Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.5, No.12, 2014
www.iiste.org

Automatic Diagnosis of Distortion Type of Arabic /r/ Phoneme Using Feed Forward Neural Network

Ayad T. Imam[1*] Jehad A. Alaraifi[2] Ibtisam Hussien J[3]
1.Faculty of Information Technology, Al-Isra University, Amman 11622, Jordan
2.Faculty of Rehabilitation, University of Jordan, P.O.Box 1544, Amman, 11941,Jordan
3.Council International Educational Exchange (CIEE), Princess Sumaya University for Technology, Amman , Jordan
* E-mail of the corresponding author: alzobaydi_ayad@iu.edu.jo

Abstract
The paper is not for recognizing normal formed speech but for distorted speech via examining the ability of feed forward Artificial Neural Networks (ANN) to recognize speech flaws. In this paper we take the Arabic /r/ phoneme distortion that is somewhat common among native speakers as a case study.To do this, r-Distype program is developed as a script written using Praat speech processing software tool. r-Distype program automatically develops a feed forward ANN that tests the spoken word (which includes /r/ phoneme) to detect any possible type of distortion. Multiple feed forward ANNs of different architectures have been developed and their achievements reported. Training data and testing data of the developed ANNs are sets of spoken Arabic words that contain /r/ phoneme in different positions so they cover all distortion types of Arabic /r/ phoneme. These sets of words were produced by different genders and different ages.The results obtained from developed ANNs were used to draw a conclusion about automating the detection of pronunciation problems in general.Such computerised system would be a good tool for diagnosing speech flaws and gives a great help in speech therapy. Also, the idea itself may open a new research subarea of speech recognition that is automatic speech therapy.
Keywords: Distortion, Arabic /r/ phoneme, articulation disorders, Artificial Neural Network, Praat

1. Introduction

Generally, sound distortion is one type of misarticulation that can be defined as a variation in the perceptual limitations of a phone used to pronounce a certain phoneme (Jakobson 1980, Ralph 1989).

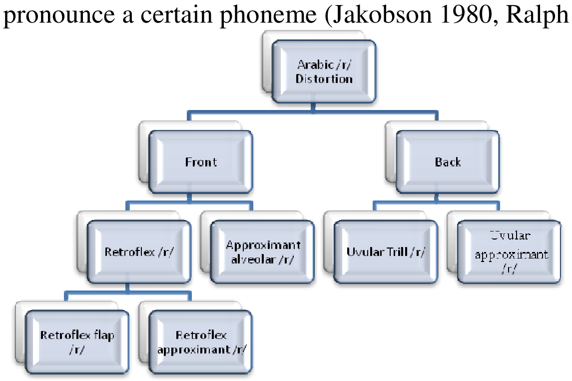

Figure 1: Arabic /r/ Phoneme Distortion Types

Classification of Arabic /r/ distortion is based on three criteria, namely: the nature of the tongue's movement, tongue's form and production's location of /r/ phoneme during its pronunciation by a patient. As illustrated in Figure 1, there are two major types of Arabic /r/ distortion, which are front distortion, where the produced sound stays within the gums, and back distortion, where the produced sound stays within the laurel area. Figure 2 shows the regions where these two types of distortion occur.

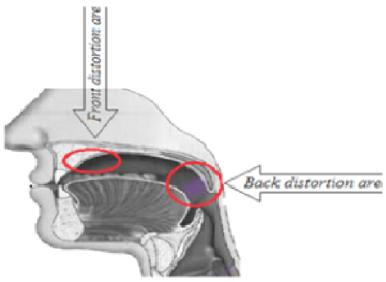

Figure 2: The regions of Front and Back distortion

Each of these two types is further classified into subtypes. The following details the distortion types

(Ibtisam et al., 2009):

