2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE) | 978-1-6654-8910-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRAIE56454.2022.10054315

Speech Summarization Using Prosodic Features and 1-D Convolutional Neural Network

Sannidhya Raj Chauhan*, Sateeshkumar Ambesange*, Shashidhar G. Koolagudi*
*National Institute of Technology Karnataka
{src.1413013, pragyan.ai.school}@gmail.com, koolagudi@nitk.edu.in

***Abstract*—In this work, we have presented a method for speech summarization of audiobooks without converting them into the transcript. The model used is the 1-D convolutional neural network. The audio is segmented into sentences based on the silence between two consecutive sentences. We have used acoustic features of the sentence audio as input to our model. The output of our model is binary, which tells us whether to include this sentence in our summary or not. Thus, we have converted the task of speech summarization into a classification task. Then we have concatenated the classified audio chunks into one summary. We have compared the generated summary against the manually done summary. For better insights, we have used a text summarizer as a reference to see what the summary should include. The transcript is used for only that; otherwise, our method is independent of the text. The results obtained show us a possibility of a language-independent audio summarizer that retains the audio quality since we have used the original audio in our summary.**

***Index Terms*—Prosodic/Acoustic features, 1-D Convolutional Neural Network, Speech Summarization.**

I. Introduction

Many attempts have been made to improve and develop a text summary. The primary goal of text summarization [1]–[3] is to examine the lexical structure between paragraphs and phrases in a text. As a result, key sentences are chosen to provide a text summary. Speech summarization, on the other hand, is a promising method that reduces audio utterances into compact and understandable speech utterances. Although speaking has been the most convenient form of human communication, it is difficult to easily examine, extract, and use them in the form of captured audio signals. As a result, it is commonly transcribed. However, the recordings are often inaccurate and dissimilar to written language, containing superfluous information like repetition and fillers.

Furthermore, unnecessary material added in transcription as a result of automatic speech recognition (ASR) failures is prevalent. Thus, transcribing is ineffective for speech. A speech summary, which captures relevant information while eliminating extraneous and inaccurate information, is required. A Good speech summarizing system reduces time spent studying audio data and enhances information extraction performance. Summarizing audio files is often done in two steps: first, the transcript of the speech is generated using ASR, and then normal text summarization is applied to the transcript to construct the summary. But, this technique has significant drawbacks such as being unsuitable for languages that do not have sufficient data for training. It is prone to mistakes in the voice recognition system.

Moreover, some critical information, such as the speaker's emotional state, is not caught in the transcript, which may be essential for deciding whether or not to include a line in the summary. To solve these restrictions, we employed a speech summarizing approach based on high-level acoustic features retrieved from audio and then processed to provide speech summarization. There are two techniques for providing summaries:

- concatenating voice fragments derived from genuine speech.
- generating audio from summarized text using a text-to-speech system.

We picked the former one as it is more convenient in our case and retains the original quality. Prosodic properties [4]–[6] are sound qualities that we hear with our senses. They vary from lexical characteristics, which take into account the actual vocabulary. An utterance contains several prosodic features, including the number of syllables, the number of pauses, the duration time, the phonation time, the speaking pace, the articulation rate, the maximum, minimum, and average frequency, and the energy.

The paper is organized as follows; section 2 addresses relevant work done on this topic in the past. Section 3 discusses dataset creation. Section 4 goes about the approach that is being used. Section 5 focuses on the experimental setup and model that was employed. Section 6 shows the results of our technique. Section 7 concludes with a discussion of the result and future work.

II. Related Work

In this section, we will discuss the research conducted to solve the challenge of speech summarization.

[7] published one of the early studies on this topic, in which speech was summarized and presented in two formats: speech-to-text and speech-to-speech summarization. The audio is handled in three units in this work: sentence, word, and between padding units. Their importance is determined by three categories of scores: linguistic, significance, and confidence [8]. They are then summarized and computed based on the average of the three scores. A pause is introduced between phrases during concatenation to form continuity. The same

method was then enhanced in [9] to a dynamic programming methodology used to generate target summaries depending on some summarization ratio.

In [10], prosodic characteristics were combined with linguistic features to create the summary. They employed linguistic elements collected from continuous voice recognition system transcripts and prosodic features derived from utterances. They utilized four prosodic features: frequency, phoneme duration, sentence length, and power. The relevance score generated from both of these sorts of characteristics was combined for summarizing. It was discovered that by introducing prosodic elements, the outcomes improved.

