EE798P Assignment 4

Adit Jain, 200038

I. Introduction

Speech segmentation is a critical task in various fields such as automatic speech recognition, speaker diarization, and audio indexing. Accurate detection of speech segments within a given sound file is essential for extracting meaningful information from audio data.

The objective of this research is to develop a machine learning model capable of accurately detecting speech segments in sound files. I'm going to evaluate the performance of our model based on segment and event-wise precision, recall, and F1-score.

II. Method Description

A. Feature Extraction

In the initial phase of my preprocessing pipeline, I focused on extracting discriminative features from the raw audio data. For this purpose, I employed Mel-frequency cepstral coefficients (MFCCs), a widely accepted representation for speech processing tasks. MFCCs effectively capture the spectral characteristics of speech signals, making them invaluable for distinguishing between speech and non-speech segments.

The process begins by segmenting the audio data into short, overlapping frames, typically spanning 20-30 milliseconds. Then I compute the Fourier Transform for each frame, followed by applying a bank of Mel filters to obtain the Mel-frequency spectrum. The logarithm of the resulting spectrum is taken to approximate the human auditory system's response to different frequencies. Finally, a Discrete Cosine Transform (DCT) is applied to obtain a compact representation of the spectrum in the cepstral domain. This yields a set of MFCC coefficients, which are used as feature vectors for training my model.

B. Labelling

Accurate labeling of the training data is paramount for training a robust machine learning model. To extract the data from the *.TextGrid* files, I wrote an elementary python code to extract the array of intervals (segments) corresponding to each audio file's 20-30 milliseconds spanning segments

C. Data Augmentation

To enhance the diversity of our training data and improve the model's ability to generalize, I'll employ data augmentation techniques. These perturbations are applied in the time domain to simulate variations in speech speed and pitch that may occur in real-world scenarios. Time-stretching involves modifying the duration of the audio while maintaining its pitch, effectively simulating faster or slower speech. Pitch-shifting, on the other hand, alters the pitch without changing the duration. By applying these random transformations to the training samples, a more robust and adaptable model is created. This would essentially account for the differences in the mean pitch of female and male speech.

D. Model Architecture

In choosing an appropriate architecture for our speech segment detection task, I considered the temporal nature of audio data. While Convolutional Neural Networks (CNNs) have excelled in image processing tasks, they may not capture temporal dependencies as effectively. On the other hand, Recurrent Neural Networks (RNNs) are well-suited for sequential data, but they might struggle with long-range dependencies.

Given these considerations, I opted for a Long Short-Term Memory (LSTM) network. LSTMs are a type of RNN that excel at capturing long-term dependencies in sequential data. This characteristic aligns with the nature of speech, which often exhibits complex temporal patterns.

My LSTM-based architecture will consist of multiple recurrent layers with dropout regularization (of around 15-20%). The recurrent layers allow the network to maintain memory of past inputs, facilitating the understanding of context over time. Dropout is applied to mitigate overfitting, ensuring that the model generalizes well to unseen data.

The final layers of the network include fully connected layers, which enable the model to make binary predictions at each time step. The sigmoid activation function is used to produce probabilities indicating whether a given time step corresponds to a speech segment.

E. Model Training

I intend to use binary cross-entropy loss as my objective function. This is well-suited for binary classification tasks, as it measures the discrepancy between predicted probabilities and ground truth labels. To optimize the model, I'm going to use the Adam optimizer with a learning rate of 0.001. Adam adapts the learning rate during training, which can lead to more stable and efficient convergence.

During training, I'll feed the preprocessed audio data (in the form of MFCC coefficients) along with their corresponding ground truth labels into the LSTM network. The model then iteratively adjusts its weights through backpropagation, minimizing the loss function. This process continues for a predetermined number of epochs. I don't want the number of epochs to be too high (which will lead to over-fitting) or too low (which will lead to under-fitting).

TO find the optimal hyper-parameters for my model, I'll use conventional GridSearchCV algorithms, provided in-built by the scikit-learn library. This will take some time but will be indisposable to fine-tune the machine learning model.

