Preproceedings of the Federated Conference on Computer Science and Information Systems pp. 221–222
DOI: 10.15439/2021F146
ISSN 2300-5963 ACSIS, Vol. 26

Speech sound detection employing deep learning

Cezary Polak, Jakub Mańkowski, Wiktor Uciński,
Patryk Schramka, Mikołaj Mysiakowski
Gdańsk University of Technology
Faculty of Electronics Telecommunication and Informatics
Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: {s165516, s172466, s160299,
s168827, s165771}@student.pg.edu.pl

Adam Kurowski
Gdańsk University of Technology
Faculty of Electronics Telecommunication and Informatics
Multimedia Systems Department
Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: adakurow@pg.edu.pl

***Abstract*—The primary way of communication between people is speech, both in the form of everyday conversation and speech signal transmitted and recorded in numerous ways. The latter example is especially important in the modern days of the global SARS-CoV-2 pandemic when it is often not possible to meet with people and talk with them in person. Streaming, VoIP calls, live podcasts are just some of the many applications that have seen a significant increase in usage due to the necessity of social distancing. In our paper, we provide a method to design, develop, and test the deep learning-based algorithm capable of performing voice activity detection in a manner better than other benchmark solutions like the WebRTC VAD algorithm, which is an industry standard based mainly on a classic approach to speech signal processing.**

I. Introduction

VOICE transmission-based techniques are constantly being improved to enable the broadcast of the human voice with the highest quality possible while reducing the demand for transmission bandwidth. In mobile networks, despite multimedia content being a more and more prominent part of transmitted data, voice transmission is still a crucial and basic functionality. To reduce the extensive occupation of the transmission bandwidth, speech detection methods have been employed in the above-mentioned applications to detect and transmit only the speech-containing part of the conversation, which leads to a decrease in bandwidth use. This class of algorithms is called voice activity detectors (VADs). Despite the significant development of techniques related to deep learning, the task is often performed by relatively simple heuristic algorithms. In literature, it is possible to find examples of VAD applications for which authors directly stress, that the choice of the VAD used for carrying out e.g. the speech quality enhancement task directly determines how good the output of such enhancement algorithms is [1].

II. Data acquisition

The first step to design and develop a voice activity detection algorithm employing any machine learning technique is to gather the database containing speech recordings and interfering signals. In our case, a signal obtained from an Internet podcast was used. It was an excerpt of an interview with president Andrzej Duda carried out by Karol Paciorek [2]. The length of the interview is 1 hour 16 minutes and 48 seconds. The recording was downsampled to the sampling rate of 16 kHz to reduce the required memory and processing power of the VAD algorithm. Additionally, procedures for introducing additive white Gaussian noise (AWGN) were designed. They were designed in such a way, that an arbitrary value of signal-to-noise ratio (SNR) can be obtained. Also, an additional recording containing cocktail party type of noise was also obtained from the Freesound online sound archive [3]. Such choice of interfering signals makes it possible to represent a wide range of real-world cases that may present difficulties for the algorithm under test in performing the voice activity detection task.

Next, recordings were annotated. In the process, all parts of the recording that contain speech fragments were marked with the appropriate label by the authors. Annotation was performed for signal frames of 200 ms as a human can easily hear if such frame contains speech. For the VAD algorithm, each of 200 ms frames was split into 20 ms frames, as this is one of the typical lengths for which VAD algorithms operate. Each of 10 short 20 ms frames derived from the single 200 ms long frame had the same label as the long, 200 ms one. Such a database is then used to generate test recordings containing combinations of speech signals and AWGN with varying SNR levels and ones contaminated by the cocktail party-type noise.

The data collected in the aforementioned process were intended to be used as the input to the convolutional neural network (CNN), and therefore they had to be parameterized. For each frame, an MFCC-gram was calculated. It was calculated with the FFT frame length of 32 points, and overlap factor equal to 0.75, a number of MFCC coefficients was set to 10. Such processing resulted in obtaining an MFCC-gram matrix having shape of 37 parameters x 10 parameters associated with every 20 ms long frame. For calculation, a librosa Python library was used. [4]. The interview used as the data source for our study was recorded with a studio-grade microphone, therefore we considered it to be not contaminated by any significant amount of noise.

III. The experiment

Both the clean and noise-contaminated audio frames were processed by two algorithms. Namely, the reference algorithm which in our case was the WebRTC VAD algorithm [5], and

TABLE I
ACCURACIES OF TWO TESTED VAD ALGORITHMS FOR THE CASE OF INPUT SIGNAL CONTAINING NO NOISE, AND FOR THE INPUT SIGNAL CONTAINING THE AWGN OR THE COCKTAIL-PARTY NOISE.