1. Front Distortion is classified into two main forms, depending on the mechanism of movement at the tip of the tongue. These forms are:
   a. Retroflex /r/: the tip of the tongue is twisted to the back, and if the inner portion of the tongue hits the back of the gums it causes retroflex flap distortion. Otherwise, if the inner portion of the tongue does not beat the back of the gums (but only approaches it), and does not generate friction, it causes retroflex approximant distortion.
      i. Retroflex flap /r/: the patient twists partly the tongue back and inwardly hits the back of the bridge gum one blow. The severity of beatings and degree of the twisting of the tongue vary, as the same patient may vary in every attempt speech while he is trying to form the correct accents for this sound depending on his self-observation.
      ii. Retroflex approximant /r/: distorted phoneme /r/ differs from its predecessor in that it produces from approaching the soles tip of the tongue from the back of the bridge gum relatively closer leaving a void for air to pass without causing friction or disorder audible pneumatic. The patient is not accustomed to this voice to raise the tip of the tongue to an extent that leads to positional touch. With this voice – as is the case with Retroflex flap – the lips remain in the form of extroversion and withdraw the root of the tongue towards the back wall of the throat, leading to narrowing of the throat area.
   b. Approximant alveolar /r/: The tip of the tongue is raised towards the gums and remains suspended in the middle of the oral cavity without twisting backwards or hitting the gum. The void provides enough space for air to pass without causing friction or disorder audible pneumatic. The root of the tongue is pulled back, causing narrowing of the throat area, while the lips remain in the form of extroversion. This form of /r/ exists in British English pronunciation of that sound.
2. Back Distortion: The most important characteristic of Arabic /r/ phoneme is that it is a front phoneme produced in the gums area. In the back distortion the pronunciation of Arabic /r/ phoneme comes from the back of the oral cavity, namely in the soft palate and uvula areas of the cavity. Thus, the distorted phoneme /r/ would look like Arabic/ɣ̩/ phoneme (this phoneme is not found in modern English anymore, although it existed in Old English)
   a. Uvular Trill /r/: In this form, distorted Arabic /r/ phoneme is produced when the uvula quickly and sequentially hits the end of the ceiling of soft palate. This happens when the tongue nape rises towards the soft palate to form a relatively narrow place, and the uvula flexes forward towards the end of the ceiling of soft palate. Now, if air passes from the vacuum spot located between the back of the soft palate and the back of the tongue, it will push the soft palate to hit the end of the soft palate as quick strikes. These strikes continue as the flow of the air stream continues. The throat area will be narrow due to the withdrawal of the root of the tongue towards the back wall of the throat. The lips remain extroverted, and the tip of the tongue remains stable at the bottom of the oral cavity.
   b. Uvular approximant /r/: In this form of distorted /r/, the uvula flexes forward towards the soft palate, and the back of the tongue rises towards the uvula up to an appropriate degree that does not allow generation of repetition or friction. This approach of the back of the tongue forms a space allowing the air stream to pass through it smoothly. The front and tip of the tongue remain stable at the bottom of the oral cavity, and the tongue's root withdraws towards the back wall of the throat. Thus, the throat area narrows, and the lips remain extroverted.

The true diagnosis of Arabic /r/ phoneme distortion type helps in starting therapy. Currently, diagnosis is made by listening to a patient's pronunciation of certain words and the expert is able to detect the type of distortion since he/she is trained to do that. It is worth mentioning that sometimes the distortion is resulted from being /r/ phoneme is accompanied with other phonemes in a word. In this research we used Praat software to develop a computer system that is able to detect the type of Arabic /r/ phoneme distortion, which is given r-Distype name.

