An Analysis of Geometric and Machine Learning Approaches to Speaker Diarization

A Technical Report submitted to the Department of Systems Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Oliver Olsen

Fall 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Oliver Olsen

Robert Riggs, Department of Systems Engineering

An Analysis of Geometric and Machine Learning Approaches to Speaker Diarization

Oliver Olsen

***Abstract*—This paper delves into the challenges and methodologies of speech diarization, focusing on the *"who spoke when"* problem in complex audio environments. This study presents a comprehensive analysis of audio features, explores geometric and pitch-based patterns for speaker identification, and recommends strategies for enhancing audio separation technology. We discuss our progression from manually labeling techniques of audio data using tools like Audacity to developing our own labeling program to transform conversational audio into Rich Transcription Time Marked (RTTM) files. We discuss our use of these datasets to evaluate a fine-tuned Pyannote learning component's accuracy. Additionally, we outline our assumptions about microphone positioning, the fidelity of our manual labeling, and the efficacy of Pyannote's machine learning tools for speaker differentiation. Our findings contribute to the development of more effective speech diarization systems, which are essential for advancing audio processing in a diverse set of applications. Our collection process translated audio files into datasets and recording a spectrum of audio features. These datasets underwent analysis to determine distinguishing patterns between speakers, looking for differences in amplitude and pitch levels in our client's device. The Pyannote audio toolkit was employed to further refine speaker differentiation. Future work will focus on creating a robust speaker diarization system using a combination of geometric analysis and machine learning optimization techniques.**

I. Introduction

Speech separation is characterized by the "cocktail party problem", colloquially understood to capture the difficulty faced in segregating individual speech signals among a mixture of background noises and competing voices, mirroring the challenge of focusing on a single conversation in a bustling party. Speech separation is crucial for enhancing the clarity and comprehension of speech in environments with multiple speakers. It involves the process of partitioning an audio stream into homogeneous segments according to the identity of the speaker. The ultimate goal is to determine not only when speech occurs but also to attribute speech segments to specific speakers within the audio. The concept of speech diarization originated with the rise of digital audio and telecommunication technologies. Initially, the focus was on improving teleconferencing systems and aiding in transcription services. Over time, the applications of diarization expanded, fueled by advancements in machine learning and signal processing. Early approaches to diarization relied heavily on speaker-specific features such as pitch, tone, and speech tempo. With the advent of machine learning, particularly unsupervised learning techniques, the focus shifted towards more sophisticated models capable of handling varied and complex audio data. These models include Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), and more recently, Deep Neural Networks (DNNs).

Despite significant advancements, speech diarization remains a challenging task due to various factors. These include the variability in speaker voice, overlapping speech, background noise, and the spontaneous nature of speech. Moreover, the lack of large and diverse annotated datasets has been a limiting factor in the development of robust diarization systems. Today, speech diarization has a wide range of applications, from enhancing voice assistants and automatic transcription services to aiding in forensic analysis and surveillance. We hope that our research can contribute to making better designed systems for source separation in messy sound settings with reverberation and background noise.

In this paper, the reader will learn the process for how we tried to determine “who spoke when” in six conversation recorded by a quad-microphone device in a multitude of settings. With this data, we worked to make recommendations to our client for creating a more effective source-separating machine. We also explain the manual and semi-manual data labeling processes and how we fine-tuned an out-of-the-box Pyannote learning component.

II. Literature Review

The development of speaker diarization technology aimed to improve automatic speech recognition in areas like air traffic control and news broadcasts. [1] The 1990's introduced key techniques such as the generalized likelihood ratio (GLR) and Bayesian information criterion (BIC)[2], which became industry standards. The 2000s saw pivotal advancements with the formation of groups like the AMI Consortium[3] and projects like the RT Evaluation by NIST[4], enhancing diarization in broadcast news and telephony. Innovations included beamforming and variational Bayesian methods.In the 2010s, deep learning revolutionized diarization, with neural networks creating significant improvements in speaker recognition. [5] This shift to neural embeddings from i-vector models enhanced performance and training efficiency. Recently, the focus has been on end-to-end neural diarization (EEND), promising to address challenges like overlapping speech and integrated optimization in speech-processing tasks. [6]