F. Scope for using multiple models in tandem

While the LSTM-based architecture shows effectiveness in capturing temporal dependencies for speech segment detection, there may be scenarios where combining different models can yield even better results. One potential approach is to employ a two-step process, utilizing a CNN for initial feature extraction followed by an LSTM for sequence analysis.

The CNN could be employed to extract lower-level features from the MFCCs, potentially uncovering hierarchical patterns in the data. The output from the CNN can then be fed into the LSTM, which excels in capturing longer-range dependencies and contextual information.

This hybrid approach could potentially leverage the strengths of both architectures, leading to a more robust and accurate speech segment detection system. However, it would require careful integration and fine-tuning to ensure optimal performance. Further research and experimentation would be needed to validate the effectiveness of this combined approach.

G. Predictions

For an unseen audio file, I'll first dissect the data into small 20-30 milliseconds overlapping chunks of audio segments, passing each through the model that'll provide me a boolean value, indicating whether or not there is human speech in the given segment. If there is 4-5 contiguous segments where no human speech is detected, that'll be the start of the silence segment and as soon as I have 2 contiguous segments where human speech is detected, it'll mark the start of the next human speech major segment.

III. Evaluation Metrics

Selecting appropriate evaluation metrics is crucial for assessing the performance of our speech segment detection model. I chose segment and event-wise precision, recall, and F1-score for several reasons:

A. Segment-wise metrics

Precision measures the accuracy of positive predictions made by the model. In the context of speech segment detection, it indicates the proportion of predicted speech segments that are actually true speech segments. High precision is crucial to ensure that the detected segments are accurate and reliable. Recall assesses the model's ability to identify all actual positive instances. In our case, it measures the proportion of true speech segments that were correctly identified by the model. High recall is vital to prevent missing important speech segments. The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance, particularly in situations where both precision and recall are important. This metric is particularly useful in our task, where both false positives and false negatives can have significant implications.

B. Event-wise metrics

In addition to segment-wise metrics, I'll also consider event-wise evaluation. This involves evaluating the entire predicted sequence as a single event, which can be critical for tasks where the temporal sequence of events is essential.

C. Reasons for choosing these metrics

I chose these metrics based on their relevance to the speech segment detection task and their ability to provide a comprehensive evaluation of the model's performance. Other metrics like accuracy, AUC-ROC, and Mean Absolute Error (MAE) are not well-suited for this task, as they do not account for the temporal nature of the data. Furthermore, precision, recall, and F1-score are commonly used in similar tasks in the field of speech processing and machine learning. This allows for meaningful comparisons with existing literature and facilitates a better understanding of our model's performance in the broader context of speech processing applications.

IV. Discussion

In this section, I present the salient points of my model:

- One of the key strengths of my approach lies in the rigorous application of data augmentation techniques. By introducing variations in speech speed, pitch, and background noise, it is possible to emulate real-world conditions more accurately. This not only improves the model's robustness to different acoustic environments but also aids in mitigating overfitting. Traditional methods often rely on pristine, controlled datasets, which may not adequately represent the variability encountered.
- While my LSTM-based architecture serves as the backbone of our approach, there is room for future exploration of hybrid models. Combining the strengths of different architectures, such as incorporating a Convolutional Neural Network (CNN) for initial feature extraction, may yield further performance improvements. This approach would leverage the hierarchical feature learning capabilities of CNNs alongside the sequence modeling strengths of LSTMs, potentially leading to even more accurate speech segment detection.

References

[1] Audio Segmentation Techniques and Applications Based on Deep Learning, G. Aggarwal, Vasukidevi G., S. Selvakanmani, B. Pant, K. Kaur, A. Verma, 2022/7994191.

[2] McFee, Brian Raffel, Colin Liang, Dawen Ellis, Daniel Mcvicar, Matt Battenberg, Eric Nieto, Oriol. (2015). librosa: Audio and Music Signal Analysis in Python. 18-24. 10.25080/Majora-7b98e3ed-003.

[3] Mesaros, A., Heittola, T., Virtanen, T., Plumbley, M. D. (2021). Sound event detection: A tutorial. IEEE Signal Processing Magazine, 38(5), 67-83.