Audio Segmentation for Robust Real-Time Speech Recognition Based on Neural Networks

Micha Wetzel, Matthias Sperber, Alexander Waibel

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
micha.wetzel@student.kit.edu, {matthias.sperber, waibel}@kit.edu

Abstract

Speech that contains multimedia content can pose a serious challenge for real-time automatic speech recognition (ASR) for two reasons: (1) The ASR produces meaningless output, hurting the readability of the transcript. (2) The search space of the ASR is blown up when multimedia content is encountered, resulting in large delays that compromise real-time requirements. This paper introduces a segmenter that aims to remove these problems by detecting music and noise segments in real-time and replacing them with silence. We propose a two step approach, consisting of frame classification and smoothing. First, a classifier detects speech and multimedia on the frame level. In the second step the smoothing algorithm considers the temporal context to prevent rapid class fluctuations. We investigate in frame classification and smoothing settings to obtain an appealing accuracy-latency-tradeoff. The proposed segmenter yields increases the transcript quality of an ASR system by removing on average 39 % of the errors caused by non-speech in the audio stream, while maintaining a real-time applicable delay of 270 milliseconds.

1. Introduction

Automatic speech recognition (ASR) systems generally respond poorly to music and noise input. The recognition delay increases due to a missing likely interpretation of the audio blowing up the search space. The number of insertions increases in those segments due to falsely detected speech. Filtering those segments out is desirable to increase the accuracy of the ASR and to improve the average recognition speed. This is especially important for real-time systems, where a small delay is required, and processing power is generally limited. According to our experience, a delay of less than 3 seconds is desirable in a real-time environment, although previous work has assumed up to 5 seconds [2].

The drawback in existing segmentation models for speech / non-speech classes is that they induce a high ($>$ 1 s) latency and are not evaluated and optimized as a preprocessing step for ASR systems. Usually only the accuracy of classifying single frames with some additional temporal context is investigated [1, 7, 5]. El-Maleh et al. reported an accuracy of 96 % when classifying audio frames of size 1 second using a quadratic Gaussian classifier, features based on Line Spectral Frequencies and higher order crossings and a simple smoothing algorithm [4]. Panagiotakis et al. predicted segment changes with an accuracy of 97 %, mostly within a 0.2 second interval. Classifying the resulting segments yielded a classification accuracy of 95 %. The latency caused is 3 seconds [6].

This work proposes and evaluates a segmentation algorithm to discriminate speech, music and noise from an audio stream, which can be used as a preprocessing step for an online ASR. In this context online refers to stream decoding with real-time requirements. Music and noise segments are replaced by silence and therefore the ASR does not need to spend valuable computation time on those segments, and does not produce wrong transcripts for those music and noise segments. As the ASR may already need a few seconds to create the transcript, the delay caused by the segmentation algorithm needs to be small (e.g. <0.5 seconds) in order to satisfy the real-time constraint.

The segmentation uses a two step approach, consisting of classification and smoothing. A multilayer perceptron is used to classify audio frames. Features based on Mel frequency cepstral coefficients and the zero-crossing rate are used as input for the classification. Different model parameters, as well as feature extraction parameters, such as frame context and number of MFCCs are evaluated regarding accuracy and induced latency.

The second step consists of a smoothing algorithm, which smoothes the classified frames to create segments of certain audio types and removes small misclassifications. Different smoothing parameters are compared. This step is necessary as using an imperfect classifier can actually lead to a decrease in transcript quality. The transcript quality is measured by the word error rate (WER). Music and noise are treated separately to increase flexibility regarding smoothing parameters.

To train and evaluate the neural network the MUSAN [8] dataset is used, which is a publicly available audio dataset containing music, speech and noise. The end-to-end performance is evaluated by comparing the resulting transcript quality (1) of the ASR as-is, (2) of the ASR when non-speech is removed manually, and (3) when the ASR uses the segmenter as preprocessing step, replacing music and noise with silence.

The experiments show that a small neural network can classify 10 ms audio frames from a live audio stream with an accuracy of 87%, while maintaining a latency of 70 ms. Using the proposed segmentation framework the end-to-end ASR performance could be increased by correcting on average 39 %(with a σ of 27 %) of the errors caused by non-speech in the audio stream. The latency caused by the segmenter is 270 ms.