Many source separation techniques identify patterns between the speakers pitch, amplitude, and spacing of words to distinguish between similar speakers in a conversation. Recently, using Mel Spectrograms has become the most popular way to label audio for deep learning. A Mel Spectrogram differs from a regular Spectrogram which plots Frequency vs Time due to the fact that it graphs Frequency on the y-axis and uses the Decibel Scale instead of Amplitude to indicate colors.[7] By looking at pitch, and decibel ratings, we worked to add to deep learning methods with probabilities associated with the differences in decibels between our clients device which contained 4 microphones. [8]

III. Method

A. Assumptions

In our study, several key assumptions underlie the methodology and interpretation of results. Firstly, we assume that the ground truth data for speech diarization was manually labeled using the Audacity platform, which may introduce subjective variability. Secondly, there is an assumption of consistent microphone positioning across different testing scenarios, implying a controlled acoustic environment that may not fully mimic real-world variability. Lastly, we utilized an out-of-the-box Pyannote learning component for labeling and differentiating speakers in audio files, an assumption that places significant reliance on the generalizability and effectiveness of Pyannote's pre-trained models in our specific application context.

I. Ground Truth Data Labeling

The assumption that the ground truth data was labeled by ear using the Audacity platform introduces a significant variable in the accuracy and reliability of the data. Human labeling, even by experts, can be prone to inconsistencies and subjective interpretations, especially in complex auditory environments. The variability in human perception and potential fatigue factors may affect the precision of the labeled data. This assumption is critical as it underpins the validity of the ground truth, which is used for evaluating and training speech diarization models.

II. Microphone Position Consistency

Assuming that the microphone positions (1,2,3,4) remained constant across different scenarios is an important consideration in the experimental setup. This assumption implies a controlled and uniform acoustic environment for each test scenario, which may not accurately reflect real-world conditions where microphone placements can vary significantly. The consistency in microphone placement is crucial for the reproducibility of the results and for ensuring that any observed variations in the data are attributable to the speech diarization technology itself, and not to changes in the audio capturing setup.

III. Use of Pyannote for Learning and Labeling

The assumption regarding the use of an out-of-the-box Pyannote learning component for labeling .wav files and distinguishing between different speakers (person A from person B) is pivotal. This presupposes that the pre-trained models and algorithms provided by Pyannote are sufficiently generalizable to the specific audio data in our study. The effectiveness of Pyannote in accurately labeling and differentiating speakers depends on factors such as the quality of the training data it was exposed to and its adaptability to different acoustic environments and speaker characteristics. This assumption underlines the reliance on the robustness and versatility of the Pyannote framework in varied scenarios.

B. Data Collection

The data collection process for our study involved meticulous manual efforts and the use of sophisticated audio processing tools. Initially, we manually labeled the .wav files using Audacity to identify and mark segments where the customer and the clerk spoke. These segments were then converted into RTTM (Rich Transcription Time Marked) files, serving as our ground truth for assessing the Pyannote learning component's accuracy.

Further, leveraging Pyaudio, we transformed the .wav files into a comprehensive dataset capturing various audio characteristics such as amplitude and pitch across multiple microphones. This detailed data, encompassing parameters like minimum, maximum, average, and standard deviation of amplitude and pitch for each microphone, was instrumental in identifying patterns and trends. These insights were crucial in distinguishing between person A and person B, in comparison to our ground truth labels. In the Pyaudio dataset, we used a sliding window technique to break down information into smaller blocks for more

detailed analysis, significantly enhancing the accuracy of pattern and anomaly detection.

	A	B	C	D	E	F
1	Time Start	Time End	Mic1 Avg Amplitude	Mic2 Avg Amplitude	Mic3 Avg Amplitude	Mic4 Avg Amplitude
2	0	1	0.016470993	0.020723794	0.019113195	0.016749789
3	0.249977324	1.249977324	0.014455706	0.018102417	0.017124783	0.015337354
4	0.499954649	1.499954649	0.012247046	0.014418408	0.019555058	0.016493883
5	0.749931973	1.749931973	0.013214747	0.013943916	0.02565235	0.021020581
6	0.999909297	1.999909297	0.018486271	0.021669429	0.033684929	0.027980218
7	1.249886621	2.249886621	0.021942184	0.02561555	0.037158196	0.031340828
8	1.499863946	2.499863946	0.023253823	0.026803687	0.033728076	0.029474539
9	1.74984127	2.74984127	0.022252013	0.026131164	0.030297755	0.026365493
10	1.999818594	2.999818594	0.017002743	0.019726594	0.022716761	0.019426483
11	2.249795918	3.249795918	0.019560454	0.02122862	0.023690674	0.020682172
12	2.499773243	3.499773243	0.021547191	0.022950974	0.025433419	0.021564943
13	2.749750567	3.749750567	0.023952473	0.026742984	0.02939045	0.023615187
14	2.999727891	3.999727891	0.029308228	0.031013917	0.031392255	0.027287425
15	3.249705215	4.249705215	0.025904847	0.028203039	0.029028269	0.024654845
16	3.49968254	4.49968254	0.023812989	0.027516288	0.026524182	0.023546972
17	3.749659864	4.749659864	0.021934915	0.023865553	0.020791571	0.020935699
18	3.999637188	4.999637188	0.016397932	0.01846771	0.017781257	0.01703156
19	4.249614512	5.249614512	0.015209508	0.017045298	0.015472942	0.015633784
20	4.499591837	5.499591837	0.013259538	0.013635434	0.013166676	0.012990588

