Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Peilin Wu April 9, 2024

Beyond Audio: Advancing Speaker Diarization with Text-based Methodologies and Comprehensive Evaluation

By

Peilin Wu

Jinho D. Choi, Ph.D.
Advisor

Computer Science

Jinho D. Choi, Ph.D.
Advisor

Davide Fossati, Ph.D.
Committee Member

Alissa Bans, Ph.D.
Committee Member

2024

Beyond Audio: Advancing Speaker Diarization with Text-based Methodologies and Comprehensive Evaluation

By

Peilin Wu

Jinho D. Choi, Ph.D.
Advisor

An abstract of
A thesis submitted to the Faculty of the Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements for the degree of
Bachelor of Science with Honors

Computer Science

2024

Abstract

Beyond Audio: Advancing Speaker Diarization with Text-based Methodologies and Comprehensive Evaluation
By Peilin Wu

This thesis introduces a novel approach to Speaker Diarization (SD), diverging from the traditional reliance on audio signals by exclusively leveraging text-based methodologies. It includes comprehensive evaluation methods tailored to textual data. By employing the T5-3B model within both the Single Prediction Model (SPM) and Multiple Prediction Model (MPM) frameworks, and incorporating data processing pipelines designed to enhance the model's performance on transcripts generated by Automatic Speech Recognition (ASR) models, this study assesses the feasibility and effectiveness of text-based SD in distinguishing "who speaks what" across various two-speaker dialogues via sentence-level Speaker Change Detection and aggregation mechanism. Furthermore, this research proposes and validates two new evaluation metrics: the Text-based Diarization Error Rate (TDER) and Diarization F1 (DF1). These metrics are specifically tailored to address the unique challenges of text-based SD and the joint assessment of ASR and SD errors. Alongside these metrics, we also propose a sequence alignment algorithm designed to align different transcripts effectively and efficiently, particularly in situations with overlapping speech.

Experiments conducted on a curated dataset, which encompasses 7 open-domain conversational contexts, demonstrate that text-based methods can perform comparably to—and, notably, for short conversations under 15 minutes, even outperform—traditional audio-based diarization systems by 2.5% to 10%. The newly proposed text-based metrics, tested on the CallHome dataset through both manual inspection and error type analysis, show an enhanced ability to accurately assess the performance of text-based SD and joint ASR and SD systems in providing informative transcription results. Moreover, the proposed multiple sequence alignment algorithm achieves better alignment results (0.99 accuracy) compared to previous dynamic programming-based methods (0.92 accuracy). These findings not only challenge existing paradigms within the field of SD but also pave the way for further advancements in conversational analysis and AI, highlighting the untapped potential of textual information in SD tasks.

Beyond Audio: Advancing Speaker Diarization with Text-based Methodologies and Comprehensive Evaluation

By

Peilin Wu

Jinho D. Choi, Ph.D.
Advisor

A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements for the degree of
Bachelor of Science with Honors

Computer Science

2024

Acknowledgments

I would like to express my sincere gratitude to my advisor Dr. Jinho D. Choi, for supporting and guiding my research at Emory University in the field of NLP in the past 2 years. Dr. Choi's rigorousness and helpfulness trained me to be a capable researcher, especially in conducting experiments in the right way. I am also thankful to Dr. Davide Fossati and Dr. Alissa Bans for kindly agree to be the member of my committee. They provide valuable inputs for my thesis project itself and also the academic writing.

I would like to say thank you to Chen Gong for co-designing and implementing the text-based evaluation metrics, as well as conducting the experiments about metrics behavior in this thesis project. Chen Gong is a researcher that is a pleasure to work with and I wish him the best in his future career.

I would also like to say thank you to all my friends and classmates at Emory University for supporting me outside of my research. They are the essential network that provided me with the resilience to pursue my studies. Their encouragement and companionship have been invaluable.

Contents

1 Introduction **1**

1.1 Thesis and Research Questions 4

2 Background **5**

2.1 Audio-based Systems 5

2.2 Utilization of Text-based Features 7

2.3 Speaker Change Detection 8

2.4 Metrics 8

2.5 Sequence Alignment Algorithm 10

3 Text-based Speaker Diarization **12**

3.1 Task Overview 12

3.2 Single Prediction Model 13

3.3 Multiple Prediction Model 15

3.4 Data Processing 17

4 Text-based Speaker Diarization Evaluation **20**

4.1 Text-based Metrics 20

4.1.1 Text-based Diarization Error Rate 21

4.1.2 Diarization F1 22

4.2 Aligning Transcripts 23

4.2.1 Current Alignment Limitations 23
4.2.2 Scoring Matrix Population 26
4.2.3 Backtracking 29
4.2.4 Optimization of Matrix Population 31

5 Experiments Setup **33**

5.1 Datasets 33
5.2 Text-based SD Approach Experiments 34
5.2.1 Data Processing 34
5.2.2 Model 35
5.2.3 Model Evaluation 36
5.3 Text-based Metrics Experiments 37
5.3.1 Metrics Behavior 37
5.3.2 Alignment Efficacy Experiment Setup 37
5.3.3 Alignment Package: align4d 38

6 Results and Analysis **40**

6.1 Model Performance 40
6.1.1 Conversational Length-based Analysis 40
6.1.2 Input Length-based Analysis 42
6.1.3 Text-based Error Types 42
6.2 Text-based Metrics 45
6.2.1 Metrics Behavior Analysis 45
6.2.2 Alignment Algorithm Analysis 47

7 Conclusion **48**

Bibliography **50**

List of Figures

3.1 Demonstration of SPM prediction on a conversation with 6 sentences. The parameters in this demonstration are set to $h = 4$ and $k = 1$. . . 14

3.2 Demonstration of MPM prediction on a conversation with 6 sentences. The maximum window size is 4 sentences, with 1 sentence shift each time. In this demonstration, the majority voting is used for aggregation. 17

3.3 Data processing pipeline utilizing ground truth transcripts and ASR-generated hypothesis transcript to obtain training and evaluation data with ASR discrepancies. 18

4.1 An example of normal transcript that need to be aligned together. . . 24

4.2 Transcripts aligned with pairwise alignment (original Needleman-Wunsch algorithm). The 2 dimensional alignment cannot handle the overlapping situation in dialogue. The character-based alignment also cannot do partial matching between words with different spelling but same position. 25

4.3 Transcripts aligned with multiple sequence alignment. The 3 dimensional alignment considers the overlapping situation. The token-based alignment matches between words with different spelling with Levenshtein Distance. 25

4.4 The backtracking example using Algorithm 2 31

6.1 Average WDER with respect to conversation length for best audio and text-based SD systems with 5 minute as the interval. 41
6.2 Percentage of three error types in the 50 randomly selected input samples. 43

List of Tables

4.1 Permutations of index tuples as the result for *index_perm* function in the case of 2-speaker conversation. The sequence is from top to bottom which cannot be violated in order to keep the correctness of MSA algorithm. 28

5.1 Corpora used for the curated dataset. 34

5.2 Categorization of models based on their approach to speaker diarization. In this table, SC means spectral clustering and AHC means agglomerative hierarchical clustering. 36

6.1 Performance comparison in terms of WDER and WDER-S with audio-based SD systems (the lower the better) with respect to time. 41

6.2 Performance comparison in WDER and WDER-S of SPM and MPM with respect to number of maximum input sentences for each sliding window. 42

6.3 Average percentages of the four types of errors over all tokens. 46

6.4 Comparing the traditional metrics (**DER**, **WDER**, **WER**) with our new evaluation metrics (**TDER**, **DF1**). For **DF1** the precision (**P**) and recall (**R**) score are also shown. For **TDER**, lower score means higher performance; For **DF1**, higher score means higher performance. 46

6.5 Performance comparison on three types of alignment algorithms on Amazon Transcribe and Rev AI transcripts on CallHome dataset using alignment accuracy. 47

Chapter 1

Introduction

Speaker Diarization (SD), a crucial task in audio processing, is aimed at identifying "who speaks when" by segmenting and attributing portions of audio to individual speakers [33]. This capability is important in a variety of applications, from transcribing meeting minutes to generating accurate medical records, facilitating easier indexing, search, and analysis of audio recordings. As conversational AI systems evolve, the integration of SD in joint Automatic Speech Recognition (ASR) and SD tasks becomes increasingly significant. This integration focuses on the "who speaks what" problem, which is essential for preparing and evaluating training data for conversational AI systems. Given the rapid consumption of publicly available data by large language models (LLMs), there's an urgent need for innovative methods to generate and refine conversational datasets.

