EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

A Direct Speech-to-Speech Neural Network Methodology for Spanish-English Translation*

Manuel Quintana[1], Miguel Bernal[1,*]

[1]Dept. of Electrical and Electronics Engineering, Sonora Institute of Technology, 5 de Febrero 818 Sur, C.P. 85000, Cd. Obregon, Mexico.

Abstract

In this work, a novel direct speech-to-speech methodology for translation is proposed; it is based on an LSTM neural network structure which has proven useful for translation in the classical way, i.e., the one consisting of three stages: speech-to-text conversion, text-to-text translation, and text-to-speech synthesis. In contrast with traditional approaches, the one in this work belongs to the recently appeared idea of direct translation without text representation, as this sort of training better corresponds to the way oral language learning takes place in humans. As a proof of concept digits are translated from an audio source in Spanish and pronounced as an audio signal in English. Advantages and disadvantages of the proposal when compared with traditional methodologies are discussed.

Received on 29 February 2020; accepted on 06 April 2020; published on 24 April 2020

Keywords: Speech processing, Neural networks, Pattern Recognition.

doi:10.4108/eai.13-7-2018.164109

1. Introduction

Motivation: This work pursues the proposal in [1] which intended to be a first step towards the investigation of voice-to-voice methodologies for translation. Since its appearance, very few works have been developed in the literature that also considered direct speech-to-speech translation as a plausible methodology in contrast with the traditional approach, i.e.: they intended to use raw audio signals or slight audio processing while disregarding the use of dictionaries or grammar-based data. Some of them have been more or less successful because they have employed attention mechanisms based on phonemes [2] or databases known as codebooks which, although phonetic, constitute a sort of dictionary [3]. None of them is purely audio-based.

This work explores this possibility still on a preliminary stage which only addresses the problem of translating a limited set of words, more specifically, digits pronounced in Spanish from a variety of voices and translated into English in another non-standard voice, i.e., the result is not a "synthetic" one, but a human-based audio which captures some of the emotions in the utterance. The full methodology is supposed to be able to deduce grammar in the same way as humans do: by supervised learning and implicit adaptation, much more before written language appear.

Problem statement: Develop a direct speech-to-speech methodology which is able to translate digits from and to audio signals without any text representation and contrasting the results against traditional translation methodologies.

State of the art: This work belongs to the field of natural language processing (NLP), which comprises a variety of areas, depending on whereas the focus is made on signal processing [4], human-machine interface [5], or artificial intelligence [6]. Two extreme approaches can be identified: on the one hand, statistical tools usually applied when the emphasis is made on signal treatment without any higher-level interpretation; on the other hand, purely conceptual handling of syntactic units and semantics, which disregards implementation details in order to better relate with linguistic communities [7]. An in-between approach is that based on deep neural networks as these biologically-inspired models have been long ago

*This work has been supported by the CONACYT scholarship 725949 and the ITSON PFCE 2019.
*Corresponding author. Email: miguel.bernal@itson.edu.mx

related with statistical analysis [8], while being capable of subsuming large learning tasks in more conceptual units [9].

In the last decades, neural networks have been well established as a useful tool to perform learning tasks by embedding highly complex, nonlinear, and parallel information into weights that mimic brain synapsis [10]. Three main landmarks can be distinguished in their evolution: a modest origin when the perceptron was nothing else than a linear filter [11], the success of the multilayer perceptron and the backpropagation algorithm for heuristic training [12], and, finally, the surge of deep learning techniques via convolutional, recurrent, and recursive neural networks [13]. The latter framework have already made a significant advance in computer vision and pattern recognition by yielding shallow models such as support vector machines (SVM) to neural networks trained with dense vector representations [14]. A key factor on the success of deep learning techniques is the understanding of the importance of regularization and its influence on the generalization capabilities of the setup [15].

Within the field of NLP, neural networks in all their varieties have played a central role [16]: convolutional neural networks (CNNs) have been suitable for feature extraction [17] and data mining [18]; recurrent neural networks (RNN) can be found in word- and sentence-level classification [19, 20]; recursive neural networks (VNNs) have been successfully employed for parsing [21]. In all these tasks, multiple processing layers on which different deep learning algorithms can be applied, have been used to learn hierarchical data. A subclass of the RNNs is that of the long short term memory (LSTM) networks, which have proved to be enjoy invariance with respect to time, a quality particularly important for audio signals and word detection, i.e., sequences of data whose characteristic are not time-attached.

