs [5].

Automatic speech attribute transcription (ASAT) [6] is a bottom-up approach which first detects a collection of speech attribute cues and then integrates such cues to make linguistic validations. Earlier research in the ASAT system uses the articulatory-based phonological features [7–11] in a new detection-based framework. ASAT has been further investigated and used in a variety of tasks including rescoring of word lattices produced by state-of-the-art HMM systems [12], continuous phoneme recognition [13], cross-language attribute detection and phoneme recognition [14], and spoken language recognition [15]. The speech cues detected in ASAT are referred to as speech attributes.
Recently, a huge advancement has been made in neural network approaches and more specifically in training densely connected, generative deep belief nets (DBNs) with many hidden layers. The fundamental concept of the DBN training algorithm [16] is to first initialize the weights of each layer greedily in a purely unsupervised way through processing each pair of the layers as a restricted Boltzmann machine (RBM) and then fine-tune all the weights jointly to additionally boost the likelihood. The resulting DBN can be seen as a hierarchy of nonlinear feature detectors that can spot complex statistical patterns in data. In terms of classification tasks, the same DBN pretraining algorithm can be used to initialize the weights in deep neural networks (DNNs) with many hidden layers. The weights in the entire DNN can then be fine-tuned using labeled data. Research has proved that DNNs are effective tools in a number of applications, including coding and classification of speech, audio, text, and image data [17–21]. The key idea was to develop acoustic models based on DNNs and other deep learning techniques for ASR. For example, the context-independent DNN-HMM hybrid architectures have recently been proposed for phoneme recognition [22, 23] and have achieved a significant performance. A recent acoustic model, the context-dependent (CD)-DNN-HMM proposed in [24], has been well employed to large vocabulary speech recognition tasks and can reduce word error rate by up to one third on the challenging conversational speech transcription tasks in comparison with the discriminatively trained conventional CD-GMM-HMM systems [25].

The main goal of the work is to improve the existing Arabic recognition system. In order to refine the whole system, we have to improve components it consists of. This is done through improving the front-end speech articulatory detectors by (i) defining speech articulatory features in terms of manner and place that can capture Arabic language variations; (ii) implementing a model for detecting speech attributes of articulation using deep learning approaches; (iii) the model satisfying the Arabic articulatory feature paradigm; and finally (iv) investigating performance of the model and applying to the phone lattice resoring for speech recognition.

The remainder of the paper is organized as follows. The system framework is described in Section 2. Next, the experimental setup is given in Section 3 in which experimental results on attribute classification and phone lattice resoring are presented and discussed. Finally, we discuss our findings and conclude our work in Section 4.

2. System Framework

This section introduces a framework for IIoT and the application of artificial intelligence (AI) in IIoT ecosystems (Industrial AI) for speech recognition. It focuses on Arabic speech attribute infused knowledge that should be addressed during the AI’s full lifecycle within an IIoT system. Such consideration would lead to better Industrial AI lifecycle from design to implementation and operation [1–3].

The method consists of three main components: (i) an ASR engine which generates the lattice of competing hypotheses, (ii) an attribute-based front end that generates the phonological-based related information, and (iii) a lattice resoring module. The role of the latter module is to combine the outputs (i.e., phone posterior probabilities) from the attribute-based module and acoustic likelihoods produced by the ASR engine.

2.1. The Automatic Recognition Module. The automatic speech recognition (ASR) module works by taking the speech signal as input and mapping it to a string of words. The ASR module consists of two types of probabilistic models, namely, acoustic model (AM) and language model (LM). During the AM phase, the speech input is transcribed into a set of candidate phoneme sequences. The phonemes are then assembled into possible words using a pronunciation dictionary. Turning to the LM phase, it takes the advantage of prior linguistic knowledge to evaluate every single candidate of word sequences. Then, all candidate word sequences are searched through using an efficient algorithm to find out the most probable matches of the speech input [26]. Finding the best word sequence given, the speech input is known as the decoding or search problem. The solution can be found using the well-known Viterbi algorithm [27].

It is worth noted that typically most state-of-the-art ASR systems apply two-pass decoding instead of one-pass decoding, where two language models are applied. In the first decoding pass, fast decoding algorithms are used to produce a nonoptimal search and yield a N-best list or a word lattice [28]. During the second decoding pass, complex but rather slow decoding algorithms are used to search within the N-best list or word lattice. This is due to the fact that the used LM in the first pass is a well-structured model with low tolerance of latency, in which a heavily pruned n-gram model is applied to build the decoder graph, whereas the adapted LM in the second-pass decoding is a more advanced but slower model. In the second LM, a more complex LM is used to rescore hypotheses generated in the first pass. In this work, we adapt neural network-based language models (NNLMs) for the second-pass LM to rescore hypotheses.