2. Related work

There have been many studies in the field of speech recognition using Artificial Neural Network (ANN). There is also ongoing research looking for enhanced ANN types to get even better results than those already presented. Although we couldn't find previous work that uses the automatic voice recognition technique to recognize the speech flaws, but we can mention here some of studies, which we get use of their ideas, among lots of published research work:

- Xiang et al. (1990) used a Multi-layer back-propagation neural network of 65 nodes of input layer, 40 nodes of first hidden layer, 30 nodes of second hidden layer and 17 nodes of output layer, to synthesis Chinese speech with high speech quality, where error rate was less than 5%.
- St George et al. (1997) used back-propagation neural network to recognize different voice patterns.
- Fackrell et al. (1999) suggested the use of cascade architecture of neural network to achieve significant

efficiency enhancement in the development of languages' prosody model used to predict the strength of phrase-boundary, the distinction of word and the interval of phoneme. Six main European languages were investigated using this model, which showed close recognition rate of between 91.0 % - 97.3 %.

- Ting (2002) was the first to use a multi-layer back-propagation neural network to develop a speech therapy system for clinical environment, which has an accuracy rate of 81.25% at syllable level and 67.71% at word level.
- Tan (2004) used a Multi-layer back-propagation neural network to develop a speaker independent/isolated word speech recognition system that gained 95% recognition accuracy.
- Macias-Guarasa et al. (2009) used a multilayer perceptron (MLP) neural network in developing a text-to-speech system. The best achievement of this NN based system was 33.53% decrease in error rate.
- Imam (2010) used Hierarchical Neural Network (HNN) to develop a speaker independent/isolated phoneme speech recognition system that achieved 69.2308% recognition accuracy.
- Abdul-Kadir et al. (2012) studied the performance of back propagation multi-layer feed-forward, which had the highest training recognition rate of 98.8% vs. cascade-forward neural networks, which had the highest training recognition rate of 92.9%. Both types had performance recognition rate of 92.9% for phonemes under study.

For dynamic data, the sample of data at any time instance is dependent on the patterns of data before and after its time instance. This feature is not taken into account by Back-Propagation algorithm, which is its defect. To overcome this defect, a new architecture of neural network is proposed, which is called the Time-Delay-Neural Network or TDNN (Waibel et al., 1989), where time-delayed inputs to the nodes are the key feature. The connection of each time delay to the node is achieved using its own weight that represents the input values in past time instances. Since the inputs to the TDNN are the delayed ones in time, TDNNs are also called the Input Delay Neural Networks. Delaying the input signal by a time unit and allowing NN to receive the original as well as the delayed signals will result in a simple TDNN. Certainly, a more complicated TDNN can be built by delaying the signal at different lengths. For example, if the input signal is of x bits delayed for y different lengths, then xy input units are required to encode the entire input. The arrival of new information will place it in nodes at one end and shift the previous information down a sequence of nodes much like the work of a shift register controlled by a clock (LiMin, 1994).

3. r-Distype system

"Distortions are more difficult to identify than accurate productions" (Kleinetal, 2011). The aim of this research is to develop r-Distype program, which is a computer system to detect distortion type of Arabic /r/ phoneme. It is a Neural Network based system implemented using Praat, which is freeware software working under various operating systems. Praat is considered a very helpful specialist tool for carrying out analysis and reconstruction of acoustic speech signals. It includes many standard and non-standard functions like articulatory synthesis, spectrographic analysis, and for developing neural networks (Boersma & Weenink, 2007).

The steps we followed to develop r-Distype program are illustrated in Figure 3. We started by preparing speech data, which was the most time consuming step because of the special care needed to reach an acceptable quality that helps r-Distype program to reach accurate diagnosing results.

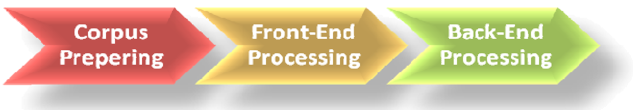

Figure 3: The Software Engineering of r-Distype System

In the following sections, a full description of the sequence of actions required to develop r-Distype computer system program to detect distortion type of Arabic /r/.

3.1 The used corpus

Since there is no standard corpus defined for distorted Arabic /r/ phonemes, the corpus used in this research was completely created by the authors. The corpus is a set of spoken Arabic words of patients who have difficulty uttering different distortion types of Arabic phoneme /r/ and saved as speech files that cover all possible types of distorted Arabic /r/ phoneme.