The text summarizing approach to speech summary is explored in [11]. Many examples of speech summaries are discussed here, including broadcast news summarizing and meeting summarization, as well as the obstacles we may encounter while dealing with them. A voice summarization strategy that integrated prosodic, syntactic, and semantic aspects and utilized transcription from speech recognition for word confidence was addressed in [12]. They then employed dynamic programming to fix the problem.

Sameer Maskey et al. [13] suggested a technique for summarizing transcript-free speech. It was entirely based on the prosodic characteristics of the speech. A total of 12 acoustic characteristics were considered. They were as follows: speaking rate, frequency minimum, maximum, mean, frequency range and slope, RMS energy slope, minimum, maximum, and mean, and sentence duration. PRAAT [1] software [14] was used to extract the features. For the prediction, the Hidden Markov model [15] was utilised. Observation variables are acoustic feature sequences, and the hidden variable we wish to identify is whether or not to include that sentence in our summary. As a result, the model was trained on a succession of acoustic characteristics in this study. To construct the summary, we must feed the acoustic characteristics vector into the hidden Markov model, which will calculate the likelihood of including it in the summary. The work in this study demonstrated that speech summary might be accomplished only through the use of acoustic characteristics without the assistance of linguistic information, giving us the ability to summarize speech without converting it to text.

Many inventive ways have been created in subsequent research, such as in [16], where computer vision techniques are employed in conjunction with structural analysis for summarizing. The 1-D time-domain voice stream is converted into a 2-D picture here. It suggests an unsupervised learning approach for detecting a repeated pattern in an image. The speech summary is determined by the longest repeated pattern. Similarly, [17] creates a cluster of recurring auditory elements. They are then sorted based on the degree of repetition, and a summary is provided as a result.

The paper [18] presents an unsupervised audio summarizing model based on graphs. There are two graph layers here. In one layer, utterances are considered nodes, whereas prosodic characteristics are handled as nodes in the other. The scores of nodes are then determined using a random walk algorithm. The summaries are created based on these scores.

Meanwhile, using a 1-D convolutional neural network, we have proposed a transcription-free supervised speech-to-speech summarization approach in our work. We retrieved the high-level prosodic characteristics from Google's LSTM-based model, which was utilized for speaker verification in [19].

III. Dataset Collection

In this section, we have discussed the dataset used for training the model.

There was no English dataset available to meet our requirements. As a result, we have built our dataset. We needed an audio recording and its summary to serve as ground truth value while training our model. So we began by downloading audiobooks from a YouTube [2] channel narrated by a speaker. The length is around 5 hours. Then we changed it to .WAV format to make it easier to work with. Then we used sentence segmentation to divide the entire audio into parts representing each sentence. The summary was created by hand. A text summarizer is also used to avoid missing any critical portions in summary. Following the creation of the summary, the sentence segments are manually tagged as 0 or 1 and stored as a .CSV file. In this case, 0 indicates that the sentence is not included in the summary, while one indicates that the sentence is included in the summary. We now have a ground truth value and audio input, but we require audio features as input to our model. So Resemblyzer extracts features from each phrase and saves them in.CSV file, which will be utilised while training our model.

IV. Proposed Method

In this section, we have discussed the block architecture (Figure 1) of the summarization system explaining the working of each block. The system consists of the following blocks:

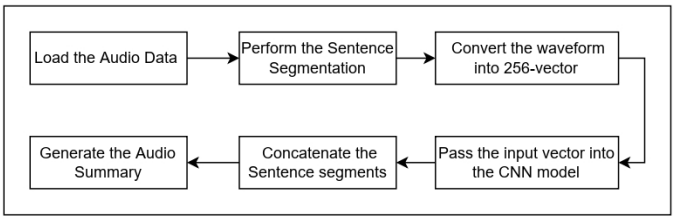

Fig. 1: Proposed Method Architecture

A. Loading the Audio Data

When training our model, the CSV file containing the inputs and matching ground truth values is loaded. When we wish to test our model, we import the audio data into our application to perform the summarization using the pydub package [3].

