INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Using X-vectors for Speech Activity Detection in Broadcast Streams

Lukas Mateju, Frantisek Kynych, Petr Cerva, Jindrich Zdansky, Jiri Malek

Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic
{lukas.mateju, frantisek.kynych, petr.cerva, jindrich.zdansky, jiri.malek}@tul.cz

Abstract

A new approach to speech activity detection (SAD) is presented in this work. It allows us to reduce the complexity and computation demands, namely in services that process streaming speech, where a SAD module usually forms the first block of the data pipeline (e.g., in a platform for 24/7 broadcast transcription). Our approach utilizes x-vectors as input features so that, within the subsequent pipeline stages, these embedding instances can also directly be employed for speaker diarization and recognition. The x-vectors are extracted by feed-forward sequential memory network (FSMN), allowing for modeling long-time dependencies; they thus form an input into a computationally undemanding binary classifier, whose output is smoothed by a decoder. Evaluation is performed on the standardized QUT-NOISE-TIMIT dataset as well as on broadcast data with large portions of music and background noise. The former data allows for comparison with other existing approaches. The latter shows the performance in terms of word error rate (WER) and reduction in real-time factor (RTF) of the transcription process.
Index Terms: x-vectors, speech activity detection, streamed data processing, deep neural networks

1. Introduction

X-vectors allow mapping of variable-length utterances to fixed-dimensional embeddings. They can be extracted using various deep neural network (DNN) architectures and provide robust representations when a large amount of training data is used. It has recently been shown that these embeddings are able to encode various attributes of an input utterance including its length, channel information, speaker's gender, speaking rate or even spoken content [1]. In [2], x-vectors were also successfully applied to the spoken language recognition task.

However, x-vectors were originally crafted for speaker recognition [3]. The speaker characteristics encoded in the embedding can also be utilized for speaker diarization (SD). In this task, x-vectors are clustered using various techniques [4, 5]. The SD systems usually operate just over speech segments of the input data. In some SD evaluation tracks [6], these segments are extracted manually; in other tracks, as well as in practice, an automatic SAD module must be used.

To the best of our knowledge, the existing SAD approaches do not utilize x-vectors (see also Sec. 2). Therefore, two systems must be employed over different features (one for SAD and one for SD) in real SD applications. This approach increases computation demands and the complexity of the whole speech processing pipeline, which is undesirable, particularly in systems designed for processing streamed data. In our case, an example of such a system is a platform for 24/7 monitoring of various TV and radio streams, namely in Slavic languages (see our multilingual radio monitoring application[1]). In this type of application, the SAD module forms the first block of the data-processing pipeline and its output, consisting of the speech segments, is used for a) speaker diarization (followed by speaker recognition) and b) speech transcription. In the latter task, the SAD module allows us to reduce the computation demands dramatically as some broadcast streams contain large amounts of non-speech parts, such as songs or advertisements.

In this work, we extend our previous investigations to the use of x-vectors for speech activity (and overlapped speech) detection [7] by presenting a complete SAD scheme, which is particularly suitable for the above-mentioned frame-wise processing. The use of x-vectors emphasizes one of the advantages of our approach because they can also be directly employed for speaker diarization and recognition in the subsequent data pipeline stages. Experimental results presented in Sec. 5 show that our approach yields results that are similar to (or even better than) the existing state-of-the-art SAD methods.

2. Related work

Most commonly, speech activity detection is run in two consecutive phases: feature extraction followed by speech/non-speech classification. In the former phase, the more classical approaches utilize, e.g., zero-crossing rate [8], energy [9], or auto-correlation function [10]. Over the years, more complex features including multi-resolution cochleagram features [11], Mel-frequency cepstral coefficients (MFCCs) [12], or pitch related features [13] have been applied with great success. Furthermore, bottleneck features extracted from DNNs have been proposed in [14]. In practice, various combinations of individual features are often used to achieve the best possible results.

