SpeakSieve: An Audio Filtering and Information Extraction Service

Akshita Gupta[1], Charvi Jindal[2], Chetan[3], Jogith S. Chandran[4], Shivam Agrawal[5], and Utkarsh Arora[6]

[1-6]Indraprastha Institute of Information Technology, Delhi

Abstract

Extracting information from audio files can be a time-consuming and tiring process, requiring a series of steps such as noise clearing, speaker separation, and transcription. While there are tools available to assist in this process, they often require experience to use effectively and may be paid or too complex for quick work. This project aims to ease the process of information retrieval from audio files by combining various steps and providing user-friendly features. By doing so, this project seeks to make audio information retrieval accessible to everyone.

1 Introduction

1.1 Problem Statement

The project aims to address the challenge of audio editing, which is a laborious process involving several steps such as noise reduction, speaker separation, transcription, indexing, and overlapping. These steps require expertise and experience in using specialized tools, which may not be available to all users. Furthermore, extracting contextual information, topics, and speakers from audio files requires listening to the entire recording, which is time-consuming and cumbersome. Despite the widespread use of audio as a preferred mode of communication and data sharing, there is a lack of user-friendly tools to edit audio files and extract specific information. The project seeks to develop an easy-to-use tool to edit audio files and enable users to extract desired information, such as speaker separation and transcription, without the need for specialized expertise or knowledge.

1.2 Motivation

In today's world, audio has become an increasingly prevalent format for communication and data sharing. Lectures, courtroom proceedings, political speeches, movie dialogues, and confidential messages are just a few examples of the types of information that are often shared in audio format. However, in order to extract specific information from these recordings, such as selecting certain speakers, creating notes, censoring sensitive content, or extracting specific dialogues, a person must possess a certain level of technical proficiency or be willing to navigate cumbersome editing processes. Our research aims to address these challenges by developing a user-friendly tool that simplifies these tasks for individuals with varying levels of technical expertise.

1.3 Novelty

In the current landscape of speech processing, most methods for transcribing and analyzing speech involve a manual process that requires significant time and effort from the user. Our proposed method, on the other hand, is fully automatic, which saves time and effort while increasing efficiency. Additionally, current methods typically involve multiple steps, with different software tools being used for different aspects of speech processing, such as transcription, speaker separation, and noise reduction. This lack of integration results in inconvenience and inefficiency. In contrast, our method integrates all of these features into a single platform, streamlining the process and making it more convenient for users.

Furthermore, the resource-intensive nature of current speech processing methods places a significant burden on the client's machine, often requiring high-end hardware to perform tasks such as speaker separation and noise reduction. Our proposed method circumvents this issue by performing these tasks on the server's machine, significantly reducing the resource requirements on the client's end. This not only increases the accessibility of speech processing tools but also allows for a wider range of users to access these tools without the need for expensive hardware.

Our web tool enables the user to perform the

following tasks:

1. Censor specific phrases from the audio
2. Extract certain words or phrases
3. Extract monologues of a specific speaker

2 Literature Review

1. This landmark paper in the field of speaker separation proposed a deep learning-based method called deep clustering for separating speech signals of multiple speakers in a single-channel audio recording. The study reports that the proposed method outperformed existing techniques for single-channel multi-speaker separation on two datasets. (Isik et al., 2016)

2. Various methods of denoising non-stationary audio sources exist, including digital signal processing and deep learning algorithms. For our purposes, since transcription accuracy is decent, digital signals processing-based algorithms like Spectral Subtraction and Wiener Filtering will suffice. Wiener Filtering requires prior knowledge of essential statistical properties of the signal and noise but can be adapted using Recursive Least Squares or Least Mean Squares algorithms. Deep learning-based methods are more accurate but computationally expensive. (Roux and Vincent, 2013)

3. The paper proposes a speech recognition system, Whisper AI, based on the GPT architecture. The model uses encoding and decoding layers to process audio signals and generate transcriptions. The system utilizes a specialized loss function, called permutation invariant training (PIT), to optimize performance without the need for self-supervision and self-training techniques. The resulting models generalize well and are competitive with prior fully supervised results, approaching the accuracy and robustness of humans.(Radford et al., 2023)

3 Methodology

3.1 Models

Speaker separation and information retrieval from audio require converting the audio signal into transcripts, extracting information on timestamps and speakers, and applying this information to the audio. While there are multiple transcription models available, they often suffer from trade-offs between speed and accuracy.