[1]https://www.fon.hum.uva.nl/praat/

[2]https://www.youtube.com/c/SherlockHolmesStoriesMagpieAudio

[3]https://pypi.org/project/pydub/

B. Perform the Sentence Segmentation

In this section, we divide our audio into little chunks. Each piece will be a sentence. Then we'll deal with each sentence segment audio separately to decide whether or not to include it in our summary.

To execute sentence segmentation, we will consider the fact that we pause while speaking to take a breath before beginning the following sentence. So, at each moment in time, we examine the loudness of the audio stream. Treat the signal as quiet if it goes below our specified level (after accounting for any noise in the recording). Then, if the stillness lasts longer than the set time, designate it as genuine silence. This quiet will serve as a break before starting a new sentence.

In our case, we used 30dB as the silence loudness threshold. This means that anything 30 decibels quieter than average loudness is considered silence, and 1 sec is the silence time threshold. If loudness is lower than the silence loudness threshold for or more than 1 second, we will treat it as the end of the sentence. The process is demonstrated in Figure 2.

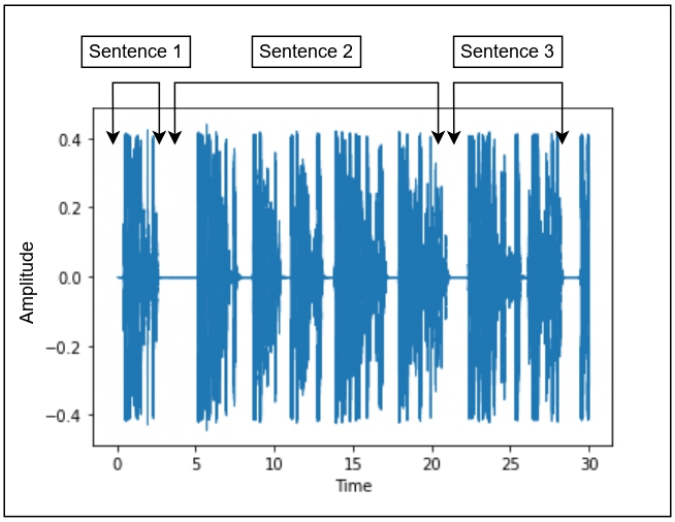

Fig. 2: Sentence Segmentation

C. Convert the Waveform into 256-vector using Resemblyzer

This block uses the Resemblyzer tool for extracting features from our dataset. Resemblyzer uses a deep learning model to generate a sophisticated representation of a voice capturing its prosodic features (the voice encoder). Given a speech audio file, it generates a vector of 256 floating values that encapsulates the characteristics of the voice spoken.

It is a pre-trained deep learning model developed at Google. It was designed to propose a new generalized end-to-end loss function for speaker verification tasks. Two kinds of speaker verification are done - one is text-dependent, and another is text-independent. A 3-layer LSTM [20], [21] with projection has been used in it. The size of the embedding vector is the same as the size of the LSTM projection. They have employed 128 hidden nodes for Text-Dependent Speaker Verification, and the projection size is 64 and 768 hidden nodes with a projection size of 256 for Text Independent Speaker Verification.

As we summarise the audio without any transcript, we have used the later one with the projection of size 256. Thus, we pass the audio to Resemblyzer, and a 256-size vector is obtained that will be used as an input to our CNN model.

D. Passing the input vector into the CNN model

In the past few decades, Deep learning has become a research hotspot in machine learning due to the advent of high-performing GPUs. It has become the most suitable option for performing complex computational and cognitive problems. Its ability to deal with a large amount of data makes it more viable to use. CNN is very good at finding the patterns and gradients in data, making it a more powerful tool for computer vision. Many researchers have successfully shown the applications of CNN in the field of image classification, cybersecurity, bioinformatics, natural language processing, and speech recognition. In this study, we have used the CNN-based model. We have designed our method based on Lenet-5 architecture.

Lenet-5 is a pre-trained model developed mainly for hand-written and printed character recognition. Yann LeCun et al. first published this work in 1998 in the paper Gradient-Based Learning Applied to Document Recognition [22]. It is one of the earliest pre-trained models with an easy and precise architecture. It is a CNN based model having multiple layers specifically designed for image classification. The architecture of Lenet-5 consists of 5 hidden layers, which is why it is called lenet-5. The layer structure consists of 3 batches of the convolutional layer with an average pooling layer followed by the two fully connected layers. The softmax classifier is applied at the output layer to classify the input image into respective classes.