An important task within the NLP area is that of automated translation, which usually involves feature analysis (based on neural networks as well as Markov chains, for which software tools such as HTK or SPHINX are available [22]), unit detection (such as phonemes, [23]), syntactic analysis (for example, for word validation [24]), and proper translation via a looking-up procedure in a database. Remarkably, to the best of our knowledge, voice-to-voice translation by direct learning is not present in the literature with few exceptions for classification [25–28] and some others for translation: [2], which consists in a neural network composed by an 8-layer LSTM, an attention model for phonemes recognition and a voice synthesis module and [3] which uses a codebook of phonemes for translation with three stages, firstly there are some BiLSTM-layers to encoding the input data into phonemes, a convolutional middle layers and a final LSTM-layers for decoding the phonemes into an audio signal.

Contribution: In contrast with the related work [1], the current one uses an LSTM neural network subject to regularization instead of a multilayer perceptron, it employs a full spectrogram which is pre-processed as a sequence, it does not perform identification, but an implicit translation whose resulting sequence is synthesized in a third-party voice. Test signals which do not belong to the training set are successfully translated as shown in: `https://github.com/cope-quintana/Spanish_English_Translation_Dataset_and_Results`. Thus, the main contribution lies on a purely audio-based translation with no text representations involved as a proof of concept towards automated translation whose grammar is inferred instead of imposed.

Organization: This work is organized as follows: section 2 makes an elementary summary of what we have to know about audio signals and pre-processing while describing the neural network, learning algorithm, and regularization methods employed to perform the learning tasks; section 3 details the experimental setup of the proposed approach for automated translation as well as the behaviour of the training and test phases; section 4 discusses the results in contrast with traditional text-based translations; conclusions are gathered in 5.

2. Preliminaries

In this section we discuss the way audio signals are obtained, normalized, and the sort of neural network we are employing. It should be kept in mind that every choice has been made as simple as possible as to guarantee reproducibility of the results here provided.

2.1. Audio signals

When recorded via microphones or other electronic devices, audio signals are usually displayed as a time function resulting from the vibrations of a membrane; its intensity (see Fig. 1, which corresponds to the Spanish word for zero) roughly corresponds to the volume and its variations have frequency content that can be further analyzed via a spectrogram such as that in Fig. 2. A spectrogram combines information in the temporal and frequency domains: its horizontal axis represents time while the vertical one corresponds to frequency; the latter is usually split in bands of 1000 Hz each, and ranges from 0 to 8000 Hz, which is the highest frequency to be found in human voice [4], though that in Fig. 2 is limited to 5000 Hz. The darker an area in the spectrogram is, the higher the energy of those frequencies at that time; when these peaks are above a certain level they are identified as formants of the signal, which play an important role in phonetic analysis [29] (they are identified by bullets in

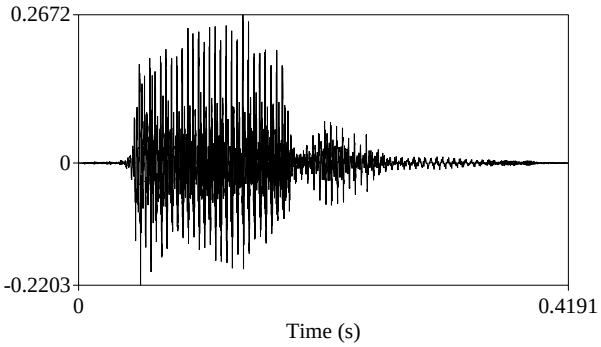

Figure 1. An audio signal corresponding to the Spanish audio "cero" (0).

Fig. 2). Spectrograms vary depending on the window and hop employed for performing the discrete-time Fourier transformation; the one shown in Fig. 2 employs 2048 samples where 44100 represent 1 second; the hop is 512 samples.

As with many other works on audio analysis, the tool we are employing to capture the audio signals is Praat, a computer program from the University of Amsterdam for analysis and manipulation of speech [30]; the tool is well beyond the needs of this report as spectral, pitch, formant, and intensity analysis, among others, can be performed with it. Tutorials are available for phonetics, learning, and statistics [31].