The DNNLMs [29] work by learning a distributed representation of words with the probability function of word sequences through a concatenation of the representations. The input is created depending on a fixed length history (i.e., the preceding $n - 1$ words), such that each word is encoded using a one-hot representation. This model has two hidden layers: the shared word feature layer and the ordinary hyperbolic tangent hidden layer.

In particular, the probability of a sequence of words can be computed from the probability of every individual word given the context of previous words based on the chain rule of probability (as a result of Bayes theorem):

$$P(w_1, w_2, \ldots, w_{t-1}, w_t) = P(w_t | P(w_1) P(w_2 | w_1) P(w_3 | w_1, w_2) \ldots$$

$$P(w_t | w_1, w_2, \ldots, w_{t-1}).$$

(1)

The probabilistic prediction $P(w_t | w_1, w_2, \ldots, w_{t-1})$, in the DNNLMs [29], is normally computed by mapping each
word \( w_{t-1} \) to an associated \( d \)-dimensional feature vector \( C_{w_{t-1}} \), which is column \( w_{t-1} \) of parameter matrix \( C \). Specifically, the learned features for word \( k \) are held in vector \( C_k \). Assume a vector \( x \) represents the concatenation of all feature vectors as follows:

\[
x = (C_{w_{t-1}}, \ldots, C_{w_{t-1}}, C_{w_{t-1}}, \ldots, C_{w_{t-1}}, \ldots, C_{w_{t-1}}).
\]

Then, a standard artificial neural network architecture for probabilistic classification is used to obtain the probabilistic prediction of the following word, beginning with \( x \), and employing the softmax activation function at the output layer.

\[
P(w_t = k|w_{t-1}, \ldots, w_{t-1}) = \frac{e^{a_k}}{\sum_{l=1}^{N} e^{a_l}}
\]

such that

\[
a_k = b_k + \sum_{i=1}^{h} W_k \tanh\left(c_i + \sum_{j=1}^{(n-1)d} V_{ij}x_j\right).
\]

The vectors \( b, c \) and matrices \( W, V \) are parameters. The \( \theta \) denotes the concatenation of all the parameters. Additionally, \( h \) denotes the number of hidden units and \( d \) denotes the number of learned word features; both play a role in handling the complexity of the model.

Finally, the neural network is trained through implementing a gradient-based optimization algorithm in order to maximize the log-likelihood as follows:

\[
L(\theta) = \sum_{t} \log P(w_t|w_{t-1}, \ldots, w_{t-1}).
\]

2.2. The Attribute-Based Detection Module. The Arabic speech attributes of interest are introduced here based on three articulatory dimensions: manner, place, and voicing, and then the implementation details of the detectors are described.

Arabic is one of the widely common languages around the globe, with about hundreds of millions speaking Arabic as a mother tongue along with tens of millions speaking Arabic as a second language. The language has its own pronunciation, spelling, and grammar rules, with twenty-eight letters. The consonants and long vowels are described by letters, while the short vowels are described by diacritic signs. These signs are applied above or below the letters associated with consonants, whereas in the short vowel the diacritic signs come just after.

Consonant letters are classified in terms of place (or point) of articulation, manner of articulation, and voicing of different speech sounds (phonemes) [31, 32]. It is referred as phonological features or acoustic-phonetic speech attributes. In terms of human speech production system, place and manner of articulation depend on where and how the sound is generated in the vocal tract. Voicing, on the other hand, is the feature that represents the vibration of vocal cords resulting in two types of speech sounds: voiced and unvoiced.

An important aspect of Arabic, as a phonetic language, is that each letter corresponds to one phoneme, unlike some other languages, such as English, in which the same letter may correspond to different phonemes. Therefore, when expressing the phonetic transcription one needs a method that relies heavily on the complexity of the spelling and pronunciation system of the language under study. The commonly used phonetic alphabets in speech technologies are Speech Assessment Methods Phonetic Alphabet (SAMPA) code [33] and International Phonetic Alphabet (IPA) code [34]. The aforementioned codes are usually used for several languages meaning they are independent of any particular language and applicable to all languages. The third code is the Buckwalter code [35] which was developed for the purpose of Arabic phonetic transcription. The Buckwalter [35] is the popular Arabic transcription scheme for extracting, storing, and representing Arabic text. It is a precise transcription that complies the spelling conventions of language as well as substitutes a one-to-one mapping that is thoroughly varying in nature to have all the required information for a good pronunciation. For that, the Buckwalter code is used in this work.

