Preliminary Outline: How to approach Multi-Speaker Identification

Abstract: The topic of this investigation is multi-speaker identification in the field of signal processing and communications. To understand the core problem in simple terms, consider an Alexa or Google Home device. How can the device distinguish between multiple people speaking? Or for dialogue stored in a monaural rather than a stereophonic format, how can different speaking subjects be discerned? This report will provide current approaches to solving this problem, analyze the pros and cons of different solutions, and suggest a path forward to solving the "who spoke when?" problem, and creating a useful product.

Introduction: Much of the research in speaker recognition assumes that voice data is only from a single person speaking. The issue is that many real-world datasets/scenarios have multiple speakers, violating this assumption. This is where the concept of **speaker diarization** becomes important. Speaker diarization is the process of partitioning an input audio stream into homogeneous segments based on speaker identity. It addresses the problem of "who spoke when?".[1] Figure 1 shows the general results expected from a diarization process.

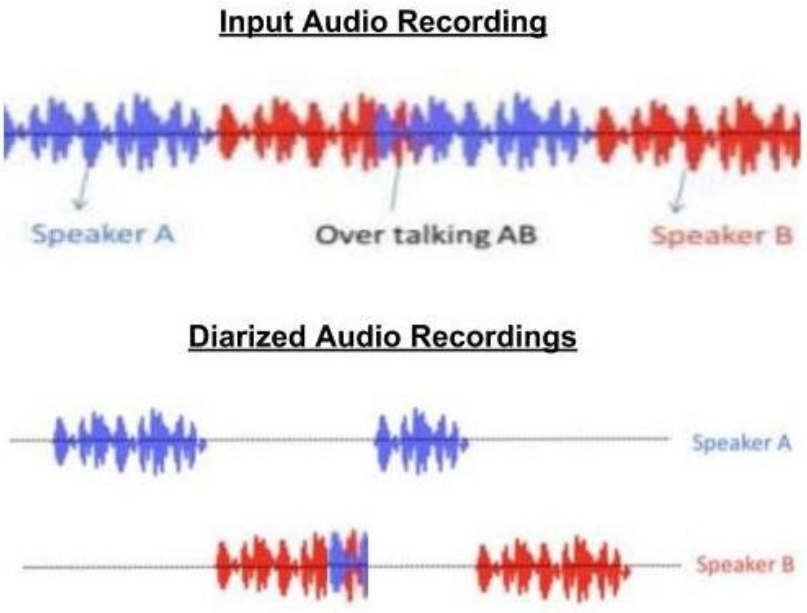

Figure 1. Input audio data undergoing speaker diarization into two speakers, A and B.[2]

The result of speaker diarization is a clear split of which person is speaking and at which time. Speaker diarization is a complex topic with many approaches being researched. The state-of-the-art approach until recently, was using i-vectors. This involved extracting i-vectors

from the audio, further reducing their dimensionality via PCA (Principal Component Analysis), applying k-means clustering, and generating a single new i-vector for each speaker.[3] However, the success of deep neural networks has recently created new, more advanced approaches to solving the diarization problem.

Modern Approaches: In 2017, Snyder et al.[4] created a model for diarization using deep neural network embeddings, called x-vectors. This deep neural network outline is shown in Figure 2.

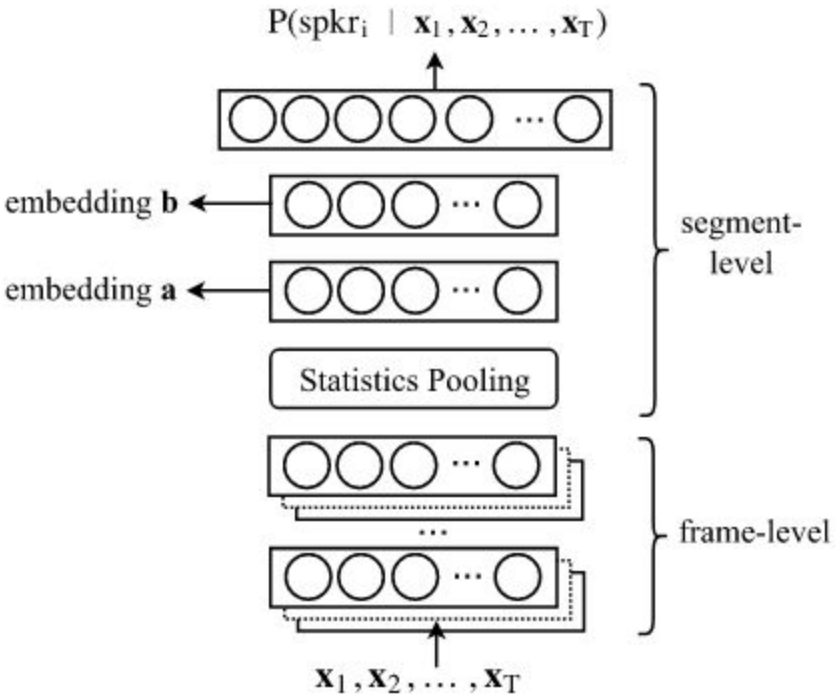

Figure 2. Diagram of the neural network proposed by Snyder et al.

