Integrated Citizen Centered Digital Health and Social Care
A. Värri et al. (Eds.)

doi:10.3233/SHTI200709

Word-Final Phoneme Segmentation Using Cross-Correlation

Emilian-Erman MAHMUT[a,1], Stelian NICOLA[a] and Vasile STOICU-TIVADAR[a]
[a] *Department of Automation and Applied Informatics, Politehnica University Timisoara, Romania*

Abstract. The goal of this paper is to present a word-final target phoneme automated segmentation method based on cross-correlation coefficients computed between a reference sound wave and a sample sound wave. Most existing Speech Sound Disorder (SSD) Screening solutions require human intervention to a greater or lesser extent and use segmentation methods based on hard-coded time frames. Moreover, existing solutions extract features from the frequency domain, which entails large amounts of computational power to the detriment of real-time feedback. The pre-processing algorithm proposed in this paper, implemented in a Python version 3.7 script, automatically generates 2 new .wav files corresponding to the phonemes found in word-final position in the initial sound waves. The newly-generated .wav files are meant to be used as valid and homogeneous input in a subsequent classification stage aimed at rigorously discriminating mispronunciations of the target phoneme and assist Speech-Language Pathologists (SLPs) with the SSD screening.

Keywords. Cross-correlation, audio segmentation, SSD

1. Introduction

Using over 100 distinct muscles in order to control minute movements, triggered by nerve impulses traveling through the cortical and subcortical structures of the brain at speeds over 100 m/s, the articulatory apparatus displays the most complex behavior in the human body [1-2]. If undetected and untreated in due time, language disorders may have severe consequences on the development of children's personality and behavior, including scarcity at school and poor social skills. The ever-increasing prevalence of persistent SSDs (Speech Sound Disorders) among preschoolers and elementary schoolers [3] in conjunction with the key role played by early diagnosis and subsequent treatment in the therapeutic outcome reinforce the need for an automated mispronunciation screening solution. The automated screening output stored in an anonymized, online database would provide access to analyses and statistics based on various demographic parameters of interest. The screening application should rigorously assess the similarity between a reference segment (the Speech Language Pathologist's pronunciation of a word or logatome containing the target phoneme) and a sample segment (the subject's pronunciation of the same word or logatome) of a target phoneme

[1] Corresponding Author, Emilian-Erman Mahmut, Politehnica University Timisoara, P-ta Victoriei no. 2, Timisoara, Romania; E-mail: emilian.mahmut@aut.upt.ro

within the same phonetic context. In any given utterance, the neighboring phonemes affect the target phoneme both progressively (sound *n* affects sound *n+1*) and regressively (sound *n+1* affects sound *n*). Most existing audio segmentation algorithms serve as a preliminary (pre-processing) step whereby new segments are created to be used as input for subsequent feature extraction, analysis and/or classification. Such pre-processing algorithms are mainly devised and used in automatic speech recognition (ASR) and multimedia applications, such as, for instance, music information retrieval (MIR). Speech processing algorithms extract features from the time and frequency domains and aim mainly to provide solutions for a robust classification of several categories of sounds: noise, silence, voiced and unvoiced phonemes and/or parts thereof. Real-time output is a ubiquitous requirement and it is achieved at the cost of a large amount of computational power, usually involving a large amount of training data in the subsequent classification stage. The fixed frame size and rate (FFSR) technique is widely-used in the ASR systems with solid results, except for recognition of speech in noisy environments. Paper [4] gives a comprehensive presentation of the challenges of this research field and proposes a speech envelope-based segmentation solution (inspired and supported by the neuroscientific perspective) to the shortcomings of the FFSR technique. An extensive classification of speech segmentation algorithms and feature extraction techniques is given in paper [5].