Figure 1. *Color Gradient Amplitude.* T
he simple color gradient allows us to see an apparent pattern in the time series data corresponding to the speaker's amplitude levels.

When we looked into the data set, we noticed that for many of the conversations, the average amplitude was a predictor of speaker location. In the figure above, we color coded the data to show a pattern of the loudest amplitude(green) to quietest(red) across each 1 second data point in the recording. Within each sliding window (.5s), amplitude was ranked and pattern appeared between microphones when each speakers talked.

It was suspected that Microphone 2 was closest to person A and Microphone 3 was closest to person B because the consistent highest amplitude of data entry for that half each second window in the conversation. We explored the minimum, average, maximum, and standard deviation, of both the pitch and amplitude readings for 6 conversations, totaling 18 minutes of conversation data.

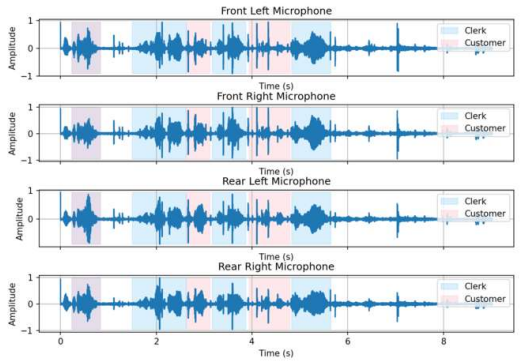

Figure 2. *Manually Labelled Ground Truth*

With the data collected, ground truth labels were stored for each conversation in RTTM files for future training in Pyannote. Figure 2 shows the ground truth depiction of each microphone in the listening device.

C. *Experimental physical set-up*

We did not have time to set up this experiment to produce verified ground truth and eliminate assumption II, that the microphone position was consistent. However, in future testing, we plan to create a dataset created in a lab setting to identify any patterns in the speaker locations based on amplitude, pitch, etc. and help design a device that can more efficiently distinguish between speaker A and B based on geometric positioning and acoustics of the device.

Setting up a more controlled environment to collect our audio data can help ensure that the placement of our four microphones, identified as 1 to 4, is consistent throughout all our test scenarios. This consistency is key for eliminating variables that could throw off our analysis.

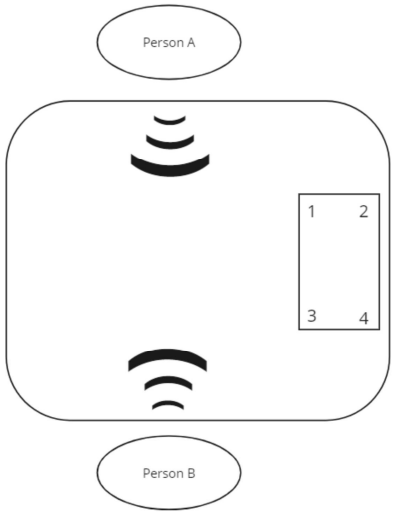

Figure 3. *Experimental Set-up*

We'll conduct recordings with two speakers, positioned at various locations around the microphones, and carefully label the sound data. Analyzing the volume, pitch, and other audio metrics from these recordings will help determine if there's a reliable way to tell the speakers apart based on the sound information alone.

The Pyannote-based learning algorithm, which has been trained to differentiate speakers could then use this data set and be fine-tuned accordingly. By combining the geometric approach with the results of the algorithm, we can get a more accurate model for identification between speakers in a conversation. This could have significant applications in devices that rely on precise speaker identification, and we believe it's a step towards creating smarter and more responsive audio recognition systems.