Turning to the speech attributes, Table 1 lists the phoneme matrix mapping of corresponding Arabic consonants [31] using Buckwalter transliteration [35]. It implies place of articulation, manner of articulation, and voicing. In particular, Arabic has twenty-eight consonants: eight of them are plosives, thirteen are fricatives, one affricate, two nasals, one lateral, one trill, and two semi-vowels. Additionally, it has three short vowels and three long vowels. The line across the top of the chart expresses the place of articulation ranging from front-most point of articulation at the lips to the farthest-back point. The far-left column expresses the manner of articulation defining the amount of closing of the articulators. Generally speaking, the articulators are described as the parts of the vocal tract, which are applied to yield different speech utterances. For such, there are four cases: (i) if the airflow is blocked completely (plosives are produced), (ii) if the airflow is blocked and then released into a fricative (affricates are produced), (iii) if the airflow is restricted but allowed through (fricatives are produced), and (iv) if the air flows smoothly (resonants are produced including nasals, laterals, and semi-vowels) [32].

In this work, we use 7 manners of articulation classes: plosives, affricates, fricatives, nasals, laterals, trill, and semi-vowel; and 9 places of articulation classes: labial, labiodental, interdental, alveolar, palatal, velar, uvular, pharyngeal, and glottal, as shown in Table 2. A further class is defined for “other” which includes any noise or unknown sound events. Studies in the literature have derived a universal set of speech attributes which are extracted for a particular language and then shared across many different languages (e.g., [14, 36]). In this work, a different strategy is adapted in which the speech attributes are defined specifically for Arabic language by taking into account its phonological characteristics.

The aim of a detector is to estimate the probability that an observation (or feature) relates to an attribute class. In this work, we use a single-label detection to classify the Arabic speech attributes rather than using multilabel task.
that associates each attribute example with multiple labels (classes). It is one of the methods used to extract the attributes features from the speech [37, 38]. The idea behind single labeling is that each attribute correlates with a single label (class) and this is more applicable to the Arabic language. During the detection task, the probability measure is computed such that the observation belongs to a particular attribute class.

One approach to speech attribute detection is known as the automatic speech attribute transcription (ASAT) framework [39]. It adapts a bottom-up framework and is based on data-driven modeling. The purpose here is to generate a confidence score or posterior probability by analyzing a speech segment to preserve some phonetical feature attribute. Every attribute extractor works on a frame-by-frame basis and yields probabilities accordingly for both target class and nontarget class as well as other model (e.g., noise) if any. All these output probabilities should add up to one. Such hard decisions from the attribute detectors are not important at this initial stage due to two reasons. The first is that a new feature vector is formed from the output of the detectors. This is done by concatenating each of these posterior probabilities from each target class [39]. The advantage is that these posterior vectors can be used later on as a high-level feature in the other parts of the system. The second plays an effective role in adjusting the risk parameters and embedding additional information for decision-making through distributing the process of decision-making to the other parts of the system [40].

Early attempts to speech attribute detection in the literature [13, 40, 41] suggest using the ASAT framework with two subsystems. The former subsystem has a bank of attribute detectors that generate the outputs in the form of confidence scores, and the latter has an evidence merger that combines low-level features into high-level feature (e.g., phone posteriors). This is followed by block where the outputs resulted from the extractors are concatenated together to produce a new vector. Each of the attribute detectors is trained using shallow artificial neural networks (ANNs) with cross-entropy cost function and softmax output layer. The purpose is to get the posterior probabilities having the speech frame.

A new approach to attribute detector modeling [40] uses deep neural network (DNN), which is pretrained with a stack of restricted Boltzmann machines (RBMs), to model each new layer of higher-level features. In particular, all attribute detectors are trained using a single DNN resulting in a posterior probability vector following the softmax layer. The vector outlines all probabilities in order to associating them to attribute classes (target, nontarget, and other). More specifically, it represents single-label detector instead of outputting posterior probability for each attribute detector for three attribute classes. In fact, despite the limitation, the model reports an improvement in the results for foreign accent recognition task [40].