In reference [6] we presented the cross-correlation based audio segmentation method for phonemes in word-initial position and briefly discussed the corresponding results. The method presented in this paper focuses on segmenting target phonemes in the final position within an utterance. The pre-processing algorithm is meant to provide adequately-extracted reference and sample segments that are homogeneous in terms of duration and context, to serve as valid input for a subsequent processing stage, i.e. an automated SSD screening solution, which is the main objective of our research project. Several criteria were adopted in the development of the SSD Screening application: non-invasiveness (reduced emotional stress), cost-efficiency (using open-source frameworks), time-efficiency (real-time feedback), mobility (access to remote/rural areas), and modularity (connectivity with computer-aided speech therapy applications).

2. Method

The homogeneously-trimmed segments are obtained using a Python version 3.7 script. The flowchart below (Figure 1) describes the segmentation of the phoneme found in final position within an utterance. The algorithm consists of 5 main steps:

- In step 1 the algorithm reads both audio files (SLP and SUB) in reverse order and generates 2 corresponding .csv (comma-separated value) files based on the .wav (waveform audio file format) file amplitude data;
- The two .csv files consisting of the amplitude data in reverse order are read in step 2. A data range encompassing the first 5000 values was considered sufficient to cover the target phoneme found in final position. Cross-correlation equates the lag value with the number of indexes by which the sample signal (SUB) is shifted to the left or to the right of the reference signal (SLP);
- The following step (step 3) declares and initializes two variables, max_corr_l and max_corr_r, in order to compute the maximum cross-correlation corresponding to each displacement, respectively to the left (lag_l) and to the right (lag_r);

- The cross-correlation coefficients are computed for every single lag to the left and to the right. If the correlation coefficient computed for the current lag is larger than the correlation coefficient computed for the previous lag, then max_corr_l respectively max_corr_r is assigned the new maximum value. The algorithm stores the index of the maximum correlation (lag_max_l or lag_max_r). Step 4 is completed once all the 10,000 correlation coefficients have been computed for the displacement to the left (lag range: 0; - 4999) and, respectively, to the right (lag range: 0; 4999). Two maximum correlation coefficients are identified, one for each direction.

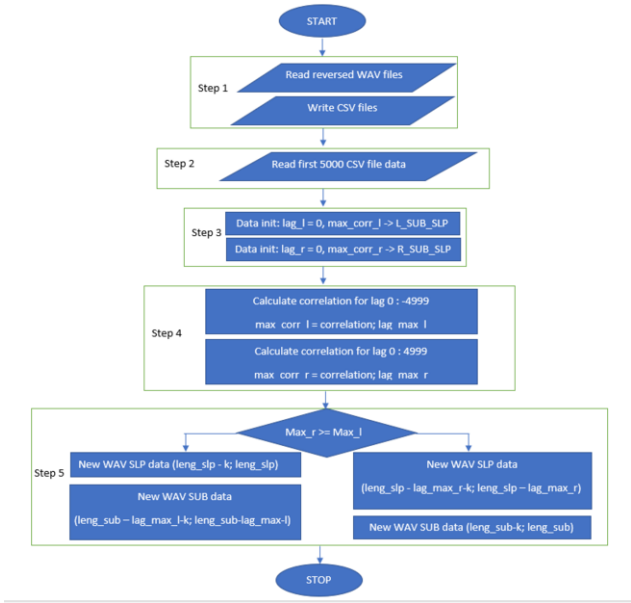

Figure 1. Pre-processing algorithm flowchart.

- In step 5, the 2 maximum correlation coefficients (left and right) are compared and 2 new audio segments (new WAV SLP and new WAV SUB) are generated. If max_corr_l > max_corr_r, the newly-generated audio files will consist of the amplitude data within the (leng_slp - k; leng_slp) range for the SLP, and within the (leng_sub – lag_max_l-k; leng_sub-lag_max-l) range for the SUB. If max_corr_l < max_corr_r, the newly-generated audio files will consist of the amplitude data within the (leng_slp - lag_max_r-k; leng_slp – lag_max_r) range for the SLP, and within the (leng_sub-k; leng_sub) range for the SUB. The value of the *k* constant appearing in the aforementioned ranges determines the number of amplitude data contained in the newly-generated audio files. The value of *k* (7,000) was determined empirically so as to cover the target phoneme in the final position within the analyzed utterance. The Python script allows for a fairly easy modification of the value of k. However, higher values of *k* determine the inclusion of larger portions of the preceding phoneme into the newly generated .wav files (reference and sample segment). The 2 newly-generated audio files have the following parameters: sample rate = 44100.0 Hz, maximum duration = 1.0s, frequency = 440.0 Hz.