In the latter phase, different classifiers can be employed, including support vector machines [15] or Gaussian mixture models (GMMs) [16, 17]. In recent years, various DNN architectures, such as fully connected (FC) feed-forward DNNs [12] or convolutional neural networks (CNNs) [18] have frequently been applied. The modeling power of recurrent neural networks (RNNs) has been exploited as well [19]. Specifically, the long short-term memory (LSTM) RNNs [20, 21] have recently gained a lot of popularity. More complex approaches, e.g., boosted DNNs [11] or convolutional LSTM neural networks [22], have also been presented. Moreover, convolutional gated recurrent unit (GRU) RNNs [23] have been utilized successfully. Recently, an adaptive context attention model was proposed in [24]. Finally, unsupervised approaches, such as rVAD [25], have been applied as well.

To improve the accuracy of the speech activity detection, the outputs from a given classifier can also be smoothed. Different techniques, such as the Viterbi decoder [12], weighted finite-state transducers (WFSTs) [26], or temporal smoothing layers, CNN or RNN (with bidirectional GRU) ones [27], have been suggested for this purpose.

[1] https://tul-speechlab.gitlab.io/

http://dx.doi.org/10.21437/Interspeech.2021-192

Table 1: *The structure of FSMN-based x-vector extractor.*

Layer	Layer context	Total context	Input x output
FSMN 1	$\ell \pm 80$	161	40×1024
FSMN 2	$\ell \pm 4$	169	1024×768
FSMN 3	$\ell \pm 4$	177	768×512
FSMN 4	$\ell \pm 4$	185	512×384
FSMN 5	$\ell \pm 4$	193	384×256
FSMN 6	$\ell \pm 4$	201	256×128
FC 1	ℓ	201	128×128
Pooling	$\ell \pm 20$	241	$(41 \cdot 128) \times 128$
FC 2	ℓ	241	128×128
Softmax	–	241	$128 \times Nspeakers$

3. Proposed approach

The proposed approach consists of three consecutive steps, which are described in detail in the following subsections:

1. Extraction of x-vectors using DNN.
2. Classification of x-vectors into two classes.
3. Smoothing the output from the classifier by a decoder.

3.1. X-vectors extraction

In the first step, a vectorized variant of the FSMN [28] is employed for x-vectors extraction. This architecture allows us to model long-time dependencies in the input signal similar to RNNs, but it eliminates the recursion by adding several memory blocks with trainable weight coefficients into each layer of a standard feed-forward FC DNN. The memory blocks use a tapped-delay line structure to encode the long context information into a fixed-size representation. That means that the FSMN layers are built on top of the context of the earlier layers; the final context is thus a sum of the partial ones.

The structure of FSMN used in this work is described in Table 1, where the symbol ℓ denotes the current frame, on which the temporal context is centered. The pooling layer computes only the means of the frames (omitting the variances) in the context of 41 consecutive frames. In all neurons, exponential linear unit (ELU) is used as the activation function. On the input, each frame of the signal is represented by 40 log filter bank coefficients (FBCs) computed from 25-ms-long frames with frame-shifts of 12.5 ms each. Table 1 shows that the extractor operates with a total context of 241 frames, which corresponds to 3.0125 seconds. Note that, within the context of the first layer (i.e., 161 frames), cepstral mean subtraction (CMS) is applied and the x-vectors are extracted after the pooling layer.

3.2. Binary classification of x-vectors

In the second step, the extracted embeddings are utilized by a binary DNN-based classifier that produces probabilities for the speech/non-speech classes. The basic and computationally undemanding architecture of the classifier employs two feed-forward FC hidden layers with 128 and 64 neurons. In Sec. 5.2, we also present results for other topologies.

3.3. Smoothing using WFST-based decoder

For smoothing the output from the classifier, a WFST-based decoder is employed in the last step. Its advantage is that it represents a general smoothing concept and allows us to model various sorts of smoothing approaches by merely choosing the corresponding transduction model and the respective semi-ring.