Another modern, deep learning approach was proposed in 2017 by Wang et al. from Google Brain and Carnegie Mellon University.[5] The proposal used LSTM (Long Short-Term Memory) networks to extract a vector representation of the audio called a d-vector. Figure 3 shows a flowchart of the d-vector diarization system proposed in the paper. The Diarization Error Rates (DER) of the d-vector method were lower than those of the i-vector method.

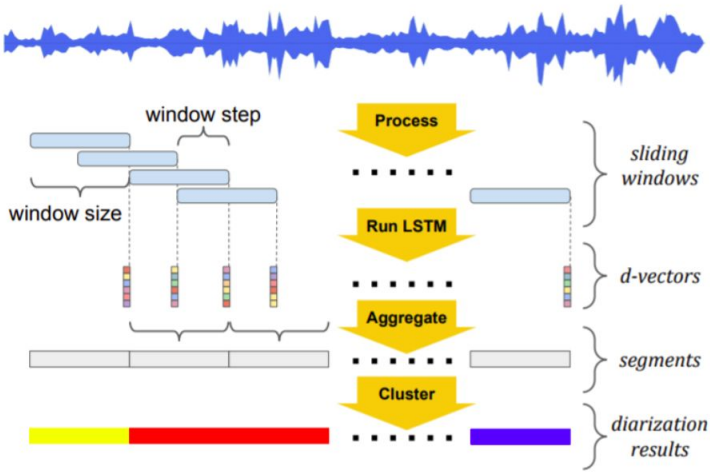

Fig. 1: A flowchart of our d-vector based diarization system.

Figure 3. A flowchart of the d-vector based diarization system.

General Difficulties: It is hard to compare the results of various speaker diarization methods because of the various methods employed by research teams. For instance, differing training data, different evaluation protocols and software, and different components in the diarization pipeline are all reasons that it is hard to evaluate and compare diarization methods.

Path Forward: Wang et al.'s proposed method is likely the path forward for a few reasons. Firstly, the Diarization Error Rate (DER) was generally lower than other methods. Second, there are public repositories featuring Wang's spectral clustering code written in python.
The d-vectors can be implemented with the help of Resemblyzer[6], an open-source repository. This repository will perform segmentation and embedding extraction on the input audio data. Resemblyzer comes with pre-trained neural network embeddings. This is key for accurate and flexible diarization. From there is the task of creating labels (which cover each value of time), attributing them to the proper speaker, and outputting them for either visual display or to feed into the system for identification purposes, which can be done with Wang's clustering prediction package and pydub. Something to think about moving forward is how to integrate the process with an Alexa or Google Home device.
In the case that a modified transcript is desired, Google text-to-speech API can be used to convert audio to text, which can be fed into the python and used to output a transcript with labels for each speaker based on the spectral clustering assigned labels for each value of time.
A very helpful repository[7] published and maintained by Wang contains many sources and other codebases relating to state-of-the-art diarization methods. It can be used to further explore options to solve this problem.

Works Cited

1. Anguera Miro, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., & Vinyals, O. (2012). Speaker Diarization: A Review of Recent Research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2), 356–370. doi:10.1109/tasl.2011.2125954
2. https://hackernoon.com/speaker-diarization-the-squad-way-2205e0accbda
3. Verma, P., Das, P.K. i-Vectors in speech processing applications: a survey. Int J Speech Technol 18, 529–546 (2015). doi:10.1007/s10772-015-9295-3
4. Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S. (2017) Deep Neural Network Embeddings for Text-Independent Speaker Verification. Proc. Interspeech 2017, 999-1003. doi: 10.21437/Interspeech.2017-620.
5. Q. Wang, C. Downey, L. Wan, P. A. Mansfield and I. L. Moreno, "Speaker Diarization with LSTM," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, 2018, pp. 5239-5243. doi: 10.1109/ICASSP.2018.8462628.
6. https://github.com/resemble-ai/Resemblyzer
7. https://wq2012.github.io/awesome-diarization/