Words are selected to be the corpus elements, instead of phonemes, due to the fact that Arabic /r/ phoneme is possibly distorted due to the effect of the neighbouring phonemes, such as /k/, in the word /maɣrafah/ or/rux:m/.

As illustrated in Table 1, two criteria have been used to maintain the corpus in order that it includes all possible ways of generating distorted /r/ phoneme, namely:

1. The position of Arabic /r/ phoneme (beginning, middle and end) in the word, and
2. The length of the Arabic vowel (in its three types: breaking, closing and opening) accompanied by Arabic /r/phoneme, whether short vowel (Arabic harakāt: fatha, kasra, or damma) or long vowel (Arabic characters: /a:, e:, i:, or o:, u:/) (Kees, 1997).

Table 1: Arabic corpus used to in r-Distortion Detector System

Vowel Type / /r/ position	Short Breaking (Kasra)	Long Breaking /e: , i:/	Short Closing (Damma)	Long Closing /o:, u:/	Short Opening (Fatha)	Long Opening /a:/
Initial word position	/rina:d/	/ri:m/	/ruʔa/	/ru:ħ	/raħa:/	/ra:mi/
Medial word position	/mariħ/	/muri:ħ/	/quruħ/	/moru:dʒ/	/marah/	/sara:b/
Final word position	/sir/	/si:r/	/mur/	/du:r/	/sar/	/sa:r/

Arabic words that have repetitive phoneme /r/ (shaddah) are also included in this corpus via selecting a sample word for each Arabic short vowel (harakā). These words are: /marrah/, /murrun/ and /sirri/.These words were uttered by patients (an adult female and a male child). Each type of distortion is uttered by a patient who has this distortion type in his talk.

To carry out this research, recording sessions involving 70 patients of different ages and genders were conducted in a quiet room. While most of the phoneme data were uttered by patients, some distortion types of Arabic phoneme /r/ were uttered by therapy experts for verification and hence to enrich the training and testing data that would be used in our system. A laptop with built-in microphone, using Easy Hi-Q Recorder software with sampling rate of 16 kHz was used to produce 16-bit mono wav file format. Each uttered phoneme sample was cut by using Real Player trimmer software. For this research, a total of 50 files of distorted Arabic /r/ phoneme types were collected; five distorted Arabic phoneme /r/ types x 10 different speakers. The resulting speech files were grouped into distorted types; each type in a certain folder. The 50 distorted Arabic /r/ phoneme types were divided into 70% training set and 30% testing set for the designed neural network system.

3.2 Front end processing

In speech recognition applications, speech signal features are extracted first in a step called front end process. Temporal Analysis and spectral analysis are the two main techniques used to extract features from speech signal. There are different types of spectral analysis techniques, namely: Critical Band Filter Bank Analysis, Cepstral Analysis, Mel Cepstrum Analysis, Linear Predictive Code (LPC) and Perceptually Based Linear Predictive Analysis (PLP). Among these different types of speech signal features, LPC coefficients were used to represent the features of speech sound in this research. LPC spectral analysis encodes good quality speech at a low bit rate, which results in precise estimated parameters of speech signal. Principally, linear prediction is a mathematical approach that uses a linear function of previous samples to estimate discrete-time signal values. While linear prediction in system analysis is seen as a part of mathematical modelling or optimisation, it is termed as LPC in digital signal processing and classified as a subset of filter theory (Manish, 2003; Mporas, 2007; Sunitha, 2008; De Jong, 2009). The two decomposed functions results from LPC analysis of speech sounds are:

1. Filter function: which entails LPC coefficients, and assumes that the input signal has been filtered using a single variable-cross-section tube (vocal tract, for example).
2. Source function: This can either be the inverse of source signal and filtered by the filter function mentioned above, or a modified inverse of source signal and filtered by the filter function, which encompasses either white noise for unvoiced speech, or buzz (pulse train) for voiced speech.
3. In the LPC model, which is mathematically illustrated in (1), the speech signals are compressed to use as inputs for the neural network and go through a training process to attain targeted outputs (Rabiner et al., 2007).