Clearly, a deep understanding of audio analysis is out of the scope of this work as we are only concerned with capturing audio signals in any form so that they can be pre-processed and fed to a neural network for pattern recognition. It is customary to use a grid of the spectrogram to vectorize its lighter/darker contents for neural network processing [32, 33]; this approach is closely linked to image processing as proven in [34]. In this work we employ the spectrogram of the audio

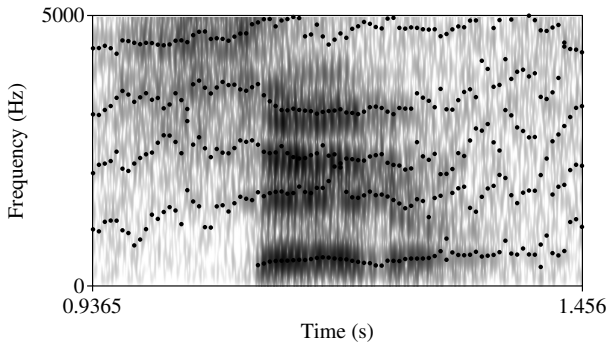

Figure 2. A spectrogram: time VS frequencies. Formants are highlighted as bullets.

signals as the input to the network, in contrast to [1] where the intensity signals were employed; a similar approach can be found in [2]. As with some image processing tasks, the spectrogram graphically depicted is converted into a sequence of data representing intensity; information concerning the frequency bands or the time of appearance of a particular part of the signal are thus subsumed in the sequence within approximately the same places.

2.2. Pre-processing of the audio signals

As digits may be spoken in a variety of volumes and durations, they require a pre-processing in order to make any sample signal have the same wide and maximum height. This a reasonable choice which may not be advisable for larger expressions as whole statements may considerably vary in size. A number of methods is available for neural-network-oriented pre-processing of audio signals, e.g.: in [35] Mel frequency cepstrum coefficient analysis is employed, [36] also uses Mel analysis adding gaussian noise, [37] preprocess music with binaural representations and harmonic-percussive source separation, and [38] mixes time-frequency representation, logarithmic magnitud compression, and frequency weighting and scaling, among others.

Since this work intends to keep the structures and algorithms as simple as possible to illustrate our hypothesis, we will perform audio processing in five steps: an audio trimmer that eliminates silences from the signal (below a certain intensity), a low-pass filter of the sample which eliminates high-frequency noise (very common in this applications), an spectrogram converter that actually generates the corresponding spectrogram for the already treated audio sample, a series of straightforward operations of shrinking/expansion of the spectrograms as to fit uniform horizontal size corresponding to the average width of all the samples, and finally a normalization of the intensity of the spectrograms data which roughly performs the same shrinking/expansion in the 3D values of the spectrogram.

Some formulas to perform the foregoing changes are MATLAB^{TM} functions *filter* for filtering, *spectrogram* for spectrogram generation, and *interp1* for horizontal normalization, while the other functions like *audio trimmer and intensity normalization* were customized functions (implemented in MATLAB^{TM}). The intensity adjustment is given afterwards by a simple formula

$$in_{new} = \frac{in_{current} \times maxIn_{global}}{maxIn_{current}}, \qquad (1)$$

where $in_{current}$ is each data of the current spectrogram, $maxIn_{current}$ it the maximum intensity of the whole current sample, $maxIn_{global}$ is the maximum intensity

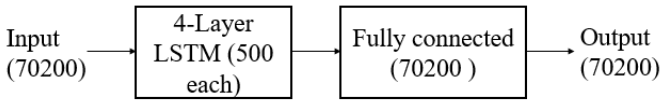

Figure 3. The neural network architecture: 4-Layer LSTM with fully connected layer.

of the whole set of samples and in_{new} is the new intensity of the current spectrogram. After these changes, every spectrogram sample signal will be scaled within a similar range of intensities where all the samples have the same maximum intensity in their corresponding maximum point. Finally in the horizontal normalization step all the samples where normalized into matrices of dimensions 300×234. After this normalization the resulting matrix was converted into a vector using MATLAB^{TM} function *reshape* into a vector of 70200 units. This is the vector that will be fed for neural network training.