Historically, SD has been approached through two main methodologies: a modular approach, which involves segmentation followed by clustering of audio segments based on speaker characteristics, and End-to-End Neural Diarization (EEND), which leverages deep learning to perform diarization in a supervised and more holistic way. For the modular approach, the audio is first cut into short segments either according to voice activity [19] or change of speaker [4, 38]. Then, speaker-related features are

extracted from each segment [45, 9, 22], and grouped together via various clustering algorithms [48, 37]. For the end-to-end approach, the clustering and potentially feature extraction stage is replaced with an one-stage supervised neural network-based model [34, 47]. Despite their advances, both approaches primarily rely on audio features, which introduce several challenges. For instance, different speakers with similar voice characteristics can confuse SD systems, and the quality of the recording can significantly impact the performance of diarization models. Moreover, from the perspective of evaluation, traditional metrics for SD, such as the Diarization Error Rate (DER) [11] and the Jaccard Error Rate (JER) [42], are not well-suited for text-based SD, failing to account for the discrete and structured nature of textual data.

Recent research has explored the use of semantic features to enhance SD performance [32, 35]. By incorporating contextual information from speech, these attempts aim to overcome the limitations posed by purely audio-based features, including direct usage of text-based features for clustering or classification [13], and indirect usage that leverages language model as a error correction mechanism [36, 49]. Evaluation metrics have also evolved, with the introduction of the Word Diarization Error Rate (WDER) [32, 43] to assess text-based SD performance. However, existing approaches to incorporating semantic features have faced significant limitations. The direct usage of text features has not leveraged the latest advancements in language models, thereby failing to utilize the full potential of semantic features for SD. The indirect approaches did use state-of-the-art language models, but merely as a post-processing step, without directly integrating semantic information into the diarization process. Furthermore, there has been a lack of exploration into using text as the sole input for SD to discover the limit of text-based approach. An all-in-one metric for comprehensive evaluation is also needed since the WDER lacks full coverage of types of text-based SD and joint ASR and SD errors.

To address the aforementioned problems and challenges, this thesis proposes a novel text-based SD approach using only the dialogue transcript as input. Our approach is compared with multiple recent audio-only SD models on a curated dataset, demonstrating superior performance in short conversations and comparable effectiveness in longer dialogues. Additionally, for comprehensive evaluation, this thesis introduce the Text-based Diarization Error Rate (TDER) and Diarization F1 (DF1) as new evaluation metrics tailored for text-based SD tasks. These metrics are complemented by the development of align4d, a tool for multi-sequence alignment on conversational data, facilitating accurate and efficient metric calculation.

The main contribution in this thesis includes:

1. A data processing pipeline specifically designed for SD based on ASR results, providing a practical method for generating and refining datasets suitable for text-based diarization analysis.

2. Text-based SD approach tailored for two-speaker dialogues, which leverages dialogue transcripts as the sole input, with competitive performance compared to other audio-based models and comprehensive error analysis.

3. The establishment of TDER and DF1 metrics for comprehensive evaluation of text-based SD and joint ASR and SD tasks, accompanied by development and implementation of a multiple sequence alignment algorithm tool.

The remaining part of this thesis is organized as follows: Chapter 2 provides an literature review on recent SD systems and evaluation metrics. Chapter 3 introduces the text-based approach for SD based on sentence-level Speaker Change Detection (SCD) and the data processing pipeline. Chapter 4 introduces the TDER and DF1 metrics, the complementary multiple sequence alignment algorithm, and visualization tool. Chapter 5 presents the experiment on text-based SD compared with recent

audio-based model as well as the analysis of the result. Finally, Chapter 6 presents the experiment and analysis on TDER and DF1 metrics with multiple sequence alignment.

1.1 Thesis and Research Questions

The thesis for this work is: We can achieve Speaker Change Detection and Diarization with textual data as the only input with a Large Language Model (LLM) based approach and correctly demonstrate its performance on text.

To achieve this thesis, here are two research questions that are need to be answered:

1. Can we train a model that detects the change of speakers with LLM-based model?

2. Can we correctly and accurately show the performance of the text-based model and compare it with audio-based models?

Chapter 2

Background

This chapter primarily introduces recent audio-based SD systems, two approaches for utilizing text-based features to enhance SD results, text-based or multimodal SD systems, speaker change detection, and the development of evaluation metrics. In addition to systems for speech processing and evaluation, sequence alignment algorithms, which are necessary for data processing and evaluation as introduced in Section 3.4, are also discussed.

2.1 Audio-based Systems

The modular approach has been the first and mainstream way of tackling SD since this task was established. For segmentation, the first stage of modular SD, Voice Activity Detection (VAD) modules like MarbleNet [19] are often used to find the silent points for segmenting within the whole audio. To avoid producing segments that are too short for subsequent processing, Speaker Change Detection (SCD) models are also sometimes used for more precise segmentation, as introduced in Section 2.3.

After segmentation, each segment of audio is required to be transformed into a vector containing speech or speaker-related information. For general speech information extraction, the relationship between frequency and power of sound waves is considered.

Based on that, to focus specially on speaker-related features, such relationship and relative conversion needs to be suitable for human hearing ability, known as Mel Frequency Cepstral Coefficient (MFCC) [1]. Besides MFCC, machine learning-based methods such as x-vector [45], d-vector [46], ECAPA-TDNN [9], and TitaNet [22], are widely used for more robust and efficient speaker-related feature extraction. Recently, with the development of Transformer models, particularly word embedding models like BERT, similar ideas have been adopted to the speech processing, resulting in Transformer-based feature extraction models like Wav2Vec2 [2] and HuBERT [17].

As the final stage, clustering groups the extracted feature vectors according to different speakers. At this stage, multiple clustering algorithms can be used including K-Means [29] and Gaussian Mixture Model [41]. To address the problem of clustering audio with an unknown number of speakers, Spectral Clustering [48] and Agglomerative Hierarchical Clustering [37] are more frequently used than other algorithms.

In addition to clustering, supervised models have also been developed as alternatives, taking advantage of the newly developed large neural networks. For such a classification model, it is important to train the model in a permutation-invariant manner, meaning that changing the sequence of output speaker labels should not alter the final diarization result as long as the grouping of segments remains unchanged. Typical permutation-invariance is achieved by arranging speaker labels according to their sequence of occurrence, referred to as Discriminative Neural Clustering (DNC) [27], or by training with a specially designed permutation-invariant objective function, which is often used in End-to-end Neural Diarization (EEND) [16, 25]. With the development of end-to-end systems and neural networks, hybrid systems that combine unsupervised and supervised models, such as the Multi-scale Diarization Decoder (MSDD) [34] or the Two-stage OverLap-aware Diarization framework (TOLD) [47], have also been proposed, achieving new state-of-the-art results.

2.2 Utilization of Text-based Features

Text-based features, mainly word choice, semantic features, and logistical cues among sentences, can be used as supplementary or even the sole features for SD. One way to utilize text features is by directly incorporating them into the SD process. In this way, text features helped normal SD system in better identifying speaker role information or characteristics. This is extremely useful in task-oriented or domain-specific conversations like Air Traffic Control (ATC) communications and psychological treatments. Specifically, the ATC scenario was tested with BERTraffic [51], wherein text-based Speaker Role Detection (SRD) and SD were treated as a token-level segmentation and classification task, performed by a finetuned BERT-base-uncased model. In the medical domain, SRD was achieved either through a pretrained n-gram language model as a feature extractor for clustering [13] or via a finetuned BERT-base-uncased model to impose constraints on clustering [12]. This approach, involving the direct use of text-based features, though theoretically promising and successful in some fields of study, still requires much more experimentation across a wider range of conversational topics and with newly developed language models.