$$s(n) = \alpha_1 s(n-1) + \alpha_2 s(n-2) + \ldots + \alpha_x s(n-x) \qquad (1)$$

Where s(n) is speech sample at time n, α1, α2 and αx are the assumed constants over the frame of speech analysis, minimising the error of mean-square over the sample of the whole speech, and x is the order of LPC. Unfortunately, while using LPC coding gives effective secure communication of low bandwidth sounds, the quality is poor and may be unacceptable when the decision of voiced/voiceless is error tolerant or fails to meet the moulds of the filter model like fricative or nasal sounds (Mporas, 2007; De Jong, 2009).

In Praat, there is LPC object type, which symbolises filter coefficients as a function of time. Praat steps for building Training Pattern (Features Extraction) are:

1. Read in a waveform file using: open→ read from file (Ctrl+O)

2. Extract the LPC: from the wave using one of the two following:
   a. Select sound file, then Formants & LPC -/ To LPC (burg)...
   b. Select sound file, then Analyse spectrum → To LPC (auto correlation)
3. Create Pattern object:
   a. Select LPC object resulting from step 3, Down To Matrix LPC,
   b. Select matrix object, Transpose Matrix,
   c. Select Transposed Matrix, Cast → to Pattern
4. Normalise the pattern:
   a. select pattern,
   b. modify→ formula (self / (biggest LPC value+1))
5. If there are more .wav files go to 1
   Otherwise combine all Pattern objects

3.3 Back-end processing

Feed forward neural networks are the only type that can be developed using Praat. The steps required to develop ANN for automatically detecting the distortion type of Arabic /r/ phoneme are:

1. Defining Categories object of the Pattern object: based on the number of distortion types of Arabic /r/ phoneme, which was five, the number of categories objects was five. Each type was related to occurrences set of certain Arabic /r/ phoneme patterns that may be produced by different patients. Of course these data occurrences are selected from the training set, not the testing set. The Praat commands used to implement this step are:
   a. Select pattern → To categories
   b. Check Categories → Edit
2. Developing the FFNet: Multiple feed forward ANNs of different architectures were developed. The input layer of all these ANNs was of 16 nodes size, due to the number of LPC parameters. Also, in all these ANNs, the last layer, which was the output layer, was size 21 nodes – the same number of the sample words used to detect distorted Arabic /r/ phoneme. Thus, the difference between these ANNs is in the hidden layer(s). The number of hidden layers and the size of each hidden layer were chosen based on trial and error principle. Praat command performing this step is: Select Pattern & Categories → To FFNet.
3. Training the ANN using the extracted LPC: The adaptation of the weights is made using the supervised learning algorithm embedded in FFNet Praat's command. Supervision means that the neural net needs to know to what category the input pattern belongs, which means that both input pattern and classifications should be submitted to the network under training. Due to the method by which the weights of the network are being modified, the term Backpropagation network is also frequently used to describe this type of neural network, where the differences between the current actual outputs and the predefined classification are propagated back from the upper layer down to the lower layers so it could be used to modify connection weights at network's layers. This procedure is performed once for each pair of pattern and its classification data set termed as epoch of learning. Performing many epochs aims either to make the neural net able to recall these pattern-category pairs or, in case the learning phase has terminated, the network gain generalisation ability so it correctly classifies any mysterious input pattern (not included in the training set). The former is very interesting because real-life data mostly contain noise. In our experiment, the number of distorted Arabic /r/ phoneme samples available determines the number of epochs, which was between 2000 and 20000 epochs. The training continued until it reached the specified epochs or an error of 1x 10-6 was reached. In Praat, performing learning requires the selection of three different objects together, the classifier that is an FFNet, the input patterns and the patterns' categories implemented as Praat command: Select FFNet and Pattern and Categories→Learn. Figure 4 shows the training phase of one of these ANNs.