2.3. The neural network and the learning algorithm

A network with 70200-input, 4-layer LSTM , 70200-output fully connected layer will be employed for this proof of concept; the input corresponds to the sequence of pre-processed spectrogram samples coming from different sources in Spanish while the output, being a processed spectrogram too, comes from a source in English. The output must be transformed into an spectrogram matrix for audio reconstruction and audio results validation. Fig. 3 shows an overview of the network architecture, the number of neurons of each LSTM layer is 500 with a dropout regularization applied [13].

The training algorithm was implemented with MATLAB^{TM} function *trainNetwork* with the following choices:

1. Initialization: The network consists of a 70200-input layer, 4-layer LSTM with 500 neurons each, followed by a fully connected 70200-output layer.

2. Training samples: Each sample has the standardized pre-processed spectrogram; it consists of 70200 points whose values represent the spectrogram transformed into a sequence. At each epoch all the samples are presented in random order.

3. Validation: Each 5 epochs the network is validated with a validation set that is not part of the training set.

4. Regularization: A dropout rate of 0.3 has been employed in each LSTM layer, which means that 30% of the synaptic weights are randomly reset after each epoch to avoid overtraining or saturation. An early-stopping-like algorithm has also been employed to prevent overfitting: based on a well-known algorithm, it has been determined how long to train by using the validation error, then retraining up to the detected appropriate level.

5. Optimizer: The stochastic optimizer Adam [39] was used to train the network in 500 epochs with a base learning rate $\alpha = 0.001$. The optimizer algorithm is based on adaptive estimates of lower-order moments for first-order gradient optimization; this makes it suitable for non-stationary objectives as those in this work.

3. Results on direct speech-to-speech translation

As stated before, the normalized and vectorized spectrogram of each sample is employed for training. Normalization guarantees they all consist in 70200 points and the maximum intensity is the same; Fig. 4 shows a normalized spectrogram just before vectorization: it is a data matrix of dimensions 300×234. The input signals belonging to each digit share some common characteristics, though some of them (specifically, number 8 whose pronunciation has a widely varying phoneme range in Northwest Mexico) are more inclined to exhibit important variations than others (for instance, number 1 is nearly the same for every speaker); the LSTM network is capable of avoiding these variations due to their specialization on sequences (i.e., detecting patterns despite their time variance).

For this experiment we employed audio samples recorded from 6 participants (5 men and 1 women). Each participant recorded 30 samples in Spanish and 30 translations in English, 3 samples per digit. The samples were recorded with Audacity^{TM}, all the samples were recorded using the same software and microphone to guarantee uniformity. Table 1 briefs the information about the experimental data. All the preprocessing operations were implemented in MATLAB^{TM}; filtering and silence removal were the first

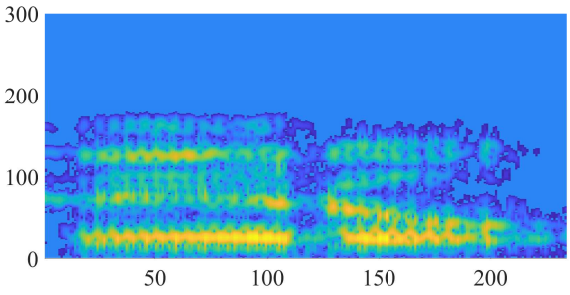

Figure 4. Normalized spectrogram corresponding to the Spanish audio "cero" (0) before vectorization.

Table 1. Experimental and participants data.

Experimental Data	
Participants	6 (5M - 1W)
Audio samples per digit	18/language
Audio samples per participant	30/language
Training samples	150/language
Validation samples	30/language
Total samples	180/language

steps, then spectrogram analysis and normalization, after which the resulting spectrograms were vectorized to feed the network. The trained neural network produced a vectorized spectrogram as an output for each test sample; these outputs were reshaped into matrices, and then, output audios were obtained using the *Fast Griffin-Lim Algorithm* [40] which is an spectrogram decoder. Fig. 5 shows a flow map of these steps.

The training settings employed for the network were selected in order to keep it as simple as possible; MATLAB^{TM} was employed for the implementation of the network due to its *DeepLearning* toolbox that allow us to easily implement a variety of neural networks in a flexible way. The network consists in a input sequence layer of 70200, 4-layer LSTM of 500 neurons each, followed by a fully connected layer, we implemented a dropout of 0.3 on each LSTM layer. The optimizer used for the training was *Adam* with a learning factor of 0.001; the networks were trained during 300 epochs with a validation each 5 epochs.