IV. PRELIMINARY RESULTS

A. *Geometric Exploration*

Our results currently rely on the assumption of the microphones being laid out in the pattern described in Figure 3. These results turned out to be inconsistent among the six conversations. We found that for 3 of the 6 conversations, the order of microphone amplitude was qualitatively predictive of the speaker. For example, in Figure 4 we

noticed that the two quietest microphones for Speaker A were the two loudest for Speaker B.

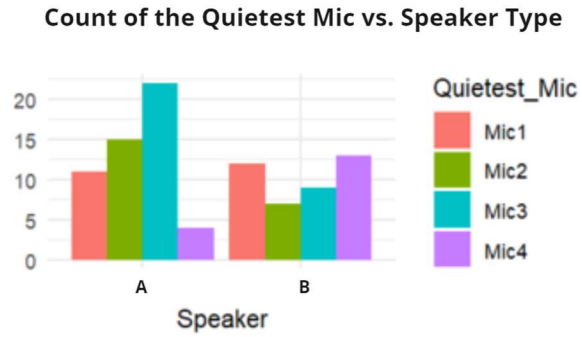

Figure 4. *Mic Levels Appear Predictive of Speaker Location*

However, this was not the case for 3 conversations in our dataset. As you can see, in Figure 5, the Mic levels were nearly the exact same for the conversation. The inconsistent results we found led us to consider another metric, pitch levels in each conversation.

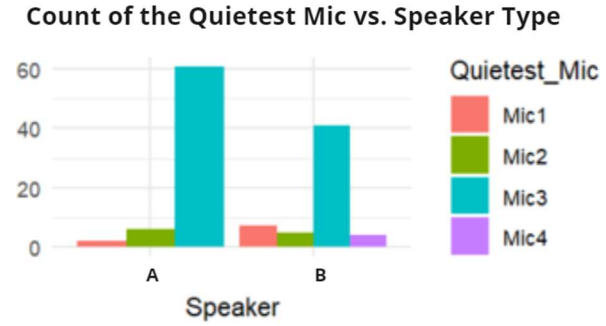

Figure 5. *Mic Levels Are Not Predictive of Speaker Location*

Additionally, we searched for patterns in pitch data to try and identify any like between the pitch of the person A when they spoke compared to person B. This looked promising at the beginning, as we saw clusters of the pitch which looked different in half of the conversations. However, we determined that this was not a significant predictor of the speaker because frequently conversations included pitch levels that were nearly the same. In the audio file, this would sound like two women speaking together, giving a similarly high-pitched voice or two older men speaking together who had a similarly low-pitched voice. After looking at the data we gathered from these conversations, we decided to move on from the metric of the quietest microphone and consider another approach, exploring the Pyannote ML pipeline and finetuning it based on our ground truth dataset.

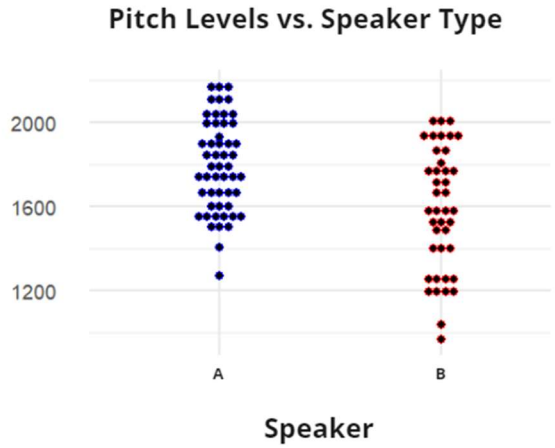

Figure 6. *Dot Plot of Average Pitch by Speaker Type.* Pitch appeared predictive of the speaker type for half of our conversations.

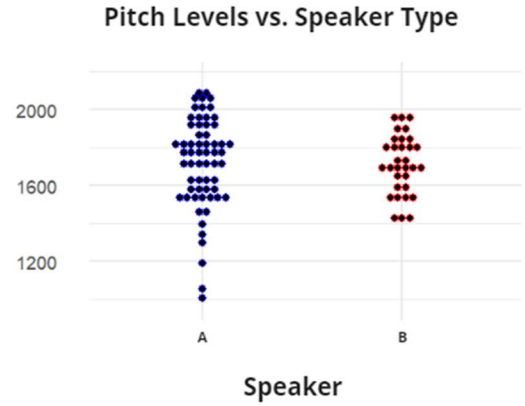

Figure 7. *Dot Plot of Average Pitch by Speaker Type.* Pitch was not predictive of the speaker for half of our conversations.