Recently, researchers in [42, 43] extend the idea of the work in [40] using single DNN that models all attribute extractors and generates the output as a confidence score. Additionally, the cross-entropy cost function is compensated with mean square error (MSE) function, and the

| Table 1: Phoneme list in terms of place of articulation, manner of articulation, and voicing. |
|---------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Plosives                        | Labial| Labiodental | Interdental | Alveolar | Palatal | Velar | Uvular | Pharyngeal | Glottal |
| Voiceless                       | [d1]  | [d2]  | [d3]  | [d4]  | [d5]  | [d6]  | [d7]  | [d8]  | [d9]  | [d10] | [d11] | [d12] | [d13] | [d14] |
| Voiced                          |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Affricates                      | [d15] | [d16] | [d17] | [d18] | [d19] | [d20] | [d21] | [d22] | [d23] | [d24] |
| Voiceless                       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Fricatives                      |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Nasals                          |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Laterals                        |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Trills                          |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Semi-vowels (Or approximants)   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |

The mapping is adapted from [31] using Buckwalter transliteration [35] which is an Arabic transcription scheme.

| Table 2: The manner of articulation classes and place of articulation classes used in this work. |
|---------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Articulation                    | Phonomological classes |
| Manner of articulation (7 classes) | Stops, affricates, fricatives, nasals, laterals, trills, and semi-vowels (or a so-called approximant) |
| Point of articulation (9 classes) | Labial, labiodental, interdental, alveolar, palatal, velar, uvular, pharyngeal, and glottal |
output softmax layer is compensated with sigmoid layer. The authors reported that the detection results of their system are posterior scores which determine the existence of every single attribute with a value between 0 and 1 resulting in the sigmoid units in the output layer of neural network.

In this work, the algorithm proposed by [42] is used to detect Arabic speech articulatory attributes as a single-label classification problem. During the training stage, the input vector for the DNNs is the concatenated context of speech frames. The output label vector is described as a binary vector that is equivalent to the amount of attribute classes (i.e., place and manner attributes). In particular, the annotation in data transcription of authors reported that the detection result of their system is measured to measure the single-label classification performance.

In this work, two types of Arabic speech articulatory attribute detectors are modeled, and these are manner (7 classes, \( M = 7 \)) and place (9 classes, \( M = 9 \)), as explained earlier. Two types of attributes are modeled with their own DNNs. Two topologies of DNN are investigated: (i) the deep neural network with layer-wise RBM pretraining (DBN-DNN) and (ii) 1D convolutional neural network (DBN-DNN).

In the training phase, the speech frame vector \( \mathbf{x}_i \) is fitted in the input of the DNNs. Furthermore, its connected target output of the network. The duration of this vector is equivalent to the number of attribute classes (7 or 9). The MSE cost function is applied to fit the label vector with the output sigmoid units. In the testing stage, the single-label detection is calculated such that the data are fed to the DNN and generate the confidence scores on the output layer. The resulted scores are then used as inputs in other modules of the system.

In particular, assume that a training set is \( \mathcal{T} = \{ (\mathbf{x}_i, y_i) | i = 1, ..., |\mathcal{T}| \} \), such that \( t \) denotes pairs of training samples \( \mathbf{x}_i \in \mathbb{R}^D \) of dimension \( D \) and binary vector of labels \( y_i \in \{0, 1\}^M \). The number of classes is denoted by \( M \) such that \( \mathcal{C} = \{ C_k | k = 1, ..., M \} \). The aim of the learning system is to train a single-label classifier in which the binary vector of labels \( y_i \) has to assign sample \( \mathbf{x}_i \) to unique classes, meaning that every data point \( \mathbf{x}_i \) has to be assigned to one class \( C_k \).

In this work, the algorithm proposed by [42] is used to detect Arabic speech articulatory attributes as a single-label classification problem. During the training stage, the input vector for the DNNs is the concatenated context of speech frames. The output label vector is described as a binary vector that is equivalent to the amount of attribute classes (7 or 9). The MSE cost function is as follows:

\[
F_1^\mu = \frac{2 \sum_{k=1}^{M} TP_k}{\sum_{k=1}^{M} (TP_k + 2FP_k + FN_k)},
\]

where \( TP_k \) denotes true positives, \( FP_k \) denotes false positives, and \( FN_k \) denotes false negatives, respectively.

2.3. Lattice Rescoring Module. The purpose of the rescoring algorithm is to integrate the confidence scores generated by the detection-based module, into the phone lattice on an arc-by-arc basis [12]. Rescoring is preformed as a linear combination of the log-likelihood acoustic score generated by the baseline ASR system and the logarithm of the phone posterior probability. It reflects the final stage in this work which verifies whether DBN-DNN-boosted accuracies can help improve the ASR performance. This is done by integrating the information generated at the output of the DBN-DNN phoneme classifier, phoneme posterior probability, into an existing ASR system through the phone lattice rescoring procedure.

A multistage decoding technique [45] is adapted in order to integrate articulatorily motivated knowledge into the ASR system. First, a speech decoder generates a collection of competing speech hypotheses. It is then followed by a rescoring algorithm to re-rank these hypotheses by incorporating additional information not used in the decoding process.