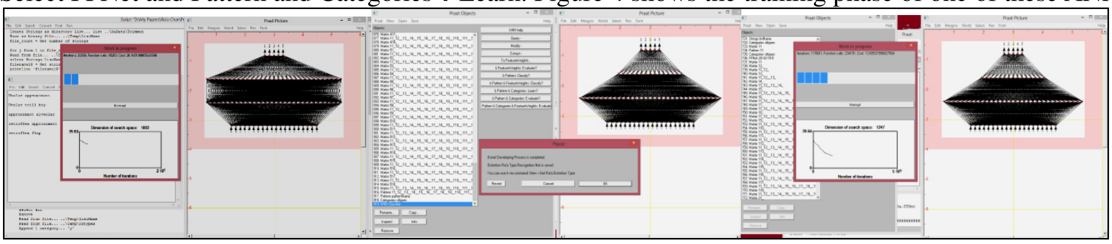

Figure 4: Training window

4. Performance

Testing the developed ANNs was carried out using a different set of data. The steps required to prepare the data to be tested by ANN for automatically detecting the distortion type of Arabic /r/ phoneme (classification phase) are:

Step I. Building the Testing Pattern (Features Extraction):

1. Open spoken word file (.wav):
   a. open→ read from file (Ctrl+O)
2. Extract the LPC:
   a. Select sound file,
   b. Analyze spectrum → To LPC (auto correlation)
3. Create Pattern object:
   a. Select LPC object, Down to Matrix LPC
   b. Select matrix object, Transpose matrix
   c. Select Transposed matrix, cast → to Pattern
4. Normalise the pattern :
   a. Select pattern, modify→ formula

Step II. Classify the Pattern object using FFNet:

1. Select pattern& FFNet→ To categories

Step III. Check Categories (read result) → Edit

A specified performance function is used to measure the performance. After training the network, we test it using the 10 samples of each class randomly selected from the distorted Arabic /r/ phoneme samples separated for testing. A confusion matrix, for each experiment, was used to present the percentage of recognition accuracy as illustrated in Table 2, which reported the developed ANNs and their achievements. Same training data were used for training these ANNs, and the same testing data were used to test the ANNs. It is important to make sure that the results are gained under the same circumstances and hence make the comparison and conclusion more reliable

Table 2: Developed ANNs and their Achievements (Error Rate)

ANN's Architecture (I-H_1-H_2-O)	*Leaning parameters*			*Testing using Training Data (error rate)*	*Testing using Testing Data (error rate)*
	Maximum Number of Epochs	*Tolerance of Minimiser*	*Cost Function*		
16-31-31-5	*200,000*	*1e-15*	*mse*	*16 out of 90 (17%)*	*27 out of 90 (30%)*
16-31-31-5	*500,000*	*1e-15*	*mse*	*15 out of 90 (16%)*	*29 out of 90 (32%)*
16-31-31-5	*200,000*	*1e-15*	*mse*	*16 out of 90 (17%)*	*23 out of 90 (25%)*
16-32-15-5	*500,000*	*1e-15*	*mse*	*4 out of 60 (6 %)*	*31 out of 90 (34%)*
16-20-40-5	*200,000*	*1e-15*	*mse*	*4 out of 60 (6 %)*	*42 out of 90 (46%)*
16-32-10-5	*500,000*	*1e-15*	*mse*	*5 out of 60 (7 %)*	*28 out of 90 (31%)*
16-36-14-5	*200,000*	*1e-15*	*mse*	*5 out of 60 (7 %)*	*25 out of 90 (27%)*
16-32-16-5	*500,000*	*1e-15*	*mse*	*6 out of 60 (10 %)*	*33 out of 90 (36%)*
16-32-14-5	*200,000*	*1e-15*	*mse*	*6 out of 60 (10 %)*	*27 out of 90 (30%)*
16-36-10-5	*500,000*	*1e-15*	*mse*	*6 out of 60 (10 %)*	*29 out of 90 (32%)*
16-32-32-5	*200,000*	*1e-15*	*mse*	*7 out of 60 (11 %)*	*34 out of 90 (38%)*
16-30-30-5	*500,000*	*1e-15*	*mse*	*7 out of 60 (11 %)*	*32 out of 90 (35%)*
16-30-16-5	*200,000*	*1e-15*	*mse*	*7 out of 60 (11 %)*	*41 out of 90 (45%)*
16-30-10-5	*200,000*	*1e-15*	*mse*	*7 out of 60 (11 %)*	*24 out of 90 (26%)*