As Fig. 6 shows, the training error of the neural network drops almost continually with the exception of a series of peaks. These correspond to the regularization steps where 30% of the neurons are dropped out, a technique which is known to avoid overfitting. The plot shows that, effectively, after these random eliminations take place the error keeps diminishing as expected. Recall that lower training error can be achieved from the point of view of optimization, though this is undesirable as the generalization capabilities

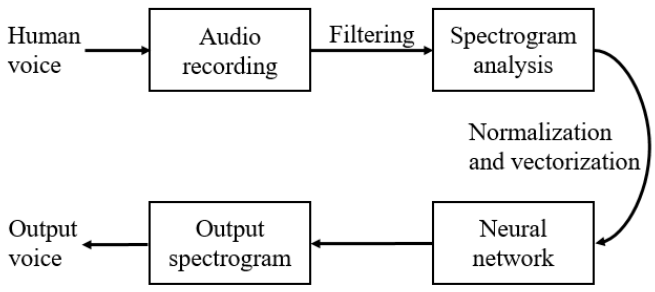

Figure 5. Flow map of the experiment.

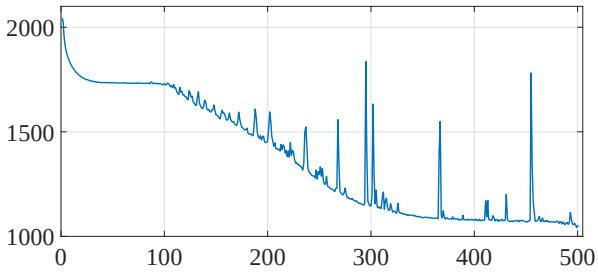

Figure 6. Neural network training error.

of the network will be affected: this is clear from comparison of the training error in Fig. 6 with that of the validation error in Fig. 7, which is examined every 5 epochs, thus resulting in a 100-sample scale instead of a 500 one: while the former keeps diminishing, the latter rises. Importantly, the plot in Fig. 6 covers the whole first training phase as to be compared with the validation error in Fig. 7 for the early-stopping-like algorithm: it retrains the network up to 300 as this is approximately the number of epochs that have passed for the validation error to rise up again (300/5 = 60, which coincides with the increasing of the validation error in Fig. 7. Note that there is a temporary increase in the validation error right before 50 (corresponding to 250 epochs): the early-stopping-like algorithm did not took this as the stop mark because it went down before completing the decision window which is about 10 (correspondingly, 50 epochs).

In `https://github.com/cope-quintana/Spanish_English_Translation_Dataset_and_Results` test signals which do not belong to the training set are translated. Validation of direct speech-to-speech translations is a challenging task as there is no standard way to establish the "right" pronunciation. A way to overcome this difficulty is to use an automatic speech recognition (ASR) algorithm which produces a text output from an audio source [2, 3]; nevertheless, this approach is obviously questionable as evaluation may include the ASR algorithm itself, i.e., if the ASR algorithm performs poor it is hard to establish whether it has been because of the ASR or because of a poor audio source. Naturally, researchers have also relied on users to validate audio translations [2]; again, objections can be raised as this affects the objectivity of the test with many users providing the "missing" elements by interpretation. Native English-spoken users may check the translations in the webpage above. Since this is only a very short proof of concept, no better validation has been made except for the human assessment shown in Table 3 under the conditions in Table 2.

Table 2. Human assessment settings.

Evaluation settings	
Participants	10 (8M - 2W)
Audio samples per digit	3
Total samples	30/language
Evaluation range	{0, 1, ..., 10}

Table 3. Human assessment of results.

Evaluated digit	Average evaluation
Zero	7.20 (1.72)
One	9.03 (0.29)
Two	8.47 (0.52)
Three	7.70 (0.80)
Four	5.53 (1.25)
Five	7.53 (1.28)
Six	6.63 (1.57)
Seven	7.57 (1.14)
Eight	7.73 (0.96)
Nine	7.90 (1.04)

4. Discussion

The individuals making the assessment are all Spanish-spoken with a TOEFL certificate of a minimum of 500 points; their appreciation ranges from 0 (which means a digit has not been identified) to 10 (implying that the digit has been distinctively clearly identified); each individual heard 3 translations of each digit; each time she/he heard a digit, she/he provided a mark (0-10). The average per person per digit has been obtained for each individual; then the final average per digit in Table 3 has been obtained by averaging the 10 individuals per digit. Without distinguishing digits, the average is 7.52, which means a rate of 75% of recognition is achieved. This is, nevertheless, not uniform: in 4 cases the individual was unable to identify the digit; 4 is the poorest identified number as the utterances is barely recognized by the individuals (2 of them were actually unable to identify it). Other sounds (with less complex phonetics) such as the digit 1 are very well evaluated.