The lattice structure [46], in this work, reflects the syntactic constraints of the grammar used during recognition. The graph, \( G(N, A) \), is direct, acyclic, and weighted, with \( N \) nodes and \( A \) arcs. The timing information is integrated in the nodes meaning that temporal boundaries are given by the arcs’ bounding nodes; the arcs, on the other hand, embedded the symbol along with the score information, in which every arc corresponds to a recognized phone or word.

Minimum Bayes risk (MBR) rescoring [47] and ROVER [48] are two of well-known rescoring paradigms in the literature. The objective cost function in the MBR rescoring algorithm utilizes the expected Levenshtein distance to re-rank the string hypotheses generated by the speech decoder. ROVER, on the other hand, is a commonly used tool that works by producing a confusion network through a multiple string alignment. Then, a voting scheme is done to get the 1-best hypotheses. This technique sometimes leads to a significant performance improvement as long as the individual systems have similar performance and comparable complexity, and exhibit different error patterns. The rescoring algorithm used in this work incorporates scores generated by the detection module into the speech lattice, and it is inspired to the decoding scheme based on a generalized confidence score proposed in [49].

In particular, combining independent sources of information can be achieved by the following formula:

\[
p(\mathbf{a}_i | \Lambda) = C \prod_{n=1}^{N} p(\mathbf{a}_i | \Lambda^1)^{\mu},
\]

where \( \Lambda^1 \) displays the set of acoustic parameters for the \( r \)th system, \( \Lambda \) is the set of acoustic parameters of the combined
system, \( p(\alpha_i | A^i) \) are different independent sources, \( C \) is a normalization constant, and \( \alpha' \) is the \( i \)th interpolation weight. Care needs to be taken since in the log-space, and the above multiplication of exponentially weighted terms becomes a weighted sum.

In the presented work, we assume \( C \) is equal to 1, and the sum of the interpolation weights \( \alpha' \) is forced to sum up to 1. Additionally, the weighted sum is processed on an arc-by-arc basis, as a linear combination of the log-likelihood acoustic score of every arc and the logarithm of the speech attribute detector combined output which belongs to the arc phone label. The linear combination occurs at a phone level for phone lattices such that each arc in a lattice corresponds to a phone in a string hypothesis. Assuming that the updated log-likelihood value is denoted as \( S_n \) for the given arc, the rescoring formula is expressed as

\[
S_n = w_p PS_n + w_l L_n,
\]

in which \( L_n \) is the log-likelihood of the \( n \)th arc, and \( w_l \) and \( w_p \) are the interpolation weights of the log-likelihood score and the phone-level score, respectively. \( PS_n \) represents the phone-level score, and it is a linear combination (with weights all equal to one) of \( PS_{nt} \) calculated at the end of each phone.

3. Experiments and Results

In the following sections, the experimental setup is presented, and the results on attribute detection are discussed. Phone recognition results through lattice rescoring are also presented.

Previous studies use neural architectures with more than one hidden layer for speech applications due to the fact that feature vectors extracted for different phonemic or phonetic classes greatly overlap in the input feature (hyper-)space. In [50], the researchers inferred that there exists a great overlap between formant frequencies for different vowel sounds by dimensional MFCCs for the bilabial class and 39-dimensional MFCCs for the alveolar class are rather small. \( f^+ \)his suggests that speech data lie on or near a nonlinear manifold [52]. In this work, we adapted neural architectures with different numbers of hidden layer for our experiments.

3.1. Experimental Setup. All experiments were conducted on Arabic Multi-Genre Broadcast (MGB) [53], which is the second round of the Multi-Genre Broadcast (MGB) [54]. The MGB challenge 2015 [54] collected the data provided by the British Broadcasting Corporation (BBC) to experiment the automatic transcription of a set of BBC shows. The shows were meant to span the multiple genres in broadcast TV and were divided into 8 genres, namely, advice, children’s, comedy, competition, documentary, drama, events, and news. The training data, for acoustic modeling, were fixed to more than 2,000 shows. The development data contain 47 shows. The Arabic MGB [53], used in this work, is a controlled evaluation of Arabic speech to text transcription and supervised word alignment using Al Jazeera TV channel recordings. The total amount of speech data crawled using the QCRI Advanced Transcription System (QATS) [55] was about 3,000 hours of broadcast programs, whose durations ranged from 3–45 minutes. For the purpose of this work, we used further alignment by only those programs with transcription on their Arabic website. A cross-validation (CV) set was generated by extracting 600 sentences out of the Arabic MGB training set, and the remaining sentences were used as training material. Evaluation was carried out on the MGB evaluation data (about 300 utterances). Transcription of the data is used in Buckwalter format.