Our input vector is a vector of 256 floating numbers. It is a one-dimensional vector, whereas character recognition deals with two-dimensional data input. The convolutional neural network was primarily designed to work with a massive amount of 2-Dimensional image data (character recognition). The speech dataset is relatively less, making it prone to overfitting. Furthermore, unlike character recognition, speech summarization is a binary classification task (whether to include the sentence in summary or not). Hence, the standard Lenet-5 does not apply to this problem as the feature maps, convolution layer strides, and fully-connected nodes configuration in Lenet-5 were designed for character recognition.

Thus there was a need to make changes accordingly

- Modifying the number of convolutional layers and convolutional layer strides.
- Using a One-dimensional convolutional layer as the data is 1d time series data.
- Addition of a Dropout layer between the fully connected and convolutional layer to avoid overfitting.
- Keeping only one FC layer after the dropout layer.

Figure 3, table I represents the model architecture and the modified LeNet-5 model. The convolutional layer strides of the

new architecture have been updated to two. Standard Lenet- 5 has three convolution layers, but in our proposed model, we have reduced one layer. So, it consists of 2 convolutional layers, two max-pooling layers, followed by two fully connected layers. The number of feature maps is increased in each layer. After the convolution layers, a dropout layer is added with a 0.8 drop rate. The standard LeNet-5 model has ten output layer nodes, but here we need only one as this is a binary classification problem.

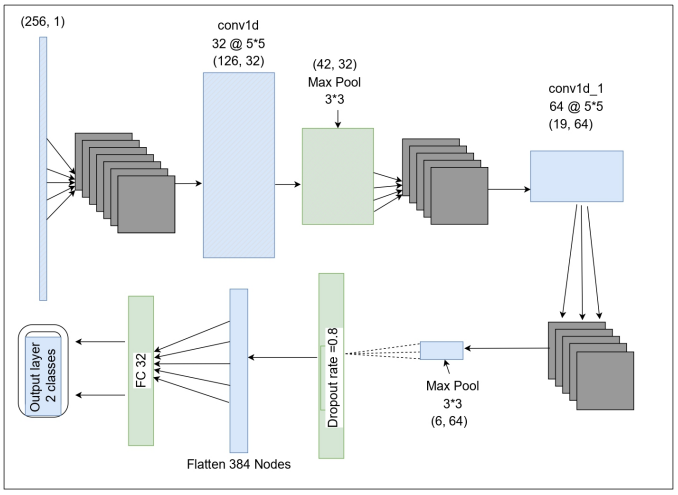

Fig. 3: Architecture of our model

TABLE I: Layer structure and parameter details of our proposed architecture

Layer (type)	Output Shape	Parameters
input	(None, 256, 1)	0
conv1d	(None, 126, 32)	192
max_pooling1d	(None, 42, 32)	0
conv1d_1	(None, 19, 64)	10304
max_pooling1d_1	(None, 6, 64)	0
dropout	(None, 6, 64)	0
flatten	(None, 384)	0
dense	(None, 32)	12320
dense_1	(None, 1)	33

E. Concatenate the sentence segments to form the summary

We assemble the audio pieces into a single audio output file throughout this step. In this case, a metric known as the summarization ratio is applied (c). c is the total number of sentences in our created summary divided by the total number of sentences in the original audio. It will be determined by the user to what degree he wants to summarize. The value will be a real number between 0 and 1. To calculate the number of sentences in the output (k), we multiply the total number of sentences in the original audio by the summarization ratio. After running those audio segments through our trained model, we get a predicted value with a range of [0, 1], where values closer to 0 indicate that they are unlikely to be included in the summary and values closer to 1 indicate that they are highly likely to be included. We create a hashmap with ⟨an audio segment index, its predicted value⟩as a key-value pair. To keep the most relevant parts at the top, we sort this hashmap by values in decreasing order. The top k phrases are then extracted, and the remainder is discarded. To keep the original audio sequence, we sort the residual hashmap by key. The audio segments are then concatenated as per index to make a single output, which is our generated summary.

V. Experimental Setup

This section covers the details of audio data, the parameters used to train the model, and the hardware configuration.

The audio files processed are in WAV format, with a sampling rate of 44 kHz and a resolution of 16 bits. Our dataset contains 5 hours of audio data. 80% of the data is taken for training, while the remainder is used for testing.