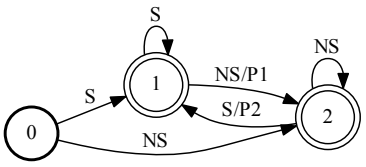

Figure 1: *The transducer (acceptor) representing the basic transduction model used for speech/non-speech detection.*

The transduction model employed in this work is depicted in Figure 1 and corresponds to the basic scheme that we have proposed for SAD over FBCs with feed-forward FC DNN and WFST-based decoder [29]. It consists of three states. The first state, denoted by 0, is the initial state. The transitions between states 1 and 2 emit the speech/non-speech labels and are penalized by penalty factors P1 and P2, respectively. Their values (100 and 100) were determined in several experiments not presented in this paper. More details about the decoding process with WFST can be found in [30], where we have also applied this type of smoothing to on-line language identification.

4. Experimental setup

4.1. Development data

For development purposes, a dataset consisting of 6 hours of TV and radio recordings in several Slavic languages (Czech, Slovak, Polish, and Russian) was constructed. These recordings contain clean speech segments and segments with music, background noise, jingles, and advertisements. Annotations of this data were at first created automatically (by our previous system) and later corrected and fine-tuned by hand. In total, 70% of all frames were marked as containing speech.

4.2. Evaluation metrics

Within this work, frame error rate (FER), miss rate (MR), and false alarm rate (FAR) are utilized to evaluate the overall frame accuracy of SAD [12]. Additionally, the quality of change-point detection (between speech and non-speech segments) given the alignment between detected and reference boundaries is expressed by the F-measure (Fm).

5. Experimental evaluation

5.1. X-vector extraction

The opening set of experiments is focused on the first step of the proposed approach: three different x-vector extractors based on the FSMN topology are investigated. In the first case, speech x-vectors (denoted by xv_s below) were calculated for all (7237) individual training speakers as usual. The training data consisted of Voxceleb2 [31], "train-360-clean" part of LibriSpeech [32] and 121 hours of clean Czech microphone recordings belonging to 922 speakers. In the second case of the xv_{s+n} extractor, the set of target speakers was extended by the noise class defined over the train part of the CHiME-4 dataset [33]. Finally, one special class representing music was also added. The resulting extractor is denoted by xv_{s+n+m}; the music data used for training contain 31 hours of various music recordings.

Given all these three types of extractors, three two-layer binary classifiers (see also Sec. 5.2) have been trained. The train-

Table 2: *Results of different x-vector extractors in comparison with FBCs on the development dataset.*

approach	FER[%]	MR[%]	FAR[%]	Fm[%]
xv_s	2.6	0.8	7.1	52.9
xv_{s+n}	**2.2**	0.7	6.0	**59.3**
xv_{s+n+m}	**2.2**	0.8	**5.8**	58.8
FBCs + DNN	2.5	**0.5**	7.7	54.3
FBCs + FSMN	2.6	**0.5**	8.2	53.2

ing dataset consists of 30 hours of clean speech, 30 hours of music and 30 hours of artificially mixed speech and music/noise recordings according to randomly chosen signal-to-noise ratio (SNR). All of these recordings are also concatenated in a random order to contain speech/non-speech transitions. For annotations, music recordings and the segments with SNR lower than 0 dB are labeled as non-speech and the rest as speech. Finally, the outputs of the classifiers are smoothed by the WFST-based decoder (see Sec. 3.3 for more details).

The results are summarized in Table 2. They show that all three x-vector extractors lead to a very low level of MR as well as FER. In terms of F-measure, the xv_{s+n} extractor yields the best results as the additional noise class improves the performance for more noisy segments. On the other hand, additional music class does not yield any further improvements.