2. Proposed Framework

The proposed segmentation framework acts as a preprocessing step for the automatic speech recognition system. It consists of three steps: Feature extraction, classification and smoothing, as seen in Figure 1. In the first step the audio stream is split into small frames from which feature vectors are extracted. In the second step a neural net classifies the feature vectors into speech, music or noise. The classification is then smoothed to avoid small misclassifications and to represent the high likelihood of adjacent frames having the same class. Every audio frame which has not been classified as speech is then replaced by silence. The resulting audio stream is fed to the ASR. This design allows the segmenter to be used as preprocessing step for an arbitrary ASR, creating a small delay.

2.1. Feature Extraction

Spectral and temporal features are extracted in this step. The audio stream is split into 10 ms frames $a_0, \cdots, a_n$, from which feature vectors $p_0, \cdots, p_n$ are extracted. N_{mel} (we choose 20) Mel frequency cepstral coefficients (MFCC) and the zero-crossing rate (ZCR) are the features used in our system.

For each feature vector p_i in each temporal direction C_f (referred to as *frame context*, we choose 6) adjacent vectors $(p_{i-C_f}, \cdots, p_{i+C_f})$ are added to obtain more temporal information. The mean, standard deviation and variance of those $2 \cdot C_f + 1$ vectors are used as the final feature vector f_i. Using the statistic functions reduces the dimensionality from $(2 \cdot C_f + 1) \cdot 21$ to $3 \cdot 21 = 63$.

2.2. Classification

The proposed neural net architecture is a small multilayer perceptron with three hidden layers and $H_1 \times H_2 \times H_3$ neurons (we choose $30 \times 20 \times 10$). The output layer consists of 3 neurons, one for each audio class (speech, music and noise). Each neuron uses the sigmoid function as activation function. The softmax function is used to convert the output of the neural net into class probabilities. The predicted label k_i of the frame a_i is the class with the highest probability.

2.3. Smoothing

In this section we propose a smoothing algorithm based on two steps, *Mode Smoothing* and *Minimum Change Support*. In the first step a new classification label l_i of the audio frame a_i is calculated by taking the mode (the most frequent occurring value) of the adjacent labels $k_{i-C_m}, \cdots, k_{i+C_m}$, where C_m is the *mode context* (we choose 20).

$$l_i = mode(k_{i-C_m}, \cdots, k_{i+C_m}) \tag{1}$$

The second step - Minimum Change Support - potentially prevents fast reoccurring class changes by a simple rule: The final class label c_i is c_{i-1}, unless at least half of the previous min_s labels (we choose 300) is l_i or l_i is speech.

$$c_i = \begin{cases} l_i & |\{l_j \mid i - min_s \leq j \leq i \wedge l_j = l_i\}| \geq \frac{1}{2} min_s \\ l_i & l_i = \text{speech} \\ c_{i-1} & \text{otherwise} \end{cases} \tag{2}$$

3. Experiments

We evaluated different parts of the segmentation algorithm. The most notable results are presented in this section. Since the delay of the segmenter is added to the delay of the ASR we focus on maintaining a small inherent delay (delay that depends on how much temporal information is used during feature extraction). We found the computational delay negligible compared to the inherent delay, and do not provide a systematic evaluation since it depends on hardware and implementation details.

3.1. Classification Experiments

The neural network for classification has been trained and evaluated with the MUSAN corpus [8], which contains over 100 hours of labeled speech, music and noise from various audio sources. We choose to use this vast corpus since it is publicly available and its audio files are under the Creative Commons license / in the US Public domain. Only a subset of the corpus is used in order to have an equal prior probability for each class. We split the corpus into a training (80%), validation (10%) and test (10%) set, making sure that an audio file is not split into multiple sets. The training set has been used to train the neural network. We used the validation set to compare the performance of different architecture settings. The best architecture, based on the validation set, has then been evaluated with the test set.