VAD algorithm	noise type	SNR [dB]						
		15	10	5	0	-5	-10	-15
WebRTC	**no noise**	0.937						
	AWGN	0.936	0.938	0.940	0.943	0.943	0.940	0.925
	cocktail	0.936	0.937	0.937	0.936	0.928	0.919	0.882
CNN-based	**no noise**	0.964						
	AWGN	0.946	0.919	0.842	0.696	0.606	0.565	0.548
	cocktail	0.925	0.850	0.706	0.606	0.568	0.555	0.549

the algorithm designed by the authors which was based on convolutional neural networks (CNNs). Processing employing neural networks was implemented with the use of TensorFlow Python library [6]. The structure of a CNN used as a VAD algorithm was as follows:

1) a convolutional layer containing 32 channels with (2,2) filters and ReLu activation function, followed by a (2,2) max pooling operation with a stride parameter set to (2,2), and a batch normalization layer,
2) a group of layers identical to 1),
3) a group of layers identical to 1),
4) a flattening layer,
5) a dense layer with 64 neurons with ReLu activation function,
6) a dropout layer with dropout coefficient of 0.3,
7) an output dense layer containing 2 neurons (as output is encoded in a one-hot manner), having softmax activation function.

IV. RESULTS

Input dataset consisted of 200190 audio frames, 53.88% (107860 frames) of the examples present in the dataset were associated with the speech signal presence, 46.12% (92330 frames) were associated with a so-called silence by which we mean frames not containing any speech signal. The input dataset was split into training (60% of examples), validation (20% of examples), and test (20% of examples) subsets. Data were divided in a stratified manner, so proportions of speech and silence in each of them were similar to ones in the original dataset. The neural network was trained for 100 epochs. The final accuracy achieved by the algorithm for the training set was 0.983, for the validation a dataset accuracy of 0.964 was achieved.

Performances of both the CNN-based, and the WebRTC VAD were evaluated on speech signals which were either noise-free recordings obtained from the podcast, or the audio fragments contaminated by additional noise. The CNN-based VAD was evaluated only on examples from the test dataset, as the test dataset was the only piece of data not used in the process of training. WebRTC VAD was evaluated on the whole dataset, as no training was necessary in its case. Two types of noise were used to contaminate the input signal, namely the AWGN, and the cocktail-party noise. The signal-to-noise ratio (SNR) of the noise signal added to the inputs of both tested algorithms was varied from 15 dB up to -15 dB with a decrement of 5 dB. Results of all tests carried out in our experiment are shown in Tab. I

V. CONCLUSION

Results obtained with the use of a CNN-based VAD algorithm are promising for conditions of nonexistent or low-amplitude noise (SNR = 15 dB, AWGN noise) if compared to the WebRTC VAD algorithm. Therefore, the algorithm proposed in our paper may be used for automatic labeling of signals for research purposes, as it is capable of obtaining better accuracies than the WebRTC VAD used as a baseline for our study. On the other hand, even for the SNR of 10 dB the performance of CNN-based algorithms drops below the baseline results and degrades in a significantly more pronounced manner for SNRs lower than 10 dB. For the cocktail-party type of noise, this degradation is even worse than for the AWGN. Therefore, the use of simple CNN-based VADs is not encouraged in the case of noisy environments. Possible countermeasures are, e.g. use of noise-contaminated audio frames for the training of the algorithm which may be an interesting future work to be carried out in the case of our research. We also plan to test our approach on signals other than on-line interviews, such as conversation of three or more people with fragments of simultaneous speech coming from two or more speakers.

REFERENCES

[1] H. Haneche, B. Boudraa, and A. Ouahabi, "A new way to enhance speech signal based on compressed sensing," *Measurement*, vol. 151, p. 107117, 2020. doi: https://doi.org/10.1016/j.measurement.2019.107117. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119309832

[2] K. Paciorek. Andrzej Duda o: LGBT, TVP, koronawirusie, głosach po Bosaku i o szansach w starciu z Trzaskowskim (in Polish). Youtube (Imponderabilia channel). [Online]. Available: https://www.youtube.com/watch?v=Izxj72bg4A4

[3] Freesound. Party Sounds recording from the online royalty free recordings archive. [Online]. Available: https://freesound.org/people/FreqMan/sounds/23153/

[4] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[5] GitHub. Python interface to the WebRTC voice activity detector. [Online]. Available: https://github.com/wiseman/py-webrtcvad

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/