Whisper, developed by OpenAI, is a prominent speech recognition system that utilizes transformer-based models trained on 680,000 hours of multilingual and multitask supervised data obtained from the web. (OpenAI, 2023) It features five model sizes, four of which have English-only versions, providing a range of speed and accuracy tradeoffs. While the tiny model runs the fastest, it sacrifices accuracy, while the large model offers the best performance but runs slower. In our implementation, the user can select the desired model to use.

Although Whisper provides highly accurate transcriptions, it generates timestamps at the utterance level rather than per word, which may have a margin of inaccuracy that ranges from several milliseconds to seconds. In order to overcome this issue, we employed WhisperX, a refined version of Whisper that utilizes forced alignment with phoneme-based Automatic Speech Recognition (ASR) models and Voice Activity Detection (VAD) preprocessing to improve the accuracy of timestamps. Figure 1 demonstrates the architecture of the WhisperX model. (Bain et al., 2023)

The use of Whisper for audio transcription is limited in that it is unable to identify individual speakers in a conversation. This limitation is particularly problematic for conversational analysis. To address this limitation, diarization techniques are employed to identify speakers in a conversation. In our implementation, we utilized Agglomerative Clustering based on an embedding pre-trained on the Vox Celeb dataset from the SpeechBrain model for diarization. This approach takes the audio file and corresponding transcription as inputs and produces embeddings for each snippet of the transcript based on the speaker's voice in the audio.

3.2 Web Interface

The web interface for our system is implemented using React JS and React Bootstrap, and is hosted locally. Users can input audio files and short text for further analysis. The website allows the user to choose the desired model parameters, and the request payload is passed to the backend using FastAPI. To ensure fast and efficient processing, we have implemented the server for FastAPI using Uvicorn, a fast ASGI (Asynchronous Server Gateway Interface) server. This allows for quick processing and retrieval of the requested informa-

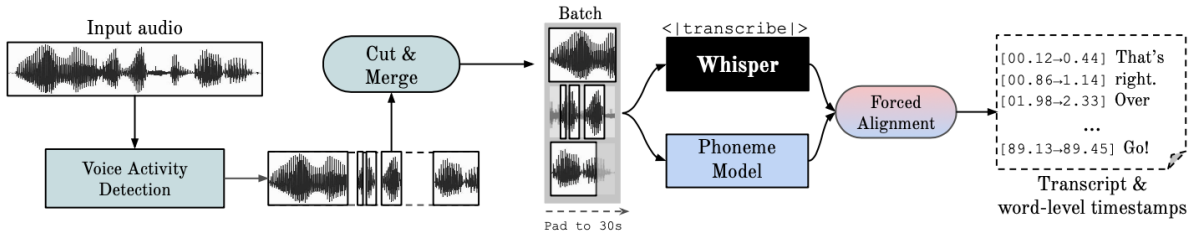

Figure 1: Whisper-Based Automatic Speech Recognition (ASR) with improved timestamp accuracy using forced alignment.

tion. As a first step, the interface takes audio, the number of speakers, the model to be used, and language as the input.

3.3 Feature Implementation

For implementing the features, we take few string inputs from the user. We first preprocess the transcriptions and user query to incorporate natural language, such as punctuation removal and stemming. We then proceed with matching of the two.

1. For the censorship feature, the user can enter a phrase or a set of phrases. The web interface extracts the corresponding timestamps for the phrase(s) to be censored by aligning them with the transcript. The audio is then masked using Pydub to censor all instances of the phrase(s).

2. For the splicing feature, the user enters the phrase and the web interface extracts the corresponding timestamps for the phrase. The audio is then spliced to create multiple audio clips, one for each instance of the phrase.

3. If the phrase entered by the user does not exist in the transcript, the audio and transcript remain the same.

4. For the speaker extraction feature, the user enters the speaker number. The web interface filters out the transcripts corresponding to that speaker and extracts the corresponding monologue from the audio by masking it using Pydub.