The model was trained using 32 batches and a learning rate of 1×10^{-3}. Previously, it was trained for 200 epochs before we noticed overfitting. To avoid overfitting, L2 regularisation and dropout are applied. Overfitting was then minimised to some extent. Then, to appropriately extend our model, we employed early stopping, and the model was eventually trained for 150 epochs.

Google Colab [4] was used for all the experiments done in this work. The system provided has 13GB video memory with Intel(R) Xeon(R) CPU @ 2.20GHz. Our choice was Google Colab because it has many advantages, such as several pre-installed machine-learning libraries, including Keras, TensorFlow, and PyTorch, are available. We can access our notebooks from any device with a simple Google log-in. All of our work is saved in our Google Drive account. It includes a collaboration feature for use when working with multiple developers on a project. It offers dedicated GPUs and TPUs for our machine-learning projects. In projects, GPU and TPU acceleration make a significant difference.

VI. Results

Accuracy, precision, recall, and F1 score are the assessment metrics employed here. The following formulas were used to calculate them:

The accuracy of a model is the proportion of adequately recognised samples among all labeled samples.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

The precision measures how accurate our model is when the anticipated values are positive.

$$Precision = \frac{TP}{TP + FP}$$

A model's recall is how well it can recognise +ve labeled samples.

$$Recall = \frac{TP}{TP + FN}$$

[4]https://colab.research.google.com/

The harmonic mean of precision and recall is used to get the F1 score.

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative. True positive in the context of our job indicates that the statement was intended to be included in the summary and is included. True negative means that the sentence was not included in the summary of ground truth and is not included by our model; false positive means that the ground truth value says it should not be considered in summary, but our model predicted it to be in summary; and false-negative means that our model discarded it from the generated summary, but it was intended to be in summary.

All of our model's performance values are shown in table II. Figures 4 and 5 show our model's loss and accuracy curves while training.

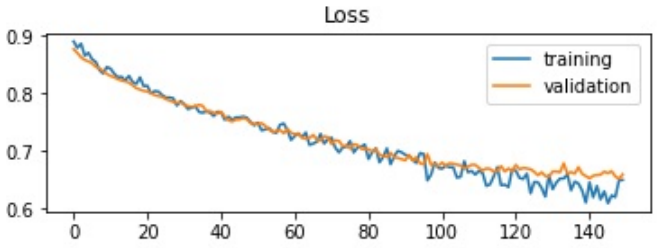

Fig. 4: Training vs Validation Loss

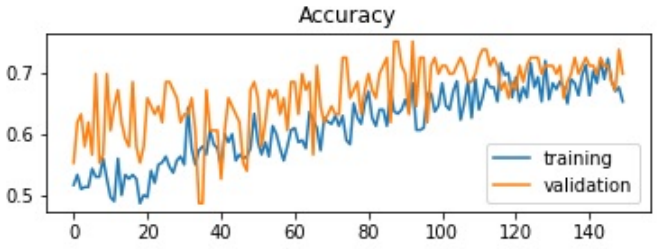

Fig. 5: Training vs Validation Accuracy

TABLE II: Performance Measure of our model

S.No.	Evaluation Metric	Values
1	Accuracy:	0.697368
2	Precision:	0.756757
3	Recall:	0.666667
4	F1 score:	0.708861

We later tested our model with new audio. After segmentation, we had 378 sentences. Our model correctly predicted 263 of them, yielding an accuracy of nearly 0.6957, which is close to our evaluated value. The confusion matrix of our results is depicted in figure 6.