In addition, an automatic alignment of speech corpora is used here to obtain the time boundary of each phoneme in the Arabic MGB using a voice activity detection (VAD). The VAD method takes the speech file as an input and works by detecting all available pauses and their positions. The speech file is said to have no repetition and contains the exact phoneme sequence when no pauses detected. Such process repeated for all speech files and filtering out all files with pauses. As it was mentioned above, the MGB corpus has phoneme transcriptions. In order to train attribute detectors, the manner or place transcription was needed, and these were obtained from the phoneme transcriptions. We mapped phoneme transcription into corresponding manner or place attribute transcription using mapping tables (see Table 1).

In the presented work, it used 7 manners of articulation classes: plosives, affricates, fricative, nasals, laterals, trill, and semi-vowel; and 9 places of articulation classes: labial, labiodental, interdental, alveolar, palatal, velar, uvular, pharyngeal, and glottal. A further class is defined for “other” which includes any noise or unknown sound events.

The ASR baseline system was trained using the Arabic MGB training data, 10.2 hours transcribed by four different annotators, resulting in more than 40 hours in total. These data were augmented by applying speed and volume perturbation, increasing the number of training frames by a factor of three to about 120 hours. The used code recipe was designed with the Kaldi repository. The baseline system was trained on 250 hours sampled from the training data, which comes from 500 episodes. This system uses a standard Mel-frequency cepstrum coefficients (MFCCs) multipass decoding. The first pass uses a GMM with 5,000 tied states, and 100 K total Gaussians, trained on features transformed with maximum likelihood linear regression (FMLLR). The second pass is trained using a DNN with four hidden layers, and 1024 neurons per layer, sequence trained with the minimum phone error (MPE) criterion. A tri-gram language model is trained on the normalized version of the sample data text (250 hours). Moreover, MFCCs were produced using spectral analysis using a 23-channel Mel filter bank from 0 to 8 kHz. The cepstral analysis was then generated with a Hamming window of 25 ms and a frame shift of 10 ms. For each frame, 12 MFCC features plus the zeroth cepstral coefficient were computed. The first and second time derivatives of the cepstra were computed as well and concatenated to the static cepstra to generate a 39-dimensional feature vector. The baseline results were reported on
10-hour verbatim transcribed development set: 34% (8.5 hours) for the nonoverlap speech and 73% (1.5 hours) for the overlap speech.

3.2. Arabic Attribute Detection Analysis. Two types of neural networks were investigated to solve the problem of attribute detection. The first deep network is simple feed-forward neural network using restricted Boltzmann machine pre-training (context-dependent DBN-DNN), and the second is convolutional neural network. All experiments are conducted under the context-dependent DNN to estimate attribute posterior probabilities per speech frame using the Arabic MGB training data.

The input feature vector for DBN-DNN is a 45-dimensional mean-normalized log-filter bank features with up to second-order derivatives and a context window of 11 frames, forming an input vector of 495-dimensional (45 × 11). The number of output classes is equal to 16 (7 for manner and 9 for place). Moreover, the “other” output class is added to both DNN to detect probable unlabeled frames. DNN topologies with 512 and 1024 number of units were investigated, and the number of hidden layers is varying from 1 up to 10 (in total 20 DBN-DNN topologies).

The process of all DBN-DNN topologies is started with the stacked restricted Boltzmann machines (RBMs) using layer-by-layer generative pretraining. The pretraining algorithm is contrastive divergence with 1 step of Markov chain Monte Carlo sampling (CD-1). The first RBM has Gaussian–Bernoulli units and been trained with initial learning rate of 0.01. The following RBMs have Bernoulli–Bernoulli units and been trained with initial learning rate of 0.4. After pretraining the weights and stacking all the layers, the final output sigmoid layer was concatenated with the DNN. The number of output units correlates with the number of classes, which are responsible of generating the attribute output scores. The fine-tunning for final weight training is done by mini-batch stochastic gradient descent with learning rate of 0.0008. Mini-batch size is equal to 128 observations. The mean square error objective function was optimized during fine-tunning. In the DNN, all sigmoid hidden units were used. All settings and parameters are used following the speech community [56].

The second is a 1D convolutional neural network (CNN) topology that takes feature maps as an input to train to detect Arabic speech attributes. These feature maps are grouped from 40-dimensional log-Mel filter bank features and their first- and second-order derivatives, and context window is size of 11 frames. In total, 3 × 11 = 33 input feature maps to which we apply 1D convolution mapping along the frequency axis. The CNN contains convolutional layer and max-pooling layer and then from 1 up to 10 fully connected hidden layers; each of them has 512 or 1024 sigmoid units.

