**Virtual Moderator**

*Jett Hays; Di Lu; Chi-Wei Feng*

**Part I: Overview**

Virtual moderator aims to provide real-time feedback on conversations by determining *what* participants say and *how* they say it. By providing real-time metrics such as volume, pace, and sentiment, speakers can optimize the acoustics of their speech while also controlling for content by cross-referencing related topics produced by the virtual moderator.

**Part II: System Flow**

Analyzing prerecorded conversations is a relatively trivial task as information is always complete and computing power is virtually unlimited. However, complexity quickly arises on the frontier of real-time conversation analysis where the viability of a system is predicated upon quick, reliable results. To manage the many different tasks our virtual moderator is responsible for, we split our program into four threads— visualization, feedback, natural language processing, and audio analysis—with all threads sharing and updating a common data source. The figure below depicts the flow for a single utterance, with each process occurring at a time than the rest and the final chunk being stored as a speech instance object.

Diagram

Description automatically generated

**Transcription:** While threading allows for the asynchronous completion of tasks, Google’s Speech-To-Text API provides transcription and timestamping services. Outsourcing transcription to Google allowed us to focus on the development of other metrics, although this approach constrains our system’s speed to that of Speech-To-Text, which is blocking. Once the word-sized transcription is received from Google, Virtual Moderator collects the final chunk depicted above and places it in the conversation according to the Speaker ID and timestamps. By only processing transcription chunks, our team avoided parsing the audio stream for potential speech occurrences, which would significantly decrease our model’s speed/ reliability.

**Natural Language Processing:** Our original proposal presented fact verification as a stretch goal. After trying a variety of fact extraction techniques—including using transformers such as ROBERTA for named entity extraction—we determined that a successful run at fact verification would require time and computing resources beyond our disposal, so we pivoted to generating topics related to conversation content. Our initial attempt involved using ROBERTA in conjunction with the CONLL NER dataset to develop sentence-based features which would then be used to label parts of speech, however, the order of magnitude class imbalances within the dataset and lengthy training times prevented this from being a viable approach.

Instead, of shelving the BERT ecosystem as a whole— which has proven to be very effective in NLP tasks— we decided to leverage the BERT-based components of Google’s autocomplete feature through scraping. This was done after extracting noun phrases with SPACY—a powerful python library founded upon pretrained neural networks—and then extracting the two most similar phrases from all possible combinations. Autocomplete suggestions were then scraped using these phrases and the top five suggestions—according to the sum of each suggestion’s similarity to each noun phrase— were returned. A simplified example of this process, and its outputs are shown below:

*Sentence*: Mayank Goel is a professor of machine learning and sensing at Carnegie Mellon.

*Most Related Phrases*: (a professor, machine learning), (machine learning, sensing)

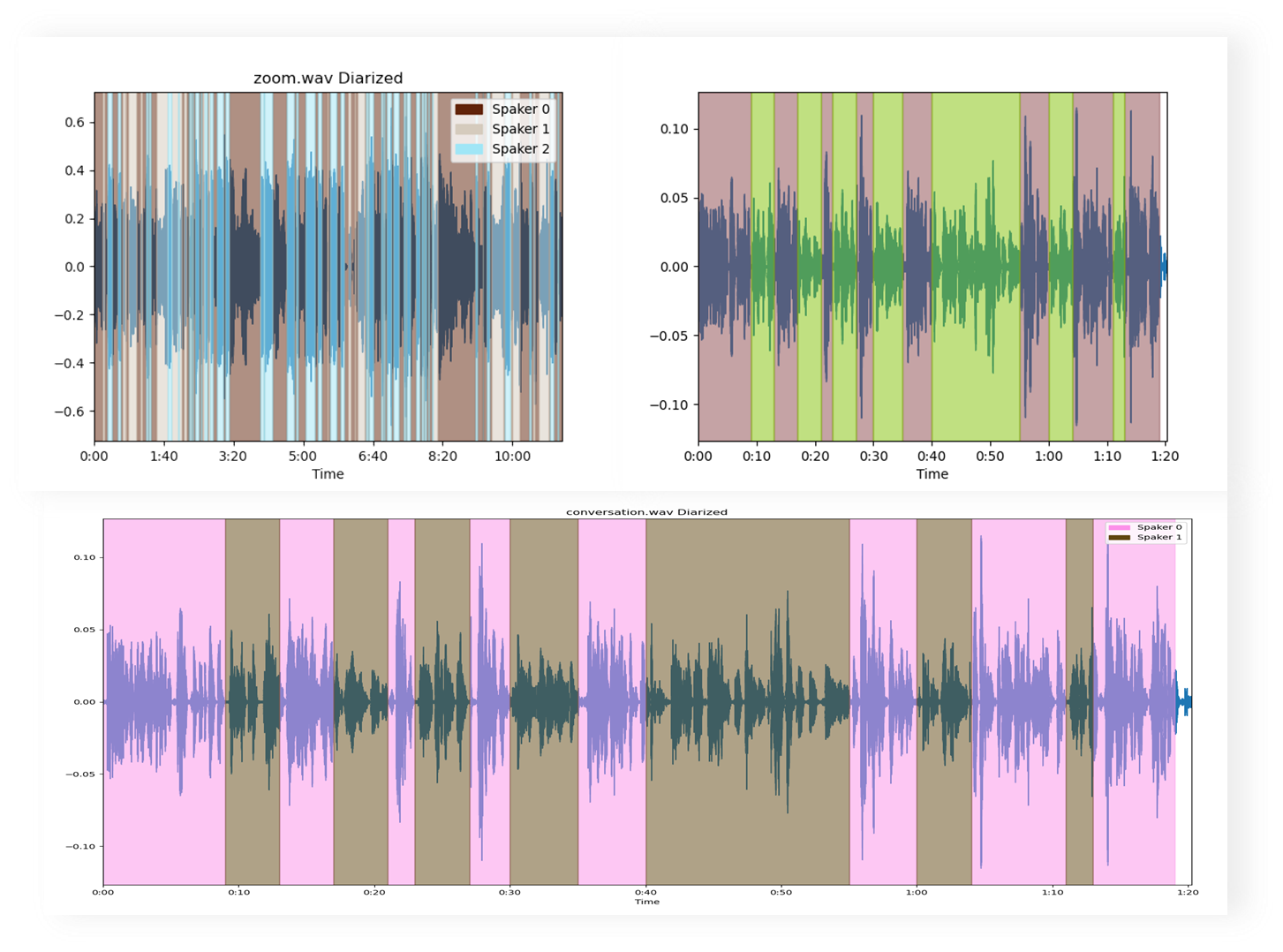
*Top 5 Related Topics*: machine learning and sensing cmu, machine learning professor stanford, machine learning sensing, machine learning and sensing lab, adaptive sensing machine learning

As there is no standard for ranking suggestions, we were unable to filter suggestions based upon relevance. An improved analysis system should run a pre-trained transformer locally, in addition to organizing suggestions with respect to confidence. For the limited resources available in our real-time moderator, however, outsourcing computation proved to be effective, as our NLP thread induced little lag.

**Speaker Identification:**

**Sentiment:**

**Display:­**

****