3. Results

The pronunciations of a population of 30 primary school pupils (subjects aged 5-7 from the CNB College in Timisoara) were fed to the pre-processing segmentation algorithm. For 63.33% of the subjects the maximum cross-correlation values were obtained by shifting the sample signal (SUB) to the left of the reference signal (SLP), while for the remaining 36.77% the maximum cross-correlation values were identified by moving the sample signal to the right. Figure 2 shows the polynomial trendline of the initial .wav files (whole word, /f-a-r/, Romanian word for *headlight*): reference (SLP, left side) and sample (SUBJECT, right side). Figure 3 displays the polynomial trendline for the newly-generated segments: reference versus sample (final phoneme, /r/). As it may be observed, the R-squared value of the automatically-generated segments is higher (i.e. major goodness-of-fit) as opposed to the corresponding R-squared value of the manually segmented initial audio file (Figure 2). The maximum and minimum amplitude values (crests and troughs) are marked by orange squares in the diagram.

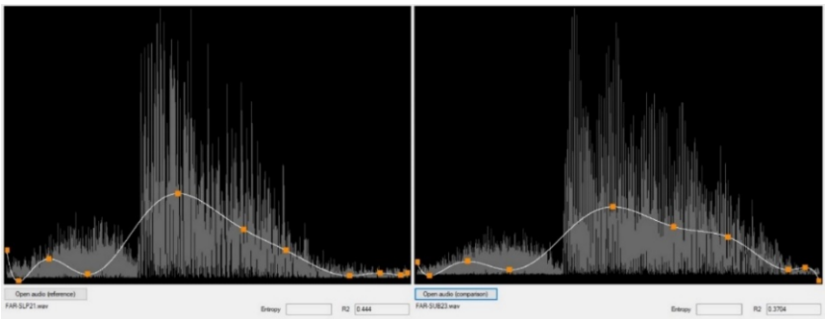

Figure 2. Initial .wav files, reference (left, $R^2 = 0.444$) versus sample (right, $R^2 = 0.3704$) (whole word /f-a-r/).

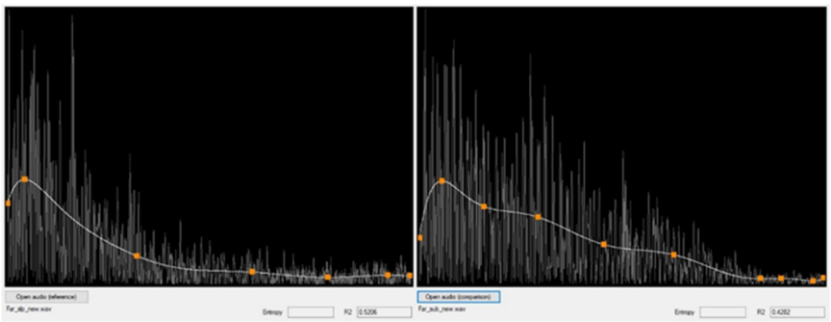

Figure 3. Newly-generated segments: reference (left, $R^2 = 0.5206$) versus sample (right, $R^2 = 0.4282$) (final phoneme /r/).