The last two rows in Table 2 are dedicated to comparison with the approach we presented in [29]. It analogically employed a DNN-based binary classifier and a WFST-based decoder smoothing the outputs of the DNN. The main difference is that the classifier was trained directly on FBCs. The DNN has five hidden layers, each with 128 neurons, and the ReLU activation function is utilized. To allow fair comparison with the SAD module proposed in this work, the same context is used, i.e., the input feature vector is formed by concatenating 50 previous frames, the current frame (39-dimensional FBCs), and the 50 following frames. Local normalization has also been performed as for the x-vectors within a two-second window.

Finally, we also trained an FSMN-based classifier with the corresponding parameters.

The results in the last two rows in Table 2 show that the DNN- and FSMN-based baselines perform comparably but yield outcomes that are overall worse than those of the x-vector systems (the only exception is given by the lower MR values).

5.2. Binary classification

To evaluate the second step in the proposed approach, several different NN architectures varying in the complexity have been explored for binary classification.

The basic topology with 2 hidden layers (denoted as DNN-2HL) is described in Sec. 3.2 and was used in all previous experiments. The less complex variants include a) simple NN without any hidden layer (NN-0HL) and b) NN with one hidden layer with 128 neurons (NN-1HL). The latter network proved to be efficient for detecting overlapped speech in our previous study [7]. On the contrary, the more complex topology is represented by DNN with five hidden layers (DNN-5HL), each with 128 neurons (it corresponds to the classifier used in the previous experiments over FBCs). This DNN was trained with a) zero input context and b) 0.5-second context window (i.e., 25 previous frames, the current frame, and 25 following frames). In addition, two more complex architectures, time-delay neural network (TDNN) and FSMN, both having five hidden layers with 128 neurons per layer, have also been evaluated.

Table 3: *The performance of various binary classifiers over xv_{s+n} extractor.*

classifier	FER[%]	MR[%]	FAR[%]	Fm[%]
zero input context				
NN-0HL	**2.2**	0.9	**5.5**	59.7
NN-1HL	**2.2**	0.8	5.9	59.8
DNN-2HL	**2.2**	0.7	6.0	59.3
DNN-5HL	2.4	0.6	7.0	54.1
0.5-second input context				
DNN-2HL	2.4	0.7	6.8	52.6
TDNN-5HL	2.8	**0.5**	8.4	57.6
FSMN-5HL	2.4	0.6	7.0	**60.8**

Table 4: *The effect of smoothing the binary classifier's output on the performance of SAD.*

smoothing	FER[%]	MR[%]	FAR[%]	Fm[%]
none	4.2	2.5	8.6	0.5
MA 1 s	2.8	0.9	7.8	32.2
MA 2 s	2.8	0.6	8.4	48.5
MA 3 s	3.0	**0.5**	9.3	40.0
WFST	**2.2**	0.8	**5.9**	**59.8**

Based on the obtained results (see Table 3), several conclusions can be made. First, no additional input context is needed for classification as the x-vectors already encode long context information in the FSMN topology. Second, the deeper DNNs do not yield any further significant improvements over the corresponding shallower architectures. Third, the FSMN-based classifier achieves the highest F-measure at the cost of worse FER and much higher computational demands. To sum it up, we have chosen the NN-1HL classifier for further experimental evaluation as it represents a compromise between the SAD performance and computational demands.

5.3. The effect of smoothing

The third experiment is aimed at the last step of the proposed approach, i.e., smoothing the output of the classifier. Here, we show the importance of the WFST-based smoothing by comparing it to smoothing with moving average (MA; with different lengths of the window) as well as to no smoothing at all.

The results are presented in Table 4. They clearly show that the WFST-based decoder yields the best results in FER, FAR, and F-measure by a large margin. The necessity of smoothing is demonstrated in the first row, corresponding to no smoothing. In this case, the non-stop changes between speech and non-speech segments resulted in an extremely small value of the F-measure.