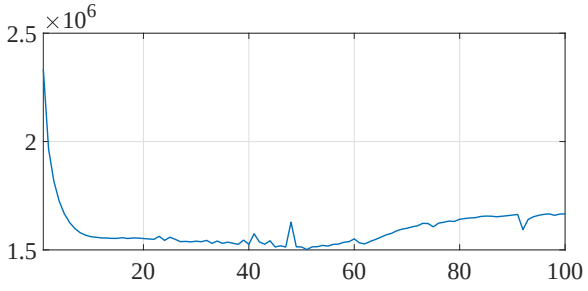

Figure 7. Neural network validation error.

Though on the good side, these results may seem mixed. Nevertheless, it should be taken into account that no text-to-text translation has been used in the process, which makes the task notably more complex. Resources for achieving this performance were scarce: 25 hours of training and 15 hours of re-training (after the early-stopping-like comparison and rerun), within a MATLAB 2019a version running on an iCore 7 3.40GHz terminal with Windows 7 OS and 8 GB RAM; this makes us consider that the methodology has a good potential provided bigger resources are put forward. Increasing the processing capabilities will not only speed up the training process, but more importantly will help us achieving the next goals towards a full speech-to-speech translation without any dictionary or text representation in between. It is our contention that grammar reordering or sense ambiguity can be overcome automatically with the proper training when longer statements are subject to translation, i.e., the neural network and the training algorithm should be able to mimic the way a human being learns grammar: by trial-and-error as well as implicit inference of the underlying structures.

5. Conclusions

A novel direct speech-to-speech methodology based on an LSTM neural network has been proposed for Spanish-English translation. The proof of concept has been made on a digit translation which has been implemented and assessed by human speakers. Thanks to the results (available on a public webpage), it has been established that, in contrast with traditional approaches, direct translation without text representation is possible and might better correspond to the way languages are learned by humans.

References

[1] Quintana M, Bernal M. A Comparison of Speech-to-Speech Neural Network Methodologies for Digit Pronunciation. In: Proc. 3rd EAI International Conference on Computer Science and Engineering and Health Services. Mexico City, Mexico; 2019. p. 01–12.

[2] Jia Y, Weiss RJ, Biadsy F, Macherey W, Johnson M, Chen Z, et al. Direct speech-to-speech translation with a sequence-to-sequence model. arXiv preprint arXiv:190406037. 2019;.

[3] Tjandra A, Sakti S, Nakamura S. Speech-to-speech Translation between Untranscribed Unknown Languages. arXiv preprint arXiv:191000795. 2019;.

[4] Lerch A. An introduction to audio content analysis: Applications in signal processing and music informatics. Wiley-IEEE Press; 2012.

[5] Poleshchuk O, Komarov E. Expert fuzzy information processing. vol. 268. Springer Science & Business Media; 2011.

[6] Kacprzyk J, Zadrozny S. Computing with words is an implementable paradigm: fuzzy queries, linguistic