The variety of diverse environments our client provided from conversations recorded in actual field settings led to significant complications. We realized the importance of finding a generalizable solution. The Pyannote pretrained learning component accounts for a variety of factors to make the problem more generalizable across a plethora of scenarios.

Both pitch and amplitude separately would not give conclusive evidence to the "who spoke when" problem. However, we suspected if we combined the predictive power of these features, and others like them, we could model speaker type more effectively.

B. Pyannote Learning Component

Pyannote is an open-source toolkit primarily designed for the process of partitioning an audio stream into homogeneous segments according to the speaker identity. It's widely used in the field of audio analysis and speech processing. Pyannote offers a range of functionalities including speaker segmentation, identification, and tracking,

making it a versatile tool for handling various audio processing tasks. Built in Python, it integrates well with other machine learning and audio processing libraries, making it a popular choice for researchers and developers in speech recognition and related areas.

We employed the Pyannote audio toolkit for fine-tuning a pre-trained speaker diarization model to adapt it to our specific audio dataset. Initially, we selected a suitable pre-trained model from Pyannote's extensive library, chosen for its baseline performance on general datasets. We then extracted relevant audio features using Pyannote's built-in feature extraction tools, crucial for the model's learning process. The fine-tuning phase involved adjusting key training parameters, such as the learning rate, batch size, and epoch count, to optimize the model's performance on our dataset. This process was iteratively refined, guided by evaluations using the Diarization Error Rate (DER) metric, until the model demonstrated a marked improvement from 54% to 24% DER with our audio data. The fine-tuned model thus emerged as a tailored solution for our specific diarization tasks, showing enhanced accuracy and adaptability compared to its original version.

C. Semi-Manual GUI

Additionally, we made a Guided User Interface (GUI) tool to speed up the manual processing of audio labeling. We have left the link to this as a Github for public use in audio labeling.

Fatigue and user error are significant barriers to creating a valuable dataset for training purposes. We hope that this tool will act as a medium to more easily process their audio labels for development, training, and testing purposes. This functionality takes in a .wav audio file and an optional RTTM file to allow the user to drag and drop labels on a visual interface, quickly and accurately assigning proper segments to speakers. Once the labeling is done, the user simply presses submit to export a RTTM and UEM file from the program. The results of creating this functionality greatly decreased the time we spent manually labeling data and allowed us to create the dataset which was used in fine-tuning the Pyannote pipeline. By doing this, we were able to speed up the process of manually labeling the audio files which took on average an hour to do for each 3-minute conversation, to less than 15 minutes in the GUI, decreasing the time for labeling by a factor of 4.

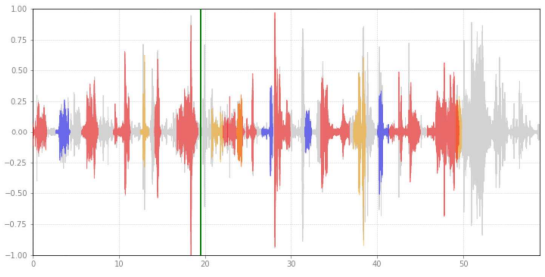

Figure 8. *Before Editing the Preloaded RTTM*

We used the GUI to quickly edit the RTTM files which Pyannote had previously labeled. At that point, we could easily adjust each RTTM file and export a more accurately labelled RTTM for future training purposes. We plan to use this dataset to gain better results when fine tuning the Pyannote pretrained pipeline. Figure 6 shows the RTTM given from Pyannote, while Figure 7 shows the GUI after editing the RTTM, ready for export.

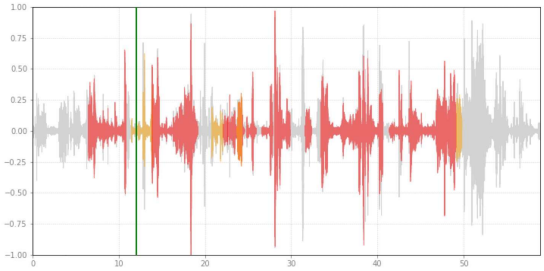

Figure 9. *After Editing the Preloaded RTTM*

The visual representations of audio waves make it much easier to identify specific segments like pauses, intonations, or speaker changes, thus improving the precision of labeling. Previously, we would use a similar functionality in Audacity, a mulit-track recording editor available on Mac and iOS devices.