6. ASR results

Given the results from evaluation on the development data, the resulting SAD approach has also been evaluated in a real speech-transcription system. This system utilizes an FSMN-based acoustic model, an n-gram language model and a lexicon containing 400k of the most frequent Czech words. Two distinct Czech datasets have been used. The first one consists of

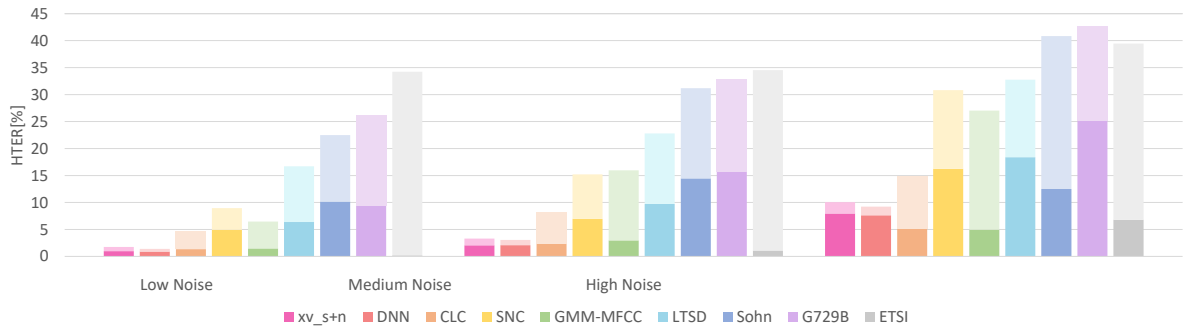

Figure 2: *A comparison among different SAD approaches across the QUT-NOISE-TIMIT corpus. The contributions of MR and FAR to HTER bars are displayed by darker and lighter shades, respectively.*

Table 5: *The influence of the proposed SAD approach on the performance of an ASR system.*

SAD	WER	RTF	WER	RTF
	TV news		local radio	
none	14.7	0.67	65.2	0.76
xv_{s+n}	**11.0**	0.33	**13.6**	**0.06**
FBCs + DNN	11.1	0.33	14.3	**0.06**
xv_{s+n} with red. context	11.4	**0.32**	14.4	**0.06**

four hours (22k words) of recordings from a live news TV channel, and approximately 60% of its content is speech. The other dataset comes from a local radio station. Its length is 8 hours (7k words), and only 10% of its duration is formed by speech.

The achieved results in terms of WER and RTF are presented in the first three rows in Table 5. They show that the proposed SAD module has slightly outperformed the similar module based on DNN and FBCs. The decrease in WER indicates how effectively the insertions are limited in non-speech parts and hardly omit any speech. Furthermore, the RTF has improved 2 and 12 times on the news and radio test sets, respectively. The RTF value of the SAD approach itself is 0.01. Note that the presented RTF values have been measured using processor Intel Core i7-3770K @ 3.50GHz.

Finally, it should also be noted that the latency of the SAD module is 3.2 seconds (out of which 1.7 seconds is caused by the decoder). This value may be considered too large for online applications (e.g., subtitling). For this reason, we have tried to reduce the latency by 0.75 seconds, taking a quarter of the context for the first fixed layer of the x-vector extractor. This operation has led to just a slight and acceptable decrease in WER (see the last row in Table 5).

7. Results for QUT-NOISE-TIMIT

A QUT-NOISE-TIMIT [17] corpus has been utilized to compare the proposed approach with five systems already presented in [17], two newer techniques [34, 35], and the DNN-based classifier with WFST-based smoothing described in Sec. 5.1. The five original approaches were: standardized advanced front-end ETSI [36], standardized VAD system ITU-T G.729 Annex B [37], Sohn's likelihood ratio test [38], Ramirez's long-term spectral divergence (LTSD) [39] and GMM-based approach over MFCCs [17]. The latter two techniques were VAD using subband noncircularity (SNC) [34] and complete-linkage clustering (CLC) for VAD [35].