data summaries, and natural-language generation. IEEE Transactions on Fuzzy Systems. 2010;18(3):461–472.
[7] Carvalho JP. On the semantics and the use of fuzzy cognitive maps and dynamic cognitive maps in social sciences. Fuzzy Sets and Systems. 2013;214:6–19.
[8] Vladimir NV, et al. Statistical learning theory. Xu JH and Zhang XG translation Beijing: Publishing House of Electronics Industry, 2O04. 1998;.
[9] Pal SK, Mitra S. Neuro-fuzzy pattern recognition: methods in soft computing. John Wiley & Sons, Inc.; 1999.
[10] Haykin SS, Haykin SS, Haykin SS, Elektroingenieur K, Haykin SS. Neural networks and learning machines. vol. 3. Pearson education Upper Saddle River; 2009.
[11] Rosenblatt F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Cornell Aeronautical Lab Inc Buffalo NY; 1961.
[12] Werbos PJ, et al. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE. 1990;78(10):1550–1560.
[13] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.
[14] Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine. 2018;13(3):55–75.
[15] Girosi F, Jones M, Poggio T. Regularization theory and neural networks architectures. Neural computation. 1995;7(2):219–269.
[16] Bird S, Klein E, Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc."; 2009.
[17] Poria S, Cambria E, Hazarika D, Vij P. A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:161008815. 2016;.
[18] Nafari M, Weaver C. Query2Question: translating visualization interaction into natural language. IEEE transactions on visualization and computer graphics. 2015;21(6):756–769.
[19] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. arXiv preprint arXiv:160301360. 2016;.
[20] Wang X, Liu Y, Chengjie S, Wang B, Wang X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1; 2015. p. 1343–1353.
[21] Socher R, Lin CC, Manning C, Ng AY. Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th international conference on machine learning (ICML-11); 2011. p. 129–136.
[22] Walker W, Lamere P, Kwok P, Raj B, Singh R, Gouvea E, et al. Sphinx-4: A flexible open source framework for speech recognition. Sun Microsystems, Report Number: TR-2004-139. 2004;.
[23] Krupakar H, Rajvel K, Bharathi B, Deborah SA, Krishnamurthy V. A survey of voice translation methodologies—Acoustic dialect decoder. In: 2016 International Conference on Information Communication and Embedded Systems (ICICES). IEEE; 2016. p. 1–9.
[24] Su J, Zeng J, Xiong D, Liu Y, Wang M, Xie J. A hierarchy-to-sequence attentional neural machine translation model. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP). 2018;26(3):623–632.
[25] Sharma PK, Sahu N. A Hybrid voice identification System with Fuzzy Technique and ART2 Neural Network on BPF Technique. Journal of Communication Engineering & Systems. 2018;8(3):1–6.
[26] Huffman WC, Pappas M. System and Method for Voice-to-Voice Conversion. Google Patents; 2018. US Patent App. 15/989,062.
[27] Lavan N, Merriman SE, Ladwa P, Burston LF, Knight S, McGettigan C. 'Please sort these voice recordings into 2 identities': Effects of task instructions on performance in voice sorting studies. British Journal of Psychology. 2019;.
[28] Lieto A, Moro D, Devoti F, Parera C, Lipari V, Bestagini P, et al. "Hello? Who Am I Talking to?" A Shallow CNN Approach for Human vs. Bot Speech Classification. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2019. p. 2577–2581.
[29] Ladefoged P, Disner SF. Vowels and consonants. John Wiley & Sons; 2012.
[30] Styler W. Using Praat for linguistic research. University of Colorado at Boulder Phonetics Lab. 2013;.
[31] Boersma P, Weenink D. Praat, a system for doing phonetics by computer (Version 6.0. 28). Institute Of Phonetic Sciences University of Amsterdam (up-to-date version of the manual at http://www fon hum uva nl/praat/). 2017;.
[32] Kingsbury BE, Morgan N, Greenberg S. Robust speech recognition using the modulation spectrogram. Speech communication. 1998;25(1-3):117–132.
[33] Khunarsal P, Lursinsap C, Raicharoen T. Singing voice recognition based on matching of spectrogram pattern. In: 2009 International Joint Conference on Neural Networks. IEEE; 2009. p. 1595–1599.
[34] Pinkowski B. Principal component analysis of speech spectrogram images. Pattern recognition. 1997;30(5):777–787.
[35] Muda L, Begam M, Elamvazuthi I. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:10034083. 2010;.
[36] Choi K, Joo D, Kim J. Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras. arXiv preprint arXiv:170605781. 2017;.
[37] Han Y, Park J, Lee K. Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification. the Detection and Classification of Acoustic Scenes and Events (DCASE). 2017;p. 1–5.
[38] Choi K, Fazekas G, Sandler M, Cho K. A comparison of audio signal preprocessing methods for deep neural networks on music tagging. In: 2018 26th European Signal Processing Conference (EUSIPCO). IEEE; 2018. p. 1870–1874.

[39] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.
[40] Perraudin N, Balazs P, Søndergaard PL. A fast Griffin-Lim algorithm. In: 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. IEEE; 2013. p. 1–4.