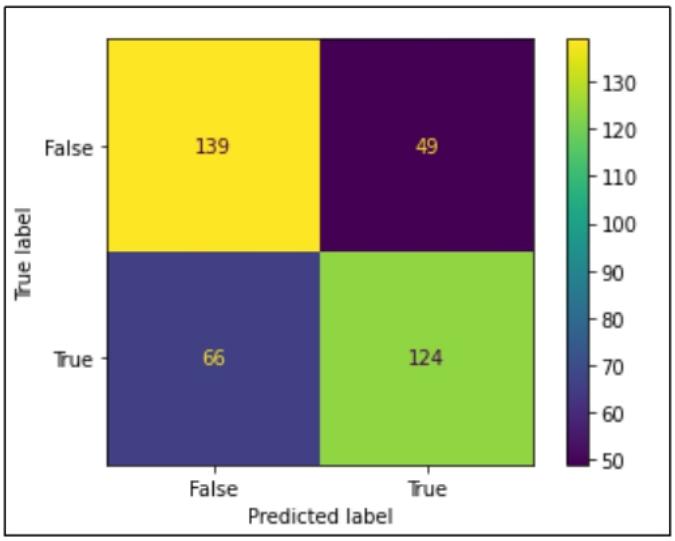

Fig. 6: Confusion Matrix

VII. Conclusions and Future Work

This work proposes a 1-D convolutional neural network model to perform speech summarization without using the transcript. Our model provides promising results in speech summarization. We have used the original audio to produce a summary that retains the natural quality. We have introduced a summarization ratio factor by which the user can set the degree of summarization.

In future, we intend to work on the noisy dataset. It will be further extended to multiple-speaker speech summarization. Right now, we need to set the segmentation parameters for different speakers. More work will be done on sentence segmentation to generalize the process for different cases. By looking at the results obtained, there is a possibility of using prosodic features in other speech-related tasks like speaker verification and speaker diarisation.

References

[1] E. Lloret, Text summarization: an overview, Paper supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01) (2008).

[2] M. Maybury, Advances in automatic text summarization, MIT press, 1999.

[3] E. Hovy, C.-Y. Lin, et al., Automated text summarization in summarist, Advances in automatic text summarization 14 (1999) 81–94.

[4] R. W. Frick, Communicating emotion: The role of prosodic features., Psychological bulletin 97 (3) (1985) 412.

[5] J. Vaissière, Language-independent prosodic features, in: Prosody: Models and measurements, Springer, 1983, pp. 53–66.

[6] S. Maskey, J. Hirschberg, Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization, in: Ninth European Conference on Speech Communication and Technology, 2005.

[7] S. Furui, T. Kikuchi, Y. Shinnaka, C. Hori, Speech-to-text and speech-to-speech summarization of spontaneous speech, IEEE Transactions on Speech and Audio Processing 12 (4) (2004) 401–408.

[8] T. Kemp, T. Schaaf, Estimating confidence using word lattices, in: Fifth European Conference on Speech Communication and Technology, 1997.
[9] C. Hori, S. Furui, Speech summarization: an approach through word extraction and a method for evaluation, IEICE TRANSACTIONS on Information and Systems 87 (1) (2004) 15–25.
[10] A. Inoue, T. Mikami, Y. Yamashita, Improvement of speech summarization using prosodic information, in: Speech Prosody 2004, International Conference, 2004.
[11] K. McKeown, J. Hirschberg, M. Galley, S. Maskey, From text to speech summarization, in: Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., Vol. 5, IEEE, 2005, pp. v–997.
[12] C.-L. Huang, C.-H. Hsieh, C.-H. Wu, Spoken document summarization using acoustic, prosodic and semantic information, in: 2005 IEEE International Conference on Multimedia and Expo, IEEE, 2005, pp. 4–pp.
[13] S. Maskey, J. Hirschberg, Summarizing speech without text using hidden markov models, in: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, 2006, pp. 89–92.
[14] P. Boersma, Praat, a system for doing phonetics by computer, Glot. Int. 5 (9) (2001) 341–345.
[15] L. Rabiner, B. Juang, An introduction to hidden markov models, ieee assp magazine 3 (1) (1986) 4–16.
[16] M. Sert, B. Baykal, A. Yazici, Combining structural analysis and computer vision techniques for automatic speech summarization, in: 2008 Tenth IEEE International Symposium on Multimedia, IEEE, 2008, pp. 515–520.
[17] R. Flamary, X. Anguera, N. Oliver, Spoken wordcloud: Clustering recurrent patterns in speech, in: 2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI), IEEE, 2011, pp. 133–138.
[18] S. K. Jauhar, Y.-N. Chen, F. Metze, Prosody-based unsupervised speech summarization with two-layer mutually reinforced random walk, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp. 648–654.
[19] L. Wan, Q. Wang, A. Papir, I. L. Moreno, Generalized end-to-end loss for speaker verification, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4879–4883.
[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[21] H. Sak, A. W. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling (2014).
[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.