Similar to the DBN-DNN settings in case of the CNN, we optimize mean square error cost function. Output layer has sigmoid units and produces sigmoid output “posterior” scores for every attribute with 16 in total (7 for manner or 11 for place) per each speech frame.

Table 3: Best models of manner detectors, with DBN-DNNs or CNNs with hidden layers from 1 up to 10 and 512 or 1024 units in each layer.

<table>
<thead>
<tr>
<th>Topology</th>
<th># of units</th>
<th># of hidden layers</th>
<th>AEER, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>512</td>
<td>5</td>
<td>10.51</td>
</tr>
<tr>
<td>CNN</td>
<td>1024</td>
<td>2</td>
<td>10.42</td>
</tr>
<tr>
<td>DBN-DNN</td>
<td>512</td>
<td>8</td>
<td>13.44</td>
</tr>
<tr>
<td>DBN-DNN</td>
<td>1024</td>
<td>2</td>
<td>13.12</td>
</tr>
</tbody>
</table>

Table 4: Best models of place detectors, with DBN-DNNs or CNNs with hidden layers from 1 up to 10 and 512 or 1024 units in each layer.

<table>
<thead>
<tr>
<th>Topology</th>
<th># of units</th>
<th># of hidden layers</th>
<th>AEER, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>512</td>
<td>3</td>
<td>13.11</td>
</tr>
<tr>
<td>CNN</td>
<td>1024</td>
<td>3</td>
<td>13.17</td>
</tr>
<tr>
<td>DBN-DNN</td>
<td>512</td>
<td>4</td>
<td>14.81</td>
</tr>
<tr>
<td>DBN-DNN</td>
<td>1024</td>
<td>8</td>
<td>14.62</td>
</tr>
</tbody>
</table>

Table 5: Best models of all Arabic attributes for both manner and place of articulation, with DBN-DNNs or CNNs with hidden layers from 1 up to 10 and 512 or 1024 units in each layer.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Classes</th>
<th>CNN</th>
<th>CNN</th>
<th>DBN-DNN</th>
<th>DBN-DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manner of articulation</td>
<td>(512 × 3)</td>
<td>(1024 × 3)</td>
<td>(512 × 4)</td>
<td>(1024 × 8)</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>12.22</td>
<td>9.14</td>
<td>13.11</td>
<td>12.34</td>
<td></td>
</tr>
<tr>
<td>Affricates</td>
<td>11.10</td>
<td>10.21</td>
<td>12.17</td>
<td>11.44</td>
<td></td>
</tr>
<tr>
<td>Fricatives</td>
<td>13.10</td>
<td>9.33</td>
<td>12.81</td>
<td>13.14</td>
<td></td>
</tr>
<tr>
<td>Laterals</td>
<td>12.12</td>
<td>10.71</td>
<td>13.51</td>
<td>11.80</td>
<td></td>
</tr>
<tr>
<td>Trills</td>
<td>11.00</td>
<td>10.52</td>
<td>10.12</td>
<td>10.55</td>
<td></td>
</tr>
<tr>
<td>Approximants</td>
<td>11.31</td>
<td>9.11</td>
<td>12.33</td>
<td>11.61</td>
<td></td>
</tr>
<tr>
<td>Place of articulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labial</td>
<td>13.20</td>
<td>13.70</td>
<td>14.11</td>
<td>14.31</td>
<td></td>
</tr>
<tr>
<td>Labiodental</td>
<td>11.35</td>
<td>12.98</td>
<td>13.17</td>
<td>14.44</td>
<td></td>
</tr>
<tr>
<td>Interdentals</td>
<td>11.42</td>
<td>12.11</td>
<td>14.81</td>
<td>14.86</td>
<td></td>
</tr>
<tr>
<td>Alveolar</td>
<td>12.12</td>
<td>13.97</td>
<td>14.36</td>
<td>14.50</td>
<td></td>
</tr>
<tr>
<td>Palatal</td>
<td>13.11</td>
<td>13.98</td>
<td>13.81</td>
<td>13.95</td>
<td></td>
</tr>
<tr>
<td>Glottal</td>
<td>13.10</td>
<td>13.88</td>
<td>14.22</td>
<td>14.55</td>
<td></td>
</tr>
</tbody>
</table>

Convolutional layer in the CNN has 128 feature maps, each of which has size of 33 frequency bands, meaning that these are generated by convolving each input feature map with the filter size of 8 (1 + (40 − 8) = 33). After that, max-pooling layer outputs the maximum values over a non-overlapping window covering the outputs of every three frequency bands in each feature map (i.e., pooling size is 3), down-sampling the overall convolutional layer output to three times smaller. After that, the output of the max-pooling layer is the input to the fully connected feed-forward part of the CNN. All CNN settings and configurations are used following the speech community [56].