Batch stochastic gradient descent has been used to train the neural network. Multiple values for the number of neurons in the three hidden layer have been tested, and $30 \times 20 \times 10$ was the lowest setting while maintaining a good accuracy. This small net architecture yields a small runtime of less than 1 ms per classification on a 2.4 Ghz single core CPU. With this setting a frame classification accuracy of 87 % has been

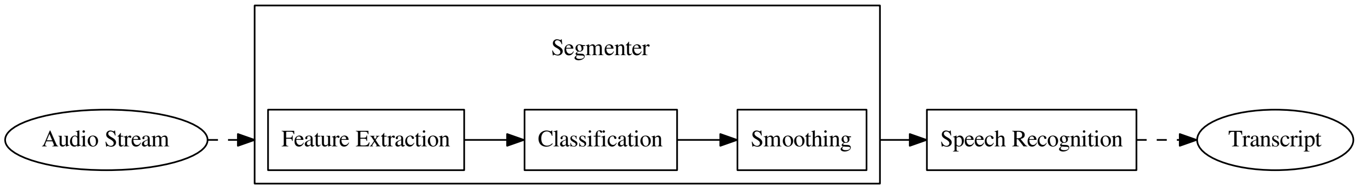

Figure 1: This figure shows the processing steps from audio to transcript. The audio stream is fed to the segmenter. The segmenter extracts audio features, classifies those feature vectors, smoothes the classification and then replaces non-speech segments with silence. The resulting audio stream is fed to the ASR which produces a transcript.

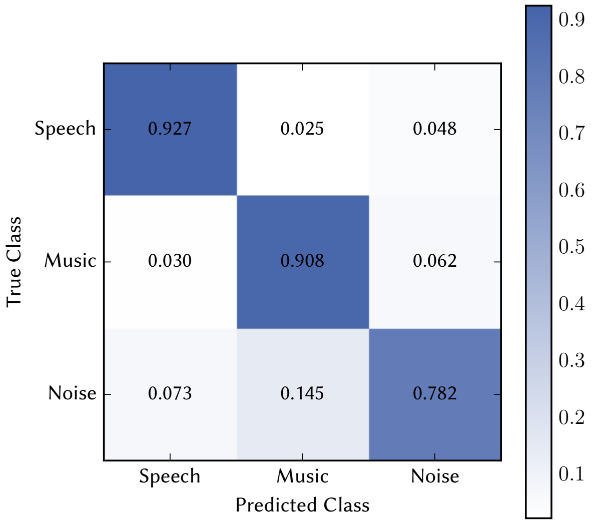

Figure 2: Confusion matrix of the accuracy of the neural network tested and trained with the MUSAN corpus.

be achieved on the test set. The corresponding heat map can be seen in Figure 2. Using the trained network to classify between speech and music, a frame classification accuracy of 95% has been achieved.

3.2. Feature Extraction Experiments

In Table 1 the classification accuracy can be seen for different values of N_{mel}. It can be seen that less than 20 MFCCs hurt the accuracy, while more than 20 MFCCs only increase the feature dimensionality, but not the accuracy.

We also investigated how adding more temporal information by increasing the frame context C_f (and thereby the inherent delay) affects the accuracy. In Figure 3 the results can be seen. Instead of the frame context the inherent delay caused by the feature extraction is shown, as the delay is of more interest. The inherent delay of the feature extraction is calculated by:

$$d_{feature} = (C_f + 1) * 10 \text{ ms} \tag{3}$$

Additionally Figure 3 compares using MFCCs as the only feature to MFCCs and the ZCR as features. It can be seen

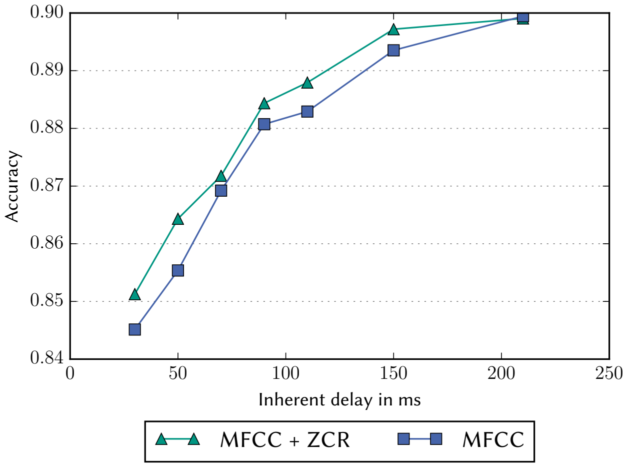

Figure 3: This figure shows the accuracy of the neural network using MFCCs as the only features versus using MFCCs and the ZCR for multiple inherent delays.

Accuracy (%)	81.5	84.6	86.2	87.0	86.9	87.2
Number of MFCCs N_{mel}	5	10	15	20	30	40

Table 1: This Table shows the accuracy of the neural network for different numbers of Mel frequency cepstral coefficients.

that there is a clear tradeoff between accuracy and inherent delay.