V. DISCUSSION

The main point of discussion in this project is how the scope of our project changed over the course of the semester. Initially, we wanted to look at the geometric approach to speaker diarization and see if we could improve our client's device setup process and design manufacturing. This proved to be challenging after looking at the data on amplitude and pitch and realizign how difficult it was to find patterns that could give clues to who was speaking when.

Hand labeling the conversations turned out to be a labor-intensive task and we turned our focus towards cutting down the process for producing quality labels to compare ground truth and predicted audio.

Once we got more results from the geometric approach of amplitude and pitch data, we realized trying to triangulate the position of the speaker based on the differences in amplitude between the microphones was not giving consistent results across the variety of environments our client was using the devices in.

These results encouraged us to look into alternative solutions, such as the Pyannote learning component, which can take in an audio file and separate the sources of speech in more generalized situations. These results proved to be more promising and led us to believe that future work should be done using a learning component that is trained on a large dataset of recorded conversations with background noise, reverberation, and in a multitude of different set-ups. By setting up this model with a variety of environmental scenarios, the learning component should be able to more consistently give quality speaker assignments across more generalizable data.

VI. CONCLUSION

Speaker Diarization is a difficult problem that requires clear definitions of speech segments and accurate speaker identification to be effective. Our exploration into the realm of speech diarization, specifically through manual labeling and the use of machine learning algorithms like Pyannote, has demonstrated promising advancements, yet also revealed the complexity of this field. As we refine our methods and harness more controlled experimental setups, we move closer to solving the nuanced challenges of speaker separation.

In future work, creating an experimental setting to test the device will help with the investigating and deciding which metrics are most predictive of speaker location and in doing so, more accurately separate speaker A from speaker B in a variety of conversational settings. Using our semi-manual guided user interface will speed up the process of labeling and testing our hypothesis for geometric evaluations. With the tool, we are able to make a more powerful dataset to train and fine tune the Pyannote learning component. We may also consider looking at alternative speech diarizing learning components that could be a better fit for our datasets.

Our study has laid the groundwork for our team to engage with enhanced speech diarization systems, with a focus on developing a reliable device capable of distinguishing between multiple speakers. By establishing a controlled environment and standardized microphone placements, we have identified a pathway to mitigate variables and improve the accuracy of our system. The integration of Pyannote's machine learning algorithm represents a promising advancement in the field of audio processing. Our findings suggest significant potential for applications ranging from smart home devices to accessibility tools for the hearing impaired. Future research will build upon these findings, aiming to create a versatile and precise diarization device.

REFERENCES

[1.] U. Jain, M. A. Siegler, S.-J. Doh, E. Gouvea, J. Huerta, P. J. Moreno, B. Raj, R. M. Stern, Recognition of continuous broadcast news with multiple unknown speakers and environments, in: Proceedings of ARPA Spoken Language Technology Workshop, 1996, pp. 61–66.

[2.] S. S. Chen, P. S. Gopalakrishnan, Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion, in: Tech. Rep., IBM T. J. Watson Research Center, 1998, pp. 127–132

[3.] AMI, AMI Consortium. http://www.amiproject.org/index. html

[4.] NIST, Rich Transcription Evaluation. https://www.nist.gov/itl/ iad/mig/rich-transcription-evaluation

[5.] Anguera, C. Wooters, J. Hernando, Purity algorithms for speaker diarization of meetings data, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, volume I, 2006, pp. 1025–1028.

[6.] Kenny, D. Reynolds, F. Castaldo, Diarization of telephone conversations using factor analysis, IEEE Journal of Selected Topics in Signal Processing 4 (2010) 1059–1070.

[7.] P. Chiariotti, M. Martarelli, P. Castellini. Acoustic beamforming for noise source localization – Reviews, methodology and applications. Mech. Syst. Sig. Process., 120 (2019), pp. 422-448, 10.1016/j.ymssp.2018.09.019

[8.] Boyang Zhang Jared Leitner Sam Thornton Audio Recognition using Mel Spectrograms and Convolution Neural Networks Noise Lab, Dept. of Electrical and Computer Engineering, University of California, San Diego Report38.pdf (ucsd.edu)