The training and testing protocols for QUT-NOISE-TIMIT corpus were followed, as recommended in [17]. During the training, the only prior knowledge given to the system was the target environment SNR, low noise (10, 15 dB), medium noise (0, 5 dB) or high noise (-10, -5 dB). The proposed SAD approach was trained as described in Sec. 3, i.e., xv_{s+n} extractor followed by a classifier with just one hidden layer and with WFST-based smoothing applied. The one exception was the use of only QUT-NOISE-TIMIT data for training of the classifier.

Figure 2 depicts the obtained results in all target SNR environments: low, medium and high. In addition to MR and FAR, half-total error rate (HTER) has also been reported. It is defined as equal-weighted average of MR and FAR. The results show, that in all noise conditions, the proposed approach outperforms most other SAD systems by a large margin. The only exception is the DNN-based approach over FBCs, which yields a slightly better performance.

8. Conclusions

In this paper, a new SAD approach suitable for processing streamed data is proposed. The method utilizes FSMN-based x-vectors as the input features to a computationally undemanding binary classifier (with only a single hidden layer), whose output is smoothed by a WFST-based decoder.

The results achieved on the development set and the QUT-NOISE-TIMIT corpus show that the proposed method yields state-of-the-art results and is capable of outperforming many other approaches while operating in a frame-wise mode and with possibility of on-line use. It also has a similar performance as a directly comparable approach based on DNN-based classifier (but over FBCs) with WFST-based smoothing. However, the main advantage of our method in comparison with the DNN-based approach is implied by the fact that the x-vectors used for SAD can also be simply employed for SD or recognition in the subsequent stages of the data-processing pipeline.

Finally, additional experiments performed in an ASR system have proved that the use of the method allows us to reduce the WER value and significantly improve the RTF level of the transcription process.

9. Acknowledgements

This work was supported by the Technology Agency of the Czech Republic (project No. TH03010018), and by the Student Grant Competition of the Technical University of Liberec under project No. SGS-2019-3017.

10. References

[1] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," in *ASRU 2019, Singapore*, 2019, pp. 726–733.

[2] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," in *Odyssey 2018, Les Sables d'Olonne, France*, 2018, pp. 105–111.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP 2018, Calgary, Canada*, 2018, pp. 5329–5333.

[4] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *ICASSP 2017, New Orleans, USA*, 2017, pp. 4930–4934.

[5] M. Diez, L. Burget, F. Landini, and J. Cernocky, "Analysis of speaker diarization based on bayesian HMM with eigenvoice priors," *IEEE/ACM Transactions Audio, Speech & Language Processing*, vol. 28, pp. 355–368, 2020.

[6] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second DIHARD diarization challenge: Dataset, task, and baselines," in *Interspeech 2019, Graz, Austria*, 2019, pp. 978–982.

[7] J. Malek and J. Zdansky, "Voice-activity and overlapped speech detection using x-vectors," in *TSD 2020, Brno, Czech Republic*, 2020, pp. 366–376.

[8] B. Kotnik, Z. Kacic, and B. Horvat, "A multiconditional robust front-end feature extraction with a noise reduction procedure based on improved spectral subtraction algorithm," in *Interspeech 2001, Aalborg, Denmark*, 2001, pp. 197–200.

[9] G. Evangelopoulos and P. Maragos, "Speech event detection using multiband modulation energy," in *Interspeech 2005, Lisbon, Portugal*, 2005, pp. 685–688.

[10] H. Ghaemmaghami, B. Baker, R. Vogt, and S. Sridharan, "Noise robust voice activity detection using features extracted from the time-domain autocorrelation function," in *Interspeech 2010, Makuhari, Japan*, 2010, pp. 3118–3121.

[11] X. Zhang and D. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *Interspeech 2014, Singapore*, 2014, pp. 1534–1538.

[12] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks," in *Interspeech 2013, Lyon, France*, 2013, pp. 728–731.

[13] Y. Shao and Q. Lin, "Use of pitch continuity for robust speech activity detection," in *ICASSP 2019, Calgary, Canada*, 2018, pp. 5534–5538.