3.3. Smoothing Experiments

Preliminary tests have shown that a mode context C_m of 20, and a min_s of 300 provide good results. To optimize those values out of domain data has been used, which differs from the test data used in the next section.

3.4. Segmentation Experiments

To evaluate the segmenter we measure the transcript quality (measured in the word error rate (WER)) of multiple TED Talks[1]. The talks contain some music and speech, which is most of the time clearly separated. In Table 2 the name of the

[1] www.ted.com

talks can be seen. Figure 6 shows the manual segmentation of the talks. The ASR is based on Janus [3]. It is a TED-based GMM system with deep bottleneck features. A 4gram language model is used which is trained on the IWSLT training data. The system contains a garbage and silence model but no music model. The vocabulary consists of 150k words. We compare three different setups for the ASR:

No segmentation The audio is directly fed to the ASR, no preprocessing step is done.

Automatic segmentation The audio is fed to our segmenter, the resulting audio, where non-speech is replaced with silence, is fed to the ASR.

Manual segmentation The audio is segmented manually, all non-speech segments are replaced with silence. The segmented audio is fed to the ASR.

The gold standard is the manual segmentation, to which we compare our segmenter. Although there is no guarantee for the gold standard to be optimal in terms of WER. Additionally the transcript quality should not be lower than without any segmentation, since this would mean that the segmenter worsens the transcript quality. To evaluate this efficiently we introduce the Rate of Resolved Segmentation Errors.

Let O be the WER when no segmentation is used, F the WER when our segmenter is used, and M the WER when manual segmentation is used. The *Rate of Resolved Segmentation Errors* is then defined as:

$$RRSE = 1 - \frac{(F - M)}{(O - M)} \tag{4}$$

The RRSE has the property that a value of 1 means the segmenter is as good as manual segmentation, and a value of 0 means that the segmenter is as good as no segmentation. It can be interpreted as the percentage of segmentation errors that are resolved, from the total amount of segmentation errors that can be resolved by manual segmentation. Note that a value above 1 and below 0 is possible. A value below 0 would mean that the segmenter leads to a worse performance. To achieve robustness this must be avoided.

The raw results can be seen in Figure 4. The corresponding RRSE can be seen in Figure 5. The average RRSE of all tests is 0.389 with a standard deviation of 0.270. No test has a RRSE below 0. The inherent delay of the Mode Smoothing is 200 milliseconds, which puts the total inherent delay to 270 milliseconds.

4. Prior Work

In this section we present prior work that investigates the classification and segmentation of audio.

Harb and Chen propose an algorithm to discriminate 30 ms frames into speech and music. First order statistics of MFCC feature vectors within a 0.2 second window are used

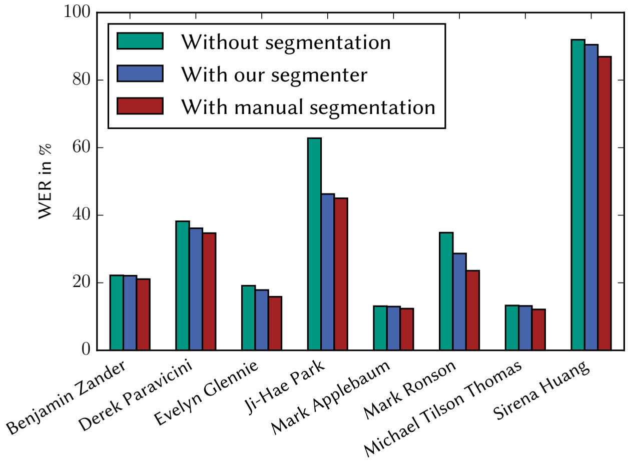

Figure 4: In this figure the transcript quality for different ted talks can be seen. We compare our segmenter to no segmentation and manual segmentation. Non-speech segments are replaced with silence.