Table 1 contains the output data: the maximum lag values to the left (lag_max_l) are contained in the [-3858; -113] range and the maximum lag values to the right (lag_max_r) are included in the [0; 2036] range. The output data confirms the effectiveness (in terms of computational workload) of the empirically-determined data range of 5000 amplitude data values used in the Python script. Increasing such data range does not produce better cross-correlation values. The R^2 value is constant (0.5206) for all the segments where the maximum cross-correlation value is to the left while in the

cases where the maximum cross-correlation value is found to the right, the R^2 value is variable.

Table 1. Maximum cross-correlation values and corresponding lags

Subject (SUB)	max_l (lag_max_l)	max_r (lag_max_r)	R^2_SLP	R^2_SUB
1	0.010026 (-2918)	**0.027897 (1493)**	0.4939	0.4441
2	**0.034544 (-1791)**	0.018945 (863)	0.5206	0.4706
3	**0.021644 (-266)**	0.016214 (388)	0.5206	0.4812
4	**0.024533 (-113)**	0.019913 (152)	0.5206	0.4146
…				
30	0.018012 (-270)	**0.019100 (161)**	0.5305	0.2765

4. Discussion and Conclusions

The cross-correlation based pre-processing algorithm is an efficient solution that generates homogeneous segments to be used as valid input for the classification stage. It does not have a temporal limitation (such as the FFSR fixed-size frames and shifts [4]) and it is language-independent. The R^2 values obtained for the newly-generated segments are better than the R^2 values corresponding to the initial, manually-segmented audio files. The value assigned to the *k* constant was validated by the newly-generated audio files. Comparing an utterance issued by an adult voice (SLP) with that of a child (primary schoolers) is a limitation of the current state of our algorithm. Therefore, our new approach to this research thread entails the calculation of the autocorrelation coefficient for the 2 newly-generated segments so as to determine the energy level of each segment. Subsequently, the ratio between the 2 aforementioned autocorrelation coefficients (autocorrel_slp/autocorrel_sub) will be used to increase the energy level of the sample files (subject signals). The current classification stage performed in our C# (.NET) application [7] is based on the representation of the polynomial trendline of the audio files. To increase the precision of the screening solution, a logarithmic function will also be added, in an attempt to obtain higher R^2 values (better goodness-of-fit) for the new segments.

References

[1] Nuwer MR, Pouratian N. Monitoring of neural function: electromyography, nerve conduction, and evoked potentials, Youmans and Winn Neurological Surgery (7th ed.), Winn HR, Philadelphia, PA: Elsevier, 2017.

[2] Conant D, Bouchard KE, Chang EF. Speech map in the human ventral sensory-motor cortex, Current Opinion in Neurobiology, 2014; 24(1): 63–67.

[3] Wrenn Y, Miller LL, Peters TJ, Emond A and Roulstone S. Prevalence and Predictors of Persistent Speech Sound Disorder at Eight Years Old: Findings From a Population Cohort Study, J Speech Lang Hear Res. 2016 Aug; 59(4): 647–673.

[4] Lee B and Cho KH, Brain-inspired speech segmentation for automatic speech recognition using the speech envelope as a temporal reference, Scientific Reports 6, Article number: 37647; 2016 Nov.

[5] Alaa ES, Sherif MA, Salah EH, Mohsen R. A Review: Automatic Speech Segmentation, International Journal of Computer Science and Mobile Computing; 2017 Apr; Vol. 6 Issue.4, p. 308-315.

[6] Mahmut EE, Nicola S and Stoicu-Tivadar V. Cross-correlation based automated segmentation of audio samples, Studies in Health Technology and Informatics, 2020 Jun 26; (Ebook) Volume 272: The Importance of Health Informatics in Public Health during a Pandemic, IOS Press, p. 241-244

[7] Mahmut EE, Berian D, Della Ventura M and Stoicu-Tivadar V, Optimization of Entropy-based automated Dyslalia Screening Algorithm, Studies in Health Technology and Informatics, 2020 Jun 16; (Ebook) Volume 270: Digital Personalized Health and Medicine, IOS Press, p. 357 – 361.