[14] L. Ferrer, M. Graciarena, and V. Mitra, "A phonetically aware system for speech activity detection," in *ICASSP 2016, Shanghai, China*, 2016, pp. 5710–5714.

[15] J. W. Shin, J. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Language*, vol. 24, pp. 515–530, 2010.

[16] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Interspeech 2012, Portland, USA*, 2012, pp. 1969–1972.

[17] D. Dean, S. Sridharan, R. Vogt, and M. Mason, "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms," in *Interspeech 2010, Makuhari, Japan*, 2010, pp. 3110–3113.

[18] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *Interspeech 2013, Lyon, France*, 2013, pp. 3497–3501.

[19] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *ICASSP 2013, Vancouver, Canada*, 2013, pp. 7378–7382.

[20] F. Eyben, F. Weninger, S. Squartini, and B. W. Schuller, "Real-life voice activity detection with LSTM recurrent neural networks and an application to hollywood movies," in *ICASSP 2013, Vancouver, Canada*, 2013, pp. 483–487.

[21] Q. Lin, T. Li, and M. Li, "The DKU speech activity detection and speaker identification systems for fearless steps challenge phase-02," in *Interspeech 2020, Shanghai, China*, 2020, pp. 2607–2611.

[22] R. Zazo, T. N. Sainath, G. Simko, and C. Parada, "Feature learning with raw-waveform CLDNNs for voice activity detection," in *Interspeech 2016, San Francisco, USA*, 2016, pp. 3668–3672.

[23] A. Vafeiadis, E. Fanioudakis, I. Potamitis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Two-dimensional convolutional recurrent neural networks for speech activity detection," in *Interspeech 2019, Graz, Austria*, 2019, pp. 2045–2049.

[24] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Letters*, vol. 25, pp. 1181–1185, 2018.

[25] Z. Tan, A. K. Sarkar, and N. Dehak, "rvad: An unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.

[26] H. Chung, S. J. Lee, and Y. Lee, "Endpoint detection using weighted finite state transducer," in *Interspeech 2013, Lyon, France*, 2013, pp. 700–703.

[27] J. Heitkaemper, J. Schmalenstroeer, and R. Haeb-Umbach, "Statistical and neural network based speech activity detection in non-stationary acoustic environments," in *Interspeech 2020, Shanghai, China*, 2020, pp. 2597–2601.

[28] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feedforward sequential memory networks: A new structure to learn long-term dependency," *CoRR*, vol. abs/1512.08301, 2015.

[29] L. Mateju, P. Cerva, J. Zdansky, and J. Malek, "Speech activity detection in online broadcast transcription using deep neural networks and weighted finite state transducers," in *ICASSP 2017, New Orleans, USA*, 2017, pp. 5460–5464.

[30] P. Cerva, L. Mateju, J. Zdansky, R. Safarik, and J. Nouza, "Identification of related languages from spoken data: Moving from off-line to on-line scenario," *Computer Speech & Language*, vol. 68, 2021.

[31] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018, Hyderabad, India*, 2018, pp. 1086–1090.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP 2015, South Brisbane, Australia*, 2015, pp. 5206–5210.

[33] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[34] J. Ramirez, J. C. Segura, M. C. Benitez, A. de la Torre, and A. J. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, pp. 271–287, 2004.

[35] H. Ghaemmaghami, D. Dean, S. Kalantari, S. Sridharan, and C. Fookes, "Complete-linkage clustering for voice activity detection in audio and visual speech," in *Interspeech 2015, Dresden, Germany*, 2015, pp. 2292–2296.

[36] J. Li, B. Liu, R. Wang, and L. Dai, "A complexity reduction of ETSI advanced front-end for DSR," in *ICASSP 2004, Montreal, Canada*, 2004, pp. 61–64.

[37] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, pp. 64–73, 1997.

[38] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *ICASSP 1998, Seattle, USA*, 1998, pp. 365–368.

[39] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, 1999.