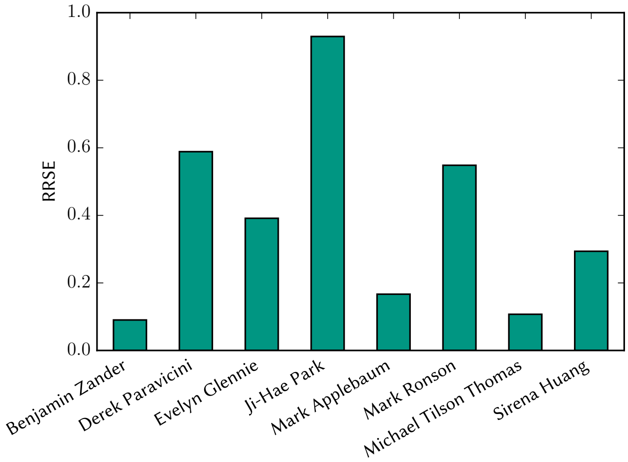

Figure 5: In this figure the Rate of Resolved Segmentation Errors (RRSE) that our segmenter achieves for multiple ted talks can be seen. A RRSE of 1.0 means that the segmenter is on-par with manual segmentation, while a RRSE of 0.0 means that the segmentation did not improve the transcript quality.

Speaker	Name of the TED Talk	Type of music	Distinctness, overlap of music and speech
Benjamin Zander	The transformative power of classical music	Piano	Occasionally talking while playing
Derek Paravicini	In the key of genius	Piano	Good distinction
Evelyn Glennie	How to truly listen	Xylophone, drum	Good distinction
Ji-Hae Park	The violin, and my dark night of the soul	Violin, piano, drum	Good distinction
Mark Applebaum	The mad scientist of music	Piano, Strange sounds (sound like noise)	Good distinction
Mark Ronson	How sampling transformed music	Synthesizer, sampling, modern music	Hard to distinguish, remixed speech as music
Michael Tilson Thomas	Music and emotion through time	Piano, Singing, Choir	Many overlaps of piano/choir and speech
Sirena Huang	An 11-year-old's magical violin	Violin, piano	Clear distinction, bad audio quality

Table 2: This table lists the TED Talks used for the evaluation. The type of music occurring in the talks is shown, as well as how good the music and speech is distinguishable and how much it overlaps.

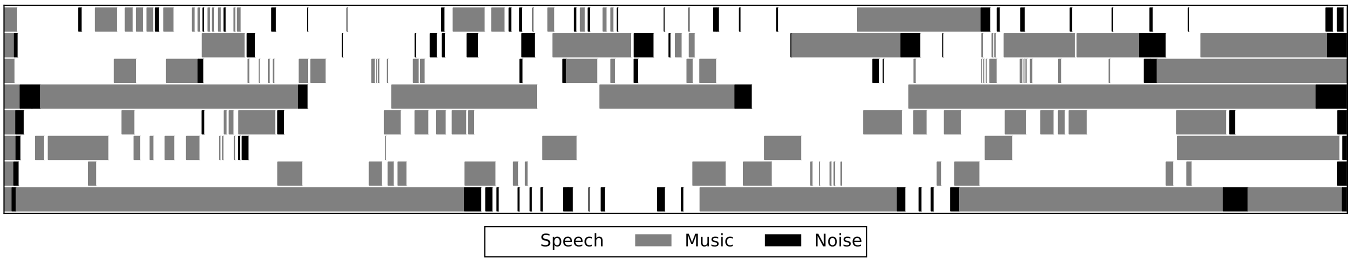

Figure 6: The different audio segments for each evaluation TED Talk can be seen in this figure. Each row represents one talk. The talks are in the same order as in Table 2.

as features. Using a neural network an classification accuracy of 93 % has been achieved. Pikrakis et al. use Restricted Bolzmann Machines to classify speech and music. The features are based upon the spectogram of the signal and MFCCs. They are extracted from three 50 ms frames. An accuracy of 92 % has been achieved, and when using a confidence threshold of 0.9 the accuracy could be increased to 96 % with 10 % unclassified frames [7]. In contrast to our work they achieved a higher classification accuracy. However they discriminated only between two classes (speech, music) while we discriminate between speech, music and noise. As Figure 2 shows our classifier performs similar in discriminating speech and music, while having a lower latency than their classifier models.

El-Maleh et al. proposed a real-time algorithm to discriminate speech and music using a quadratic Gaussian classifier, classifying feature vectors based on Line Spectral Frequencies and higher order crossings. Using an audio frame size of 20 ms and smoothing the classifications with the knowledge of the two preceding frames an accuracy of 78% has been achieved. Without the smoothing the accuracy decreased by 5-10%. Since no information of the succeeding frames is needed the latency is only 20 ms. When classifying 50 frames combined as a 1 second window, thus increasing the latency to 1 second, an accuracy of 96% has been reported [4] Comparing really short latencies, we achieved an accuracy of 85.1% with a latency of 30 ms.