As a performance measure for Arabic attribute detection analysis, we used average equal error rate (AEER) measure [56]. The AEER is computed across all manner and place articulatory detectors and for every particular neural network model. The average error rate for manners depends on the number of hidden layers and number of units.
Tables 3 and 4 illustrate the performance of DNNs vs CNNs using AEER of the articulatory attribute detectors that show the lowest value of the AEER in terms of manner and place of articulations, respectively. It can be seen that CNNs consistently outperform DNNs for all different scenarios of tuned parameters, producing 10.42% relative error reduction for manner and 13.11% for place compared with the DBN-DNN models. This supports the claim that if the amount of training data is limited CNNs have better generalization ability than DNNs.

Table 5 demonstrates the performance of DNNs vs CNNs using AEER of all Arabic articulatory attribute detectors using the AEER. It is clearly shown that CNN (1024 × 3) significantly outperforms DNNs for all manner of articulation, with 9.14% for stops, 9.33% for fricatives, 9.80% for nasals, and 9.11% for approximants. Turning to the place of articulation, we can observe that lower relative error reduction scores can be delivered using a CNN (512 × 4), producing 13.20% for labial, 13.11% for palatal, 13.13% for velar, and 13.10% for glottal. This demonstrates the power of the attribute features in helping discriminating between phonemes. For instance, a set of each pair of phonemes can be considered which are similar in articulation such as /m/ and /n/, /q/and /k/, and /t/ and /d/ for development of the system to get better results through reducing the overlap between the confusable phoneme pairs. Overall, the manners of articulation behave better than the places of articulation.

3.3. Phone Lattice Rescoring Analysis. The detection-based ASR paradigm aims to outperform automatic speech recognition using only detectors of speech attributes as inputs. We accomplish this for the Arabic phone recognition task by rescoring phone lattices as explained. We compare the baseline with four systems using CNNs and DBN-DNN topologies with the ASR for all the speech attributes. Table 6 gives the recognition results in terms of accuracy of the baseline system and after rescoring with different attribute detectors. The baseline performance of 73.00% improves to 74.32% with CNN (512 × 3) and falls slightly with DBN-DNN (1024 × 8)-based detectors. It is encouraging that there are enhancements in the results for the detection-based ASR system compared to the baseline.

4. Conclusion

We have demonstrated in this work that we can achieve high accuracies for Arabic speech articulatory attribute detection using two models based on the cutting-edge deep neural networks. These are convolutional neural networks (CNNs) and feed-forward neural networks pretrained with the stack of restricted Boltzmann machines (DBN-DNN) [56]. Attribute detection for Arabic language should be considered as the single-label detection problem. All experiments have been conducted on the Arabic MGB dataset. Furthermore, phone lattice rescoring has proven useful in a detection-based ASR application. This opens up new opportunities to some old problems, such as speech recognition from a phoneme lattice and from phonological parsing [57]. A number of experiments have been conducted in order to find out the DNN model parameters for highest accuracy. Additionally, the effect of number of hidden fully connected layers and number of units was studied using average equal error rate (AEER) as an evaluation measure. It can be inferred that manner and place articulatory detectors trained with the CNN models work better than DBN-DNN models. The CNNs, compared with the DBN-DNNs, provide a significant error reduction for manner compared to place.

The lowest AEER has been demonstrated for manners by CNN with 2 hidden fully connected layers; each has 1024 units, and for places, the good performance was shown by CNN with 3 hidden layers and each of them had 512 units. The two CNN topologies have single convolutional and max-pooling layers. We can summarize that detectors trained with CNN models improved generalization ability compared to DBN-DNNs particularly if the amount of training data is limited. A good improvement is also observed on the phoneme classification task with excellent frame-level accuracy of 74.32% by using DNNs. This improved phoneme prediction accuracy, when integrated into the ASR system through a phone lattice rescoring framework, outcomes in improved recognition accuracy.

The next direction of the research is to apply multilabel learning approach as the inference framework above deep neural networks. Another investigation will be concerning the application of speech articulatory attributes in different fields of speech research.
Data Availability
The Multi-Genre Broadcast data used to support the findings of this study are openly available at https://www.mgb-challenge.org/index.html.

Conflicts of Interest
The author declares no conflicts of interest.

References