4 Database

Our system utilizes pre-trained models for speech recognition and speaker diarization. For speech recognition, we use Whisper, a transformer-based model trained on 680,000 hours of multilingual and multitask labeled data collected from the web, of which about 70% of the data was in English language(OpenAI, 2023).

Meanwhile, for speaker diarization, we employ SpeechBrain, a deep learning toolkit trained on the VoxCeleb dataset (Ravanelli et al., 2021). VoxCeleb is an audio-visual dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube. It contains speech from over 7,000 speakers spanning a wide range of different ethnicities, accents, professions, and ages.(A. Nagrani*, 2019)

The use of pre-trained models allows us to leverage the benefits of large-scale, high-quality training data without incurring the cost and time associated with training our own models.

5 Code

The web interface is built using ReactJS and React Bootstrap for the frontend, and Python for the backend which includes the ML models. For the backend, we have utilized various Python libraries such as pandas, pydub, NLTK, Whisper, WhisperX, PyTorch, and scikit-learn.

The source code for the project is available on GitHub at this link. The repository contains instructions for running the code

6 Evaluation

Improved accuracy: Clustering with embeddings on independent speech fragments can lead to improved accuracy in speaker identification and separation as it considers the unique voice characteristics of each speaker. In contrast, PyAnnote's speaker diarization model relies on speaker change detection based on acoustic features, which can result in errors when multiple speakers have similar acoustic properties.

Robustness to noise and overlapping speech: Clustering with embeddings can also be more robust to noise and overlapping speech, which can

make it difficult for traditional speaker diarization models to accurately identify and separate speakers. By analyzing each speech fragment independently, the clustering algorithm can better differentiate between speakers even in challenging acoustic environments.

Scalability: Clustering with embeddings is also more scalable than PyAnnote's speaker diarization model, which can become computationally intensive when processing large amounts of audio data. With clustering, the embedding vectors for each speech fragment can be precomputed and stored, enabling faster and more efficient processing.

Adaptability: Clustering with embeddings is also more adaptable to different languages and accents as it is based on speaker-specific voice characteristics rather than acoustic features that may vary between languages or accents. This makes it a more versatile approach for speaker identification and separation in multilingual and multicultural environments.

We have also used subjective listening tests and blind-audio testing where a group of listeners were required to rate the tool on the basis of audio quality and effectiveness of speaker separation. The majority of participants rated the whisper model + speech clustering model higher than the acoustic speaker diarization model. Hence we chose the Speech Clustering pipeline for our final submission.

7 Limitations

There are some limitations to our approach that need to be taken into account when interpreting the results. Firstly, the model was trained only on English data, so its accuracy may be reduced when transcribing non-English speech. Secondly, the model was trained mainly on American accents, and may therefore have reduced accuracy when transcribing speech with other accents. Finally, when speakers are speaking over each other or there is significant background noise, the accuracy of the transcription can be reduced.

8 Contribution

Akshita Gupta: Implemented the feature to extract specific phrases, evaluation, report, presentation
Charvi Jindal: Research work, model testing, model evaluation, implemented the feature to extract audio specific to a speaker
Chetan: Front-end of the website, report and presentation
Jogith S. Chandran: Research work, model testing, model evaluation, model implementation
Shivam Agrawal: Front-end of the website, report and presentation
Utkarsh Arora: Implemented the feature to censor specific phrases, evaluation, report, presentation

References

W. Xie A. Zisserman A. Nagrani*, J. S. Chung*. 2019. Voxceleb: A large scale audio-visual dataset of human speech.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint, arXiv:2303.00747*.

Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. 2016. Single-Channel Multi-Speaker Separation Using Deep Clustering. In *Proc. Interspeech 2016*, pages 545–549.

OpenAI. 2023. Whisper: General-purpose speech recognition model. GitHub repository.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. 2021. SpeechBrain: A general-purpose speech toolkit. ArXiv:2106.04624.

Jonathan Le Roux and Emmanuel Vincent. 2013. Consistent wiener filtering for audio source separation.