mse: minimum squared error cost function ; mce: minimum cross entropy cost function

5. Discussion

In this paper, we have described a script written in Praat that automatically develops feed forward ANNs to detect the distorted type of Arabic /r/ phoneme. The resulting ANN tests sound file of a spoken word is a recorded word uttered by a patient. Different multiple architectures of feed forward ANN have been examined and their results reported.

As shown in the Table 2, it is almost impossible to get a free error ANN that is able to detect the distortion type of Arabic phoneme /r/. This is normal due to the recorded achievement of ANN in speech recognition applications in that successful speech recognition application is a complex system which encompasses elements other than ANN. In addition to feed forward object, Pratt has kNN (nearest neighbourhood) classifier object. We tried to include it in a recognition system encompassing ANN and kNN, even though this mixture did not give promising results. Also, we tried to change the number of LPC (the input parameters), but we noted only a small difference which is not stable for all test cases, i.e. could enhance the performance in a certain group, but badly affect the other. Changing the learning phase parameters (maximum number of epochs, tolerance of minimiser and cost function either using minimum squared error cost function or minimum cross entropy cost function) leads to some changes, although this cannot be generalised to all cases.

As the aim of this program is to diagnose the distortion type of Arabic /r/ phoneme, which requires great accuracy, the uses of speech processing technique are not suitable for such application, unless one can give 100% error free and completely independent human assistant in its work. Such criteria, we believe, are not achievable at least with current speech processing techniques and tools. We recommend also testing the complex speech processing system that uses other tools and techniques like hierarchical neural network, or Hidden Markov Model (HMM).

References

Abdul-Kadir, N. A., Sudirman, R., Mahmood, N. H., & Ahmad, A.H. (2012). Applications of cascade-forward neural networks for nasal, lateral and trill Arabic phonemes. Information Science and Digital Content Technology (ICIDT), 2012, 8th International Conference, Korea (South)

Boersma, P., & Weenink, D. (2007). Praat (Version 4.5.25) [Software]. Latest version available for download from www.praat.org

De Jong, N. H., & Wempe, T. (2009). Praat script to detect syllable nuclei and measure speech rate automatically. Behavior Research Methods, volume 41; issue 2, pp. 385–390, doi:10.3758/BRM.41.2.385

Fackrell, J. W. A., Vereecken H., Martens J-P, and Coile B. V. (1999) Multilingual prosody modelling using cascades regression trees and neural networks. Sixth European Conference on Speech Communication and Technology EUROSPEECH 1999, Budapest, Hungary, pp. 1835–1838

Ibtisam, H. J., Nawwaf, H., & Al-Arayfi, J. (2009). The articulation disorders in producing the Arabic /r/- descriptive and analytical study. Dirassat: Human and Social Sciences

Imam, A. T. (2010). Relative-fuzzy: A novel approach for handling complex ambiguity for software engineering of data mining models. PhD thesis, De Montfort University

Jakobson, R. (1980). Structure of language and its mathematical aspects: Proceedings of Symposia in Applied Mathematica. AMS Bookstore

Kasabov, N. K. (1996). Foundations of neural networks, fuzzy systems and knowledge engineering. Cambridge, London: The MIT Press

Kees, V. (1997). The Arabic language. Edinburgh University Press

Klein, H.B., Grigos, M.I., & McAllister-Byun, T. (2011). Acoustic properties of children's /r/ productions: Impact on listeners' perceptions. American Speech Language Association Annual Convention, San Diego, CA

LiMin, Fu. (1994). Neural networks in computer intelligence. McGraw-Hill International Editions, Computer Science Series.