In contrast to most previous work Panagiotakis et al. proposes an method to predict segment changes first and then classify those segments. The segments had a minimum size of 1 second and the accuracy of the change instant was mostly within an interval of 0.2 seconds. The segment changes have been predicted with an accuracy of 97% and the segments have been classified correctly into speech, music or silence with an accuracy of 95% [6]. This is one of the few works focusing in creating bigger segments of speech and non-speech. But since a segment is only classified when a new segment begins this method is not suited for real-time segmentation without modification. Additionally the segmentation algorithm causes a 3 second delay, which we consider too large to be applicable as a preprocessing step for an online ASR.

5. Conclusions

In this work a system to increase speech recognition transcript quality by filtering out music and noise segments from the audio stream has been developed and evaluated, while focusing on a low delay to maintain real-time capability. In contrast to previous work, we evaluated our system in terms of the actual transcript quality. We used a neural network to classify audio frames into speech, music and noise based on Mel frequency cepstral coefficients and zero-crossing rate features. The feature space is reduced by taking statistics over multiple frames. We optimized multiple feature extraction parameters with the aim of a good transcript quality while maintaining a small delay. By using the extensive publicly available audio dataset MUSAN the classification results are comparable to possible future work. On this dataset an accuracy of 87 % could be achieved with a real-time delay of 70 ms.

In addition to the classification step, we proposed a smoothing algorithm, which removes small misclassifications and creates smooth audio segments.

The experiment results have shown that the segmentation algorithm is able to increase the ASR performance by removing on average 39 % of the errors that can be resolved by segmentation. The standard deviation is quite high (27%), indicating that the performance gain depends a lot on the type of audio. Using the algorithm in a real-time environment creates an acceptable delay of 270 ms, plus a negligible computational delay due to its small neural network architecture and computationally cheap smoothing algorithm.

Using an audio segmenter as preprocessing step for an ASR has shown to be a promising way to increase the performance of online ASR systems. To further increase the performance of the segmenter to be on-par with manual segmentation while reducing the latency the classification and smoothing model can be improved. As seen in previous work, different features can be explored to possibly increase the classifier performance. However, the results of our experiments indicate that the smoothing makes up for a classifier that is not perfect and it seems that the task of improving the smoothing is more promising.

One way the smoothing could improved is by increasing the start boundary of the speech segment. The beginning of a speech segment could be shifted by a small amount, to make sure that the start of the speech is included in the segment. This could be achieved by sending the ASR the previous frames when a change to speech is detected by the segmenter. As the ASR would receive a short burst of additional audio it would shortly have a higher delay.

Another possibility would be to run the segmenter not as a preprocessing step, but alongside the ASR, sending class changes to the ASR along with the time stamp when the change occurs. The ASR would then need the capability to remove the last seconds of the transcript when a change to non-speech is sent with a time stamp from the past. This method would allow the segmenter to have a higher latency generally resulting in a better segmentation quality.

References

[1] H. Harb and Liming Chen. "Robust speech music discrimination using spectrum's first order statistics and neural networks". In: *Proceedings of the 7th International Symposium on Signal Processing and Its Applications*. Vol. 2. 2003, pp. 125–128.

[2] Walter Lasecki et al. "Real-time captioning by groups of non-experts". In: *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012, pp. 23–34.

[3] Alon Lavie et al. "JANUS-III: Speech-to-speech translation in multiple languages". In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. IEEE. 1997, pp. 99–102.

[4] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. "Speech/music discrimination for multimedia applications". In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. Vol. 6. 2000, pp. 2445–2448.

[5] Martin F McKinney and Jeroen Breebaart. "Features for audio and music classification." In: *ISMIR*. Vol. 3. 2003, pp. 151–158.

[6] C. Panagiotakis and G. Tziritas. "A speech/music discriminator based on RMS and zero-crossings". In: *IEEE Transactions on Multimedia* 7.1 (2005), pp. 155–166.

[7] Aggelos Pikrakis and Sergios Theodoridis. "Speech-music discrimination: A deep learning perspective". In: *Proceedings of the 22nd European Signal Processing Conference*. IEEE. 2014, pp. 616–620.

[8] David Snyder, Guoguo Chen, and Daniel Povey. "MUSAN: A Music, Speech, and Noise Corpus". In: *Computing Research Repository* abs/1510.08484 (2015).