With the development of language models and their applications in various Natural Language Understanding tasks, an approach has emerged that identifies and corrects unreasonable points in the diarization results from audio-based systems. This is mainly aimed at fixing issues that arise during the process of aligning audio-based SD results to transcripts, or similar issues occurring during segmentation, where the beginning or end of sentences is aligned with incorrect speakers. Various language models, especially Transformer-based models such as RoBERTa-base [36], a GPT-like structured model with 2B parameters [35], and PaLM2 [49], have been tested with this approach, achieving significant improvements over the audio-based systems to which they were compared. Although this approach does utilize the latest advancements in language models and a larger, more diverse range of data, the participation of

text-based features is not as pronounced as in the direct usage approach, thus not fully unleashing the potential of language models in utilizing text-based features.

2.3 Speaker Change Detection

Based on the previous sections, it is clear that segmentation plays an important role in both audio-only and multimodal SD systems, positioning the Speaker Change Detection (SCD) task as vital for SD. Furthermore, SCD itself is also beneficial for understanding conversations and can be considered an SD system if the conversation consists of only two speakers. Thus, it is necessary to review the past works on SCD, especially text-based approaches.

Similar to SD, SCD has been dominated by audio-based approaches due to the clear advantages of audio features, with the exception of Li et al., who leveraged text information along with uniform segmentation and GMM clustering [28]. Text-based SCD began with Meng et al., who introduced a sentence-level attention layer and a hierarchical RNN network [30]. After that, although audio-based systems continued to be the mainstream solution, text-dependent approaches began to draw attention with the development of both sequence-to-sequence neural networks and word embedding models. A multimodal model that adopted an encoder-decoder architecture with early fusion of text and audio embeddings was also proposed, achieving new state-of-the-art results [50]. However, fully text-based SCD approaches, such as those by Meng et al., still lack sufficient study.

2.4 Metrics

For the evaluation of SD, the Diarization Error Rate (DER) [11] is the standard metric reported in previous studies. It evaluates SD system performance by considering the fraction of time duration of incorrect speaker labels (the lower the DER score, the

better the performance). To better understand the specific types of errors in SD systems, DER can be broken down into four components: (a) Speaker Error E_{spk}: the fraction of time during which an incorrect speaker label is provided, compared to the ground truth label. (b) False Alarm E_{fa}: fraction of duration that a speaker label is given when there is no speech at all. (c) Missed Speech E_{ms}: fraction of duration that no speaker label is given when there is speech. (d) Overlap E_{overlap}: the fraction of time during which multiple speakers appear in the same segment and are not all correctly identified. The total DER score can be computed as:

$$\textbf{DER} = E_{\text{spk}} + E_{\text{fa}} + E_{\text{ms}} + E_{\text{overlap}} \tag{2.1}$$

Note that, by convention, E_{overlap} is often counted as E_{ms} or directly ignored, as correctly identifying overlap is generally considered a challenging task for SD systems. These four separate parts can also be computed together by the following equation:

$$\textbf{DER} = \frac{\sum_{s=1}^{S} dur(s) \cdot [max(N_{ref}(s), N_{hypo}(s)) - N_{correct}(s)]}{\sum_{s=1}^{S} dur(s) N_{ref}(s)} \tag{2.2}$$

where S represents the total number of segments included in the calculation, $dur(s)$ denotes the time duration of a single segment s, $N_{ref}(s)$ is the number of speakers in the ground truth transcript for the segment s, $N_{hypo}(s)$ represents the number of speakers predicted by the SD system for the segment s, and $N_{correct}(s)$ is the number of speakers accurately predicted by the SD system for the segment s.

To evaluate SD on text transcripts, the Word Diarization Error Rate (WDER) [32, 43] was proposed to measure inaccuracies of joint ASR and SD systems at the word level. The WDER can be calculated as follows:

$$\textbf{WDER} = \frac{S_{\text{IS}} + C_{\text{IS}}}{S + C} \tag{2.3}$$

where S and C are the number of ASR substitutions and correct words, S_{IS} and C_{IS} are the portions with incorrect speaker labels for S and C. It is important to note, as suggested by El Shafey et al. [43], that the standard ASR-specific metric, the Word Error Rate (WER) [21], is also required when evaluating WDER, since WDER does not account for insertion and deletion errors from ASR systems. The WER is calculated as:

$$\textbf{WER} = \frac{S + D + I}{N} \tag{2.4}$$

where S represents the number of word substitutions, D represents the number of deletions, I represents the number of insertions, and N represents the number of total words in reference (ground truth) transcript.

2.5 Sequence Alignment Algorithm

The sequence alignment problem is a common task in the realm of bioinformatics, where sequences of DNA, RNA, and amino acids must be aligned to determine their similarities in order to analyze their functional or evolutionary relationships. In speech and conversational data processing, due to differences between reference (ground truth) and hypothesis (model-generated) data, token-level alignment is also necessary for similarity detection, label matching, and semantic analysis. The analogous needs for alignment tasks in bioinformatics and computer science make it useful to review and examine the algorithms used in both fields.

In the field of bioinformatics, sequence alignment is often performed using dynamic programming-based algorithms with heuristics. Needleman-Wunsch algorithm [31] and Smith-Waterman algorithm [44] are two methods often used in alignment based on dynamic programming. Specifically, the Needleman-Wunsch algorithm adopts a method similar to the dynamic programming solution for the Longest Common Subsequence (LCS) problem. A two-dimensional table is first populated based on the

best-aligned solution for each sequence position, using a predefined scoring function as heuristics. Then, a backtracking algorithm is employed to follow the highest-scoring path and reconstruct the optimal global alignment. Compared to the Needleman-Wunsch algorithm, the Smith-Waterman algorithm changes the scoring rules by setting the lower limit of any cell in the scoring table to 0. In this way, Smith-Waterman shifted the focus from global alignment to local alignment.

Beyond bioinformatics, algorithms related to sequence alignment are also explored within the field of Natural Language Processing, particularly in monolingual word alignment. Rather than focusing solely on the actual content of the sentence, this task emphasizes aligning words with high semantic similarities. In contrast to dynamic programming, neural network-based models are used to gain a better understanding of word-level semantics. Although only a few studies have been conducted on this specific task, Jiang et al. [20] and Lan et al. [24] proposed hybrid neural network and Conditional Random Field (CRF) models to address it. They formulated word alignment as token classification tasks, achieving state-of-the-art results and providing corresponding datasets for training and evaluation.

Chapter 3

Text-based Speaker Diarization

In this chapter, the model and data processing pipeline for text-based SD is introduced. Specifically, two models, the Single Prediction Model (SPM) in Section 3.2 and Multiple Prediction Model (MPM) in Section 3.3, as well as the data processing pipeline for preparing specialized data for doing SD on ASR-generated transcript in Section 3.4 are discussed.

3.1 Task Overview

This thesis focuses on the task of text-based SD for two-speaker conversations specifically. These conversations are not only prevalent but also carry practical importance, serving as a basis to showcase the effectiveness of the text-based SD approach. Additionally, the strategies developed for handling conversations between two speakers can be readily adapted for scenarios involving multiple speakers, requiring only minor modifications. This adaptability is further elaborated in the Appendix.

To solve for two-speaker conversation SD, this thesis employ a strategy that centers on Speaker Change Detection (SCD) at the sentence level. This sentence-level approach is chosen over a word-level one, unlike most of the previous works introduced in Chapter 2, due to its ability to capture richer contextual information, which

significantly enhances the accuracy of SD. Moreover, this finer level of granularity effectively mitigates the issue of label imbalance often encountered in word-level diarization, where most labels will be 'unchanged,' as a single sentence typically contains multiple words. This approach provides a more evenly distributed dataset for both training and evaluating the model.

3.2 Single Prediction Model

In order to predict the speaker changes, the model needs to take the sentences before and after the point of prediction as the input. The model can then be trained with a binary classification objective about "change" or "unchange". In this way, the task can be formally defined as: Consider a sequence of sentences within a dialogue, denoted as $S = \{s_1, s_2, ..., s_n\}$, comprising n individual sentences. The goal is to assign a binary label y_i to every adjacent pair of sentences (s_i, s_{i+1}) within this sequence. Here, $y_i = 1$ signifies that there is a change in the speaker from sentence s_i to sentence s_{i+1}, whereas $y_i = 0$ indicates that the speaker remains the same. Consequently, the outcome of this prediction task is represented as a sequence of binary labels $R = \{y_1, y_2, ..., y_{n-1}\}$, corresponding to speaker changes across the sequence. This sequence of predictions, R, facilitates the reconstruction of the dialogue's speaker structure.

To achieve such a task, the model should be able to take a sequence of word or sentence embeddings as input and provide a binary output as a prediction. Arbitrary number of sentences can be used as the input as long as the point of prediction is within the sentences and labeled with special token. Formally, if the point of change is between the pair of sentences (s_i, s_{i+1}), the input has h sentences preceding and k sentences following the sentence s_{i+1} to create a context set C_i. This set for the i-th prediction comprises $\{s_{i-h+1}, s_{i-h+2}, ..., s_i, s_{i+1}, ..., s_{i+k+1}\}$. The binary prediction y_i for each sentence pair is generated through the function f, utilizing the context C_i and

model parameters θ, where f can be any suitable model designed to process sequential data and output a binary decision. The training objective is to reduce the binary cross-entropy loss L across all instances of y_i, optimizing the model's performance in accurately predicting speaker changes.

$$L(\theta) = -\frac{1}{n-1}\sum_{i=1}^{n-1}[y_i \log(f(C_i,\theta)) + (1-y_i)\log(1-f(C_i,\theta))] \tag{3.1}$$

In real time prediction, the SPM acts like a sliding window that sweeps through the entire transcript from begin to end, with the first sentence set to s_i and shift one sentence forward every time until s_{i+1} becomes the last sentence. At the beginning and the end of conversation, the number of historical and future sentences may not be enough to meet the parameter for h and k, which is acceptable in order to make sure each point of prediction is covered. To deal with insufficient sentences within the segment of $\{s_{i-h+1}, s_{i-h+2}, ..., s_i\}$ and $\{s_{i+2}, ..., s_{i+k+1}\}$, the model must be trained with data under these situations.

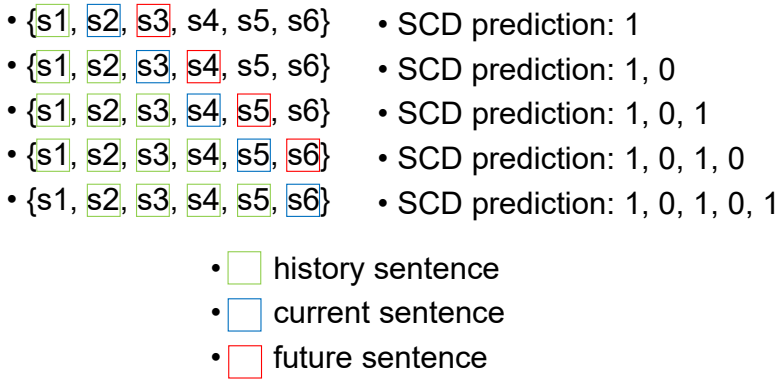

Figure 3.1: Demonstration of SPM prediction on a conversation with 6 sentences. The parameters in this demonstration are set to $h = 4$ and $k = 1$.

3.3 Multiple Prediction Model

Using sentence-level SCD for two-speaker SD requires significant amount of robustness for each prediction of speaker change. One incorrect prediction of changing speakers may cause the entire trailing predictions of speakers to be flipped.

- SPM SCD prediction: 1, 0, 1, 0, 1
- Correct result: 1, 0, 1, 1, 1
- Correct speaker label: A, B, B, A, **B**, **A**
- Predicted speaker label: A, B, B, A, **A**, **B**

While straightforward, the SPM approach is error-prone due to its reliance on a limited contextual window with only one prediction at each point. Even if only one speaker change is indirectly predicted, the whole sub-sequence of speaker labels after the incorrect change prediction is completely incorrect. To enhance the robustness of SCD, leveraging the flexible output format of sequence-to-sequence models, this thesis also introduces the Multiple Prediction Model (MPM).

Compared with SPM, MPM also makes predictions on different subsequence of input, but produces multiple predictions of speaker change for every point of change. This is achieved by training the model to produce predictions between every adjacent sentence pair. For the whole conversation, a sliding window technique that shifts through the sequence of sentences with certain amount of overlap produces different subsequences from begin to end. Finally, all predictions are aggregated together via specific aggregation mechanism.

Consider $W = \{w_1, w_2, ..., w_m\}$ to represent a series of segments, with each segment w_j containing a subset of sentences from the sequence S, and m indicating the total count of such segments spanning the entire conversation. These segments are designed to partially overlap with both their preceding and succeeding segments. The aim is to

generate a series of binary predictions y_i for each segment w_j, where every prediction within y_i indicates a potential change in speaker occurring between two successive sentences within the segment w_j. The prediction associated with a specific segment w_j is computed as $y_j = g(w_j, \phi)$, where the function g operates similarly to f, designed for processing sequences to yield binary outcomes, and ϕ is the parameters tailored for this multiple prediction scenario. The training process focuses on minimizing an adjusted loss function L', which accommodates the nuances of making multiple predictions within each segment.

$$L'(\phi) = -\frac{1}{\sum_{j=1}^{m} |w_j| - 1} \sum_{j=1}^{m} \sum_{i=1}^{|w_j|-1} [y_{ji} \log(g(w_j, \phi)_i) + (1 - y_{ji}) \log(1 - g(w_j, \phi)_i)] \quad (3.2)$$

where each y_{ji} is the i-th binary label within the set y_j, and $g(w_j, \phi)_i$ denotes the i-th binary label produced by the function g for the segment w_j.

Through producing predictions for every adjacent sentence pair in each single subsequence and allow overlapping among different subsequences, there are multiple predictions for the same point of change that is generated with different input context. An aggregation strategy can be used to utilize these predictions to enhance the robustness of final predictions of speaker change. For any prospective speaker change point p, the final outcome Y_p can be deduced by using predictions from all input containing point p as follows:

$$Y_p = Aggregate(\{g(w_j, \phi)_p | p \in w_j\}) \quad (3.3)$$

Under most of the circumstances, the majority vote can be used as an effective strategy to produce robust results. For more complex situations, such vote can also be enhanced with weighted average, using the confidence score from the last hidden layer or other equivalent scores.

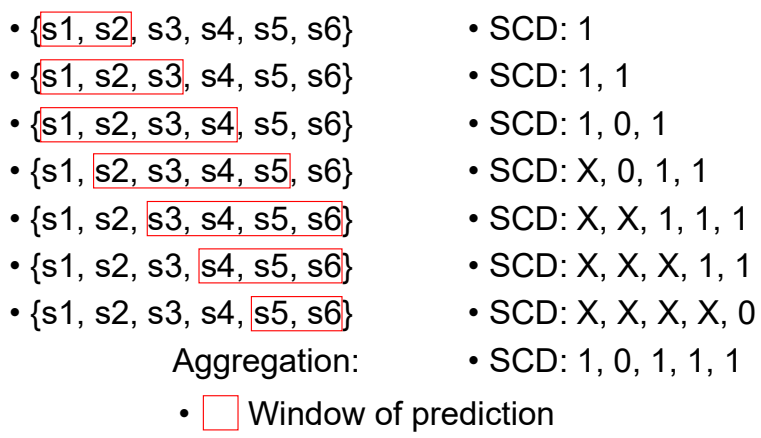

Figure 3.2: Demonstration of MPM prediction on a conversation with 6 sentences. The maximum window size is 4 sentences, with 1 sentence shift each time. In this demonstration, the majority voting is used for aggregation.

- MPM SCD prediction: 1, 0, 1, 1, 1
- Correct result: 1, 0, 1, 1, 1
- Correct speaker label: A, B, B, A, B, A
- Predicted speaker label: A, B, B, A, B, A

In real world predictions, similar to SPM, MPM also works in a sliding window way. To maximize the information utilized in each input and aggregation mechanism, both the length of each window and the overlap between windows should be maximized. The begin and end windows are also required to make predictions under conditions that number of sentences is lower than the maximum as SPM and need to be trained specifically. In Figure 3.2, the aggregation mechanism (majority voting) successfully corrects the errors at second and fourth position of change predictions.

3.4 Data Processing

To prepare data well-suited for the model, it is vital to focus on the primary situations in which the model will operate. As stated in Chapter 1 about the increasing need

for joint ASR and SD tasks, it is most realistic to train the model on datasets that contains characteristics of ASR-generated scripts.

To mimic such environment, the transcripts used for training and evaluation should also come from ASR systems, but with ground truth labels for each word in the generated transcript so that text-based evaluation can be done. This can be achieved by aligning the ASR-generated transcript with the reference transcript, which has ground truth speaker labels, thereby creating optimal one-to-one mappings for each word in the transcript. By doing so, the ground truth speaker labels are mapped to the ASR-generated transcript, creating the new training and evaluation data for text-based model. Such mapping can be done with sequence alignment algorithm like Needleman-Wunsch algorithm or Smith-Waterman algorithm introduced in Section 2.5.

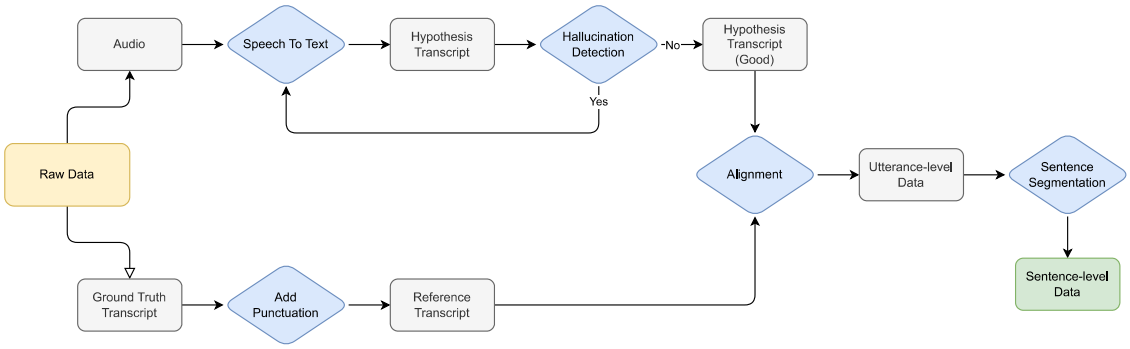

Figure 3.3: Data processing pipeline utilizing ground truth transcripts and ASR-generated hypothesis transcript to obtain training and evaluation data with ASR discrepancies.

However, considering the potential errors brought by ASR in overlapping speech segments, where words spoken by different speakers might blend together, neither of the aforementioned algorithm can accommodate such condition. To achieve cross alignment while also considering other common types of ASR errors (i.e. insertion, deletion, and substitution), a novel dynamic programming-based multiple sequence alignment algorithm is used. The detail about this algorithm is introduced in Section

4.2.

Other than using alignment algorithms that can handle ASR errors, the ASR system itself is also required to be the state-of-the-art to minimize the amount of errors. This usually involves systems with sequence-to-sequence models, such as OpenAI Whisper [39] or Meta SeamlessM4T [8], which are prone to hallucination with unclear input audio. Therefore, detection and correction mechanisms are necessary as a post-processing stage for ASR systems.

Besides audio processing, the reference (ground truth) transcript may also need adjustments. For old datasets with non-standardized coding of files, the reference transcript itself might lose its punctuation, making evaluation and alignment challenging. To solve this, all transcripts that are missing the whole or partial of the punctuation are fixed with LLM, as LLM are trained on massive datasets with correct grammar and are good at detecting grammatical errors with producing relative fixes.

Finally, after alignment with punctuation-fixed reference transcripts and ASR-generated non-hallucinated hypothesis transcripts, the aligned transcripts are segmented on the sentence-level to provide enough semantic features for the model. Through this process, the transcripts with ASR discrepancies and ground truth speaker labels are produced for both training and evaluation.

Chapter 4

Text-based Speaker Diarization Evaluation

This Chapter introduces the development of Text-based Diarization Error Rate (TDER) and Diarization F1 (DF1) as two comprehensive metrics for evaluating text-based SD as well as joint ASR and SD tasks. The Multiple Sequence Alignment (MSA) algorithm that is required for evaluation and calculating metrics are also introduced. The main content of this section has been published at [14]. The text-based metrics and MSA algorithm were developed in collaboration with Chen Gong.

4.1 Text-based Metrics

The audio-based metrics introduced in Section 2.4 exhibit the following drawbacks when evaluating text-based SD and joint ASR and SD tasks:

1. Audio-based metrics like DER are not designed to be compatible with textual data since the basis of audio data is continuous in terms of time, whereas the textual data is discrete with the minimal unit of word, token, or character.

2. Current text-based metrics like WER and WDER do not account for all types

of possible errors that may occur in text-based SD or joint ASR and SD tasks. Specifically, WER as a metric for evaluating ASR systems did not contain terms for SD errors; WDER, though designed to evaluate SD on textual data, did not consider the insertion and deletion errors from ASR systems, which may also affect the SD performance.

Based on these considerations, we propose two metrics designed to comprehensively evaluate text-based SD and joint ASR and SD tasks, aiming to address the identified issues.

4.1.1 Text-based Diarization Error Rate

To evaluate the SD systems on transcripts while maintaining a similar behavior as the previous wide-adopted DER metric, and for convenient comparison between audio-based SD systems and text-based SD systems on the same transcript, the Text-based Diarization Error Rate (TDER) is proposed as an adaptation of the original DER to textual data. This adaptation is achieved by changing the duration's basis from time to the portion of the transcript, with the word serving as the unit. The TDER can be expressed as:

$$\textbf{TDER} = \frac{\sum_{u=1}^{U} len(u) \cdot [max(N_{ref}(u), N_{hypo}(u)) - N_{correct}(u)]}{\sum_{u=1}^{U} len(u) N_{ref}(u)} \tag{4.1}$$

where U is the transcript consisting of a series of utterance u, $len(u)$ is the number of words in utterance u, $N_{ref}(u)$ is the number of speakers in the ground truth transcript for the utterance u, $N_{hypo}(u)$ is the number of speakers predicted by the SD system for the utterance u, and $N_{correct}(u)$ is the number of speakers predicted correctly by the SD system for the utterance u. Since the change only affects the method of calculating the fraction and duration, TDER can be divided into four components (E_{spk}, E_{fa}, E_{ms},

E_{overlap}) and calculated in the same manner as Equation 2.1:

$$\textbf{TDER} = E_{\text{spk}} + E_{\text{fa}} + E_{\text{ms}} + E_{\text{overlap}} \tag{4.2}$$

with the fraction of duration calculated as the fraction of words.

Compare to the original DER, the TDER remains its behavior because of its unchanged way of calculation, as the duration in terms of words is analogous to duration in terms of time. Therefore, TDER can be directly used for audio-based SD systems with simple mapping performed with timestamp provided by the ASR system or the ground truth transcript for performance comparison, which provides convenience for comparing text-based approaches to audio-based approaches. It is also worth noting that TDER can be seen as a generalization of WDER as WDER only considers errors from incorrect speakers, meaning that

$$\textbf{WDER} = E_{\text{spk}} \tag{4.3}$$

This also means that TDER and WDER can be used interchangeably if every word in hypothesis and reference transcripts has a speaker label, meaning that the portion of E_{fa}, E_{ms}, and E_{overlap} are 0.

4.1.2 Diarization F1

While TDER resolves the problem for making convenient comparisons with respect to the same transcript, for joint ASR and SD task, it is also required to make comparisons based on different ASR-generated transcripts, which is unachievable by modifying the previous metrics. To comprehensively evaluate 'who speaks what' outcomes, we propose the Diarization F1 (DF1) metric. DF1 consists of two parts, the precision

and recall, calculated as follows:

$$\textbf{Precision} = \frac{\textbf{align\_speakers}(T_{hypo}, T_{ref})}{len(T_{hypo})} \tag{4.4}$$

$$\textbf{Recall} = \frac{\textbf{align\_speakers}(T_{hypo}, T_{ref})}{len(T_{ref})} \tag{4.5}$$

$$\textbf{DF1} = \frac{2 \cdot \textbf{Precision} \cdot \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{4.6}$$

where T_{hypo} is the hypothesis transcript from ASR, T_{ref} is the reference transcript with ground truth labels, **align_speakers**() is the function providing alignment between two input transcripts and return the number of aligned words with same speaker labels. In this way, the **Precision** represents the fraction of words with correctly predicted speaker label to the hypothesis transcript, with the **Recall** representing the fraction of words with correctly predicted speaker label to the reference transcript. Finally, the DF1 is calculated as the harmonic mean of **Precision** and **Recall** like the usual F1-score. Through aligning transcripts, a fair and comprehensive performance comparison between the result from a joint system of ASR and SD can be compared with the ground truth transcripts by locating the words without ASR errors through alignment at first, and further locating the words free from SD errors through comparing speaker labels.

4.2 Aligning Transcripts

4.2.1 Current Alignment Limitations

Due to the nature of conversational transcripts and ASR system, which should faithfully convert every possible words sequentially, the alignment between the hypothesis and reference transcript is required to be one-to-one and the majority of words at the same position in both hypothesis and reference transcripts are largely the same in terms

of spelling. As introduced in Section 2.5, both the dynamic programming-based and machine learning-based approaches can be used in aligning sequences of transcripts. Between these two approaches, the machine learning based one, though might have the advantage in aligning according to the actual meaning of words, failed to utilize the high spelling and positional similarity of the words. Furthermore, aligning just with semantic meanings may introduce correct but unwanted non-one-to-one mappings, which over complicates the problem.

The dynamic programming based approach, on the contrary, considering only the position and the spelling of the words for one-to-one alignment, is a better approach in the task of aligning ASR generated transcripts and ground truth transcript for comparing speaker labels. However, the flexibility of dynamic programming-based approaches is uncomparable to the machine learning-based approaches, especially in handling overlapping situations where the words from different speakers are mixed together. In addition, the original dynamic programming algorithms are designed to align on the character-level, such as the four types of nucleotide in DNA sequence and twenty types of amino acids in protein. This characteristic makes it challenging for dynamic programming algorithms to accommodate minor spelling mistakes in single words during ASR transduction. Such minor mistakes might still be classified as correct in mapping, given their often tolerable nature.

Reference Transcript
A: You're going to go to uh Emory.
B: Indeed, indeed.

Hypothesis Transcript
A: You're gonna to go to indeed indeed Emory.

Figure 4.1: An example of normal transcript that need to be aligned together.

To increase the flexibility of original dynamic programming based algorithms, this

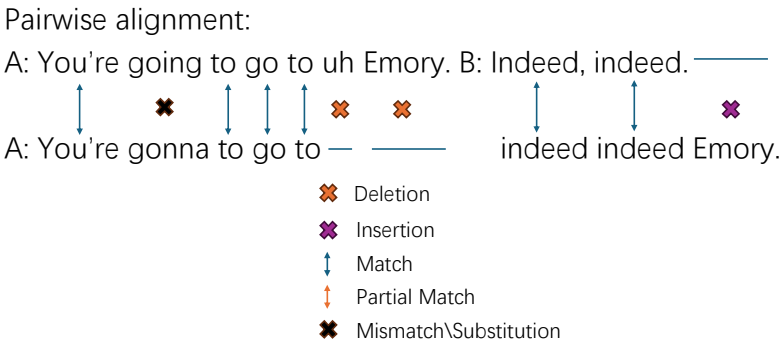

Figure 4.2: Transcripts aligned with pairwise alignment (original Needleman-Wunsch algorithm). The 2 dimensional alignment cannot handle the overlapping situation in dialogue. The character-based alignment also cannot do partial matching between words with different spelling but same position.

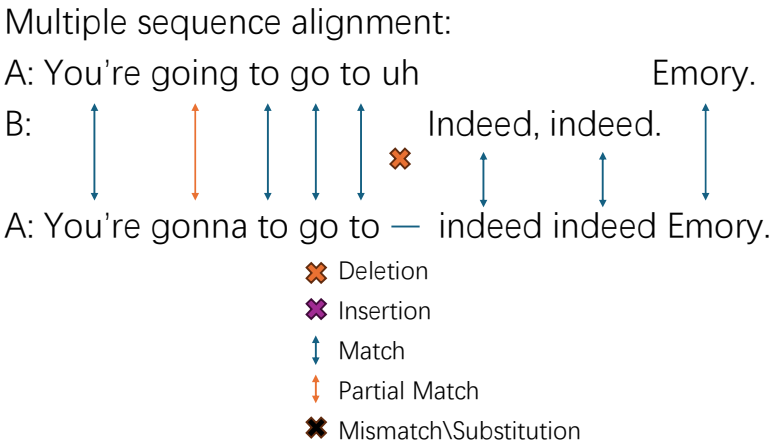

Figure 4.3: Transcripts aligned with multiple sequence alignment. The 3 dimensional alignment considers the overlapping situation. The token-based alignment matches between words with different spelling with Levenshtein Distance.

thesis proposes a Multiple Sequence Alignment (MSA) algorithm based on the original Needleman-Wunsch algorithm. Our improvements are in two parts: (a) Extending the Needleman-Wunsch algorithm to handle more than two sequence to provide correct alignment in overlapping segments. (b) Introducing Levenshtein Distance as the criterion for populating scoring table in the MSA to make the algorithm tolarate the minor spelling mistakes. The main content of this chapter has been published at [14].

4.2.2 Scoring Matrix Population

Our MSA remains the same two-stage structure as the Needleman-Wunsch algorithm. The MSA requires the reference transcript to be separated into multiple sequences according to the speaker label. Formally, denote $X = [x_1, \ldots, x_\ell]$ as a sequence that is formulated by enumerating every word within the hypothesis transcript, without consideration for any division into segments. For each speaker j, define $Y_j = [y_{j1}, \ldots, y_{jm}]$ as a sequence enumerating the tokens associated with Speaker Y_j in the reference transcript. Given a set $E = [X, Y_1, \ldots, Y_n]$, the initial step of the algorithm involves constructing a scoring matrix F. This matrix is a multidimensional array, with its dimensions shaped by the lengths of the input sequences, and initially, all entries of this matrix are set to zero. The pseudocode for the population of scoring table is in Algorithm 1.

Algorithm 1: Populating Scoring Matrix

Input : $E = \{X, Y_1, \ldots, Y_n\}$
Output : Matrix F

Create $F \in \mathbb{R}^{(|X|+1)\times(|Y_1|+1)\times\cdots\times(|Y_n|+1)}$;
$C \leftarrow [\gamma \subset \{0, 1, \ldots, n\}] \setminus \varnothing$;
foreach $\gamma \in C$ **do**
    foreach $\psi \in \mathit{index\_perm}(\gamma, E)$ **do**
        $F_\psi \leftarrow \mathit{score}(\psi, E, F)$;
    end
end
return F;

Once the scoring matrix is declared and initialized with zero (`L1`), a list including all combinations of $\{0, .., n\}$ expect for the empty set (`L2`) is then generated with the ascending order of the number of elements in the combination and the number itself. It is important to keep such ascending order as the computation for higher number combinations requires the result from lower number combinations. For example, for a conversation with 2 speakers, the list is first filled with all the combinations with only 1 element

$$[\{0\}, \{1\}, \{2\}]$$

then filled with all the combinations with 2 elements

$$[\{0, 1\}, \{0, 2\}, \{1, 2\}]$$

finally filled with all the combinations with 3 elements

$$[\{0, 1, 2\}]$$

In the context of a combination, the numerals symbolize the input sequences, with 0 signifying X and any i, where $i > 0$, denoting Y_i. For each distinct combination γ, alongside E, they are input into the *index_perm* function. This function outputs a collection of index tuples (`L3-4`). These tuples are formulated such that they contain indices corresponding to the relevant sequences, while indices for the remaining sequences are maintained at 0. In the case of 2-speaker conversation, the results for *index_perm* function are shown in Table 4.1.

For each tuple $\psi = (i, j, \ldots, k)$ indicating the indices of each sequence, where i signifies the position of x_i within X, j represents the position of y_{1j} in Y_1, and k the position of y_{nk} in Y_n, the *score* function evaluates the scores from all preceding cells

γ	*index_perm*(γ, E)	**Size**
$\{0\}$	$[(1,0,0),..,(\|X\|,0,0)]$	$\|X\|$
$\{1\}$	$[(0,1,0),..,(0,\|Y_1\|,0)]$	$\|Y_1\|$
$\{2\}$	$[(0,0,1),..,(0,0,\|Y_2\|)]$	$\|Y_2\|$
$\{0,1\}$	$[(1,1,0),..,(\|X\|,\|Y_1\|,0)]$	$\|X\| \cdot \|Y_1\|$
$\{0,2\}$	$[(1,0,1),..,(\|X\|,0,\|Y_2\|)]$	$\|X\| \cdot \|Y_2\|$
$\{1,2\}$	$[(0,1,1),..,(0,\|Y_1\|,\|Y_2\|)]$	$\|Y_1\| \cdot \|Y_2\|$
$\{0,1,2\}$	$[(1,1,1),..,(\|X\|,\|Y_1\|,\|Y_2\|)]$	$\|X\| \cdot \|Y_1\| \cdot \|Y_2\|$

Table 4.1: Permutations of index tuples as the result for *index_perm* function in the case of 2-speaker conversation. The sequence is from top to bottom which cannot be violated in order to keep the correctness of MSA algorithm.

directly adjacent to x_i, including:

$$\{(i-1,j,..,k),(i,j-1,..,k),\ldots,(i,j,..,k-1)\}$$

as well as those diagonally prior to x_i, such as:

$$\{(i-1,j-1,..,k),\ldots,(i-1,j,..,k-1)\}$$

Based on this, it calculates the score for F_ψ by considering the aforementioned positions (L5).

$$\begin{aligned} F_{i,j,..,k} \quad &\leftarrow \quad \max(\mathcal{G}(E,F,(i,j,..,k))) \\ \mathcal{G}(E,F,\psi) \quad &\leftarrow \quad \begin{cases} F_{i-1,j,..,k} & +match(x_i) \\ F_{i,j-1,..,k} & +match(y_{1j}) \\ \qquad\vdots & \\ F_{i,j,..,k-1} & +match(y_{nk}) \\ F_{i-1,j-1,..,k} & +match(x_i,y_{1j}) \\ \qquad\vdots & \\ F_{i-1,j,..,k-1} & +match(x_i,y_{nk}) \end{cases} \end{aligned} \tag{4.7}$$

where the match function represents the heuristics for each single alignment between words. The match function considers four possibilities, fully match, partially match, mismatch, and gap, with the scores given in a descending order, meaning that the score of fully match must be greater or equal to partially match, and the score of partially match must be greater or equal to mismatch. The *match* function can be formally written as:

$$match(x, y) \leftarrow \begin{cases} S_{fm} & \text{if } LD(x, y) = 0 \text{ (fully match)} \\ S_{pm} & \text{if } LD(x, y) \leq d \text{ (partial match)} \\ S_{mm} & \text{if } LD(x, y) > d \text{ (mismatch)} \\ S_{gap} & \text{if only one word is given (gap)} \end{cases}$$

where S_{fm}, S_{pm}, S_{mm}, S_{gap} represent the score for fully match, partially match, mismatch, and gap. The scores must be given in a descending order, meaning that $S_{fm} \geq S_{pm} \geq S_{mm} \geq S_{gap}$. If two words were given as the input, the *match* function will calculate the Levenshtein Distance (String Edit Distance) between them with function $LD(x, y)$. The greater the Levenshtein Distance, the larger the difference between two words, with 0 distance be exactly same spelling. To tolerate minor spelling mistakes, if the Levenshtein Distance is below certain boundary d, it may be counted as partially match, which will be rewarded with a higher score than mismatch.

4.2.3 Backtracking

After population of scoring matrix, the MSA algorithm then reconstructs the optimal alignment through backtracking from the very last cell of the matrix by tracing the highest score adjacent to the current cell with the direction towards the very first cell. The pseudo code for backtracking is shown in Algorithm 2.

This backtracking algorithm accepts the list of input sequences E and the scoring

Algorithm 2: Backtracking Scoring Matrix

Input : $E = \{X, Y_1, \ldots, Y_n\}$,
the scoring matrix F.
Output : The alignment matrix A

```
1 Create A ∈ ℝ^{|E|×ρ};
2 ψ ← (|X|, |Y_1|, ..., |Y_n|);
3 while ψ ≠ (0, 0, ..., 0) do
4 |   (ψ', α) ← argmax(G(E, F, ψ));
5 |   Append α to A accordingly;
6 |   ψ ← ψ';
7 end
8 return A;
```

matrix F returned by Algorithm 1 as the input, and returns the alignment matrix A as the output. The algorithm constructs A, where the 0'th row is populated with tokens from X and the i'th rows with tokens from Y_i or with gap tokens (`L1`). The exact count of ρ, which represents the total number of columns, remains indeterminate at this phase due to the unpredictable quantity of gap tokens required for optimal alignment, a detail that only becomes clear upon the conclusion of the backtracking process. Therefore, $\rho = \max(|X| + g_x, |Y_i| + g_i : \forall i)$ signifies the number of columns, where g_x and g_i represent the count of gap tokens introduced to achieve the optimal alignment between X and each Y_i. This backtracking operation initiates at the matrix's final cell, marked by ψ (`L2`). Following this, it selects a cell (`L4`) through the *argmax* function, which identifies the index tuple ψ' and the sequence of tokens α that elevate the alignment score to its peak ($|\alpha| = |E|$). Within α, the $0|i$'th element corresponds to either the current token in $X|Y_i$ or a gap symbol '$-$'. An example that the backtracking determines that the last sequence of Y should be a gap will look like:

$$
\begin{aligned}
A_0 &\leftarrow A_0 \oplus [x_i] \\
A_1 &\leftarrow A_1 \oplus [y_{1j}] \\
&\vdots \\
A_2 &\leftarrow A_n \oplus [-]
\end{aligned}
$$

After appending the words or gap, it moves to the subsequent cell identified by ψ' (L6). This step is repeated, moving backwards through the cells, until it arrives at the initial cell (L3). Figure 4.4 depicts the backtracking of a 2-speaker conversation case by Algorithm 2.

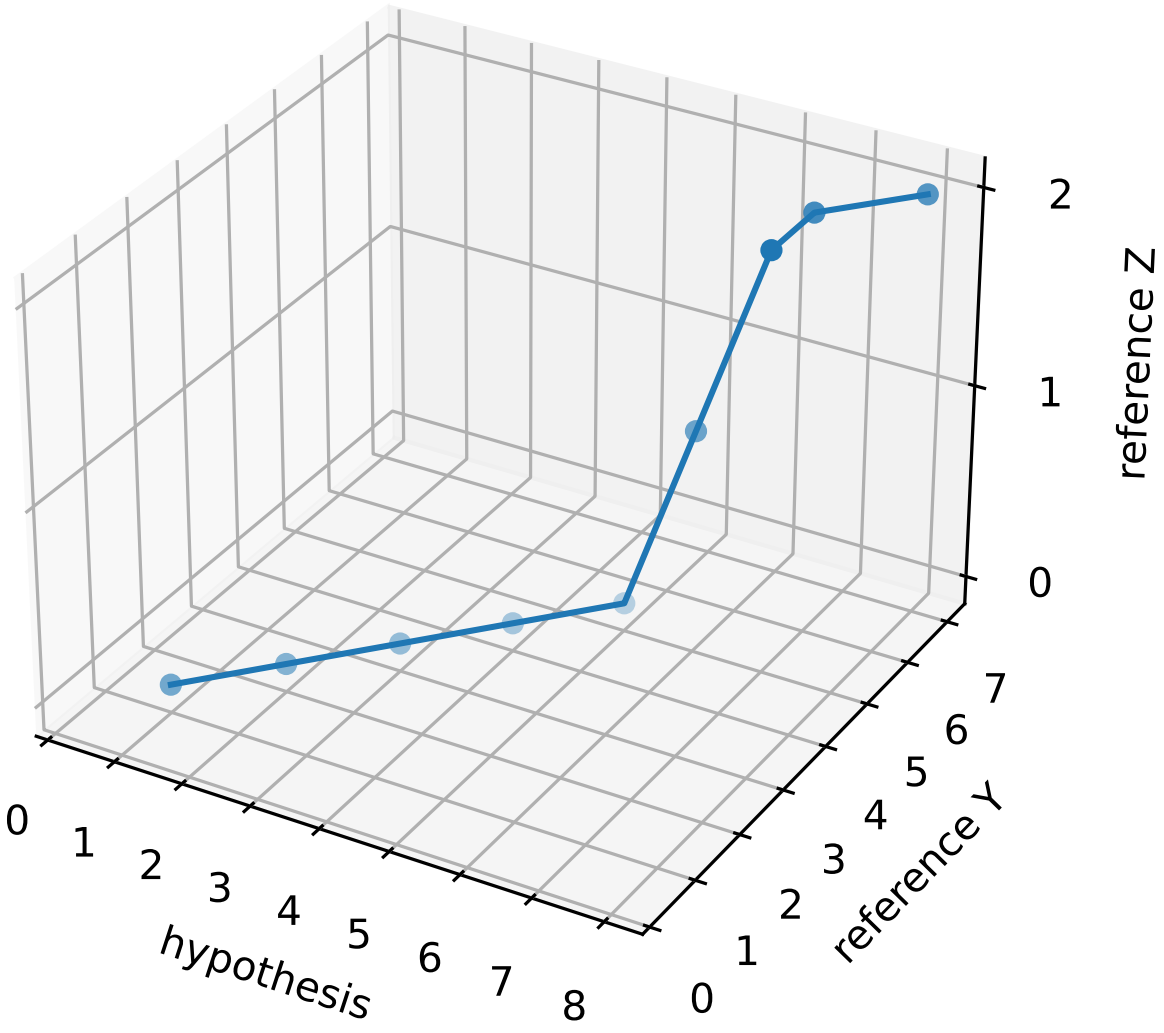

Figure 4.4: The backtracking example using Algorithm 2

4.2.4 Optimization of Matrix Population

Although effective, the current MSA becomes time and resource-intensive as the number of speakers and conversation length increase. This is because the time and space complexity for populating the matrix is $O(L^{n+1})$, where L represents the longest

sequence among $E = [X, Y_1, \ldots, Y_n]$, and n is the number of speakers. In order to reduce memory usage, in real world implementation, the MSA imports segmentation mechanism by detecting absolutely fully aligned segments as anchors or barriers. Then, the whole transcript is separated at the mid point of each anchor and perform alignment for each segment separately. The final alignment result is the concatenation of each segment's alignment result. In this way, the increase in memory usage for storing the gigantic scoring matrix and the computation for populating the matrix is controlled by reducing the length of the transcript.

Chapter 5

Experiments Setup

This chapter introduces the specific setups and parameters used in evaluating the text-based SD model and evaluation metrics. Section 5.1 introduces the data used for both parts of experiments.

5.1 Datasets

In order to accurately evaluate the performance of our text-based models and the behavior of the proposed metrics, this thesis uses a curated dataset consisting of 7 widely-adopted conversational corpora that include both audio and ground truth transcripts with speaker labels. The accompanied audio files allow us to compare our model with recent audio-based SD models. All 7 datasets are focused on daily conversational topics, which rules out the influence of domain-specific knowledge requirements for language models.

For training and evaluation of the text-based SD model in Chapter 3, the portion of the dataset featuring conversations with 2 speakers is used. The dataset is split into train, development, and test set with a ratio of 80:10:10. To prevent data leakage among three different sets, such splitting is done on the conversation level.

For text-based SD metrics evaluation, which requires in-depth examination of

Corpus	Hour	# of Dialogue
AMI Corpus [7]	100	171
CallFriend [5]	20	41
CallHome English [6]	20	176
CHiME-5 [3]	50	20
DailyTalk [26]	20	2541
ICSI Corpus [18]	72	75
SBCSAE [10]	23	60

Table 5.1: Corpora used for the curated dataset.

individual examples rather than quantity, only the CallHome English [6] dataset is used. Specifically, 10 conversations are selected for checking alignment effectiveness, and the entire corpus is used for metrics evaluation.

5.2 Text-based SD Approach Experiments

5.2.1 Data Processing

For introducing ASR specific discrepancies, the Whisper [39] model from OpenAI is used because of its superior speech-to-text performance up to the time of conducting this experiment. In terms of hallucination, Whisper mainly experiences two types of errors, which mainly happen at the audio segments with low fidelity or low loudness, especially for the segments in long audio:

1. The model repeats the last sentence before the low quality segments for multiple times.

2. The model transcribes the speech to languages other than English. This is often happened together with repetition of incorrectly transcribed sentences.

These errors can be easily detected through basic coding or through a Large Language Model (LLM). To avoid hallucination in re-transcribing audio, both the Whisper large-v2 and medium.en model are used in transcribing audio as they have similar

transcribing performance [39] but with different behavior of hallucination in terms of positions of segments. Despite switching models, tuning inference parameters, including temperature, number of beams for beam search, as well as fixing decoding languages, are also used to mitigate hallucinations. On the audio side, loudness normalization and separate transcription for difficult segments with low loudness and low quality. Through these techniques, all the audios are able to be transcribed successfully.

For aligning transcripts, the self-implemented **align4d** package is used, which is described in Section 5.3.3. OpenAI GPT-4[1] and spaCy[2] are used for fixing punctuation in ground truth transcript and for sentence segmentation after transcript alignment.

5.2.2 Model

In the experiments for both SPM and MPM, we utilize the T5-3B model [40] due to its adaptability regarding the format of inputs and outputs because of the nature of a sequence-to-sequence model. Besides, the bi-directional attention in the encoder part of T5 model makes it possible to utilize semantic features before and after the point of prediction simultaneously. Though there are a wide range of model size for choosing, the 3B parameter model is chosen for achieving balance between its performance capabilities and efficiency considering our training devices (Nvidia H100 GPU). The choice of the standard T5 model over its more specifically fine-tuned counterparts like FLAN-T5 is made because non of the tasks further fine-tuned on is similar to SD or SCD. As such, our approach does not incorporate instruction fine-tuning. We employ a majority voting mechanism