Macias-Guarasa, J., Montero J. M., Ferreiros J., Cordoba, R., San-Segundo, R., Gutierrez-Arriola , J., D'Haro, L. F., Fernandez, F., Barra, R., & Pardo, J. M. (2009). Novel applications of neural networks in speech technology systems: Search space reduction and prosodic modelling. Intelligent Automation and Soft Computing, 15(4), pp.631–646

Manish P. K. (2003). Feature extraction for speech recognition, M. Tech. Credit Seminar Report, Electronic Systems Group, EE. Dept., IIT Bombay /India

Mporas, I., Ganchev, T., Siafarikas, M., Fakotakis, N. (2007). Comparison of speech features on the speech recognition task. Journal of Computer Science, 3(8), 608–616

Ohde N. Ralph, McCarver, M.E. and Sharf, D. J. (1989). Perceptual and acoustic characteristics of distorted /r/. Journal of Acoustic Society of America, Vol. 86

Rabiner, L. R., & Schafer, R. W. (2007). Introduction to digital speech processing, foundations and trends in signal processing. Vol. 1 Issue 1-2. The Essence of Knowledge, USA

St George, B. A., Wooten, E. C., & Sellami, L. (1999). Speech coding and phoneme classification using Matlab and neural networks. Information Sciences; Elsevier publishing company, 90, Pp.109–119

Sunitha, K.V.N., & Kalyani, N. (2008). Speech recognition tools in local language for Kissan Help. 2nd International Conference on Computing for National Development, 8th – 9th Feb. 2008, Hyderabad/India

Tan C. L., & Jantan, A. (2004). Digit recognition using neural networks. Malaysian Journal of Computer Science, Vol. 17; Issue 2, pp. 40–54

Tebelskis, J. (1995). Speech recognition using neural networks. PhD thesis, Carnegie Mellon University

Ting, H. N. (2002). Speech analysis and classification using neural networks for computer-based Malay speech therapy. Master's thesis, UTM Malaysia

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. (1989). Phoneme recognition using time-delay neural network. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.37; Issue 3, pp.328–339

Xiang, Z. J., & Bi, G. (1990). A neural network model for Chinese speech synthesis. IEEE International Symposium. Circuits and Systems, IEEE Explore, pp.1859–1862

Authors

Ayad T. Imam received his Ph.D degree in computer science from De Montfort University, Leicester, UK in 2010. Currently, Dr. Ayad is an Assistant Prof. at Al-Isra University / Amman / Jordan. Dr. Ayad has number of published papers in various computer science and Software Engineering topics. Dr. Ayad is a reviewer in a number of journals and conferences of Computer and Information related areas.

Jehad A. Alaraifi received her first MA in Speech Language Pathology & Communication Disorders from the University of Jordan, Jordan in 1994, and a second M.A. in Speech Language Pathology & Audiology from San Jose' University, California, USA in 2001. She's certified from the American Speech Language and Hearing Association (ASHA) in 2001. Currently, she's an instructor at the University of Jordan. She has number of published papers in speech and phonetic related topics

Ibtisam Hussien J. received her Ph.D degree in Arabic phonetics from University of Jordan, Jordan, in 2003. Currently, she is an Associate Prof. at Council International Educational Exchange (CIEE), Princess Sumaya University for Technology, Amman, Jordan. Dr. Ibtisam has number of published papers in different Arabic language topics, and she is a reviewer in number of conference and journals related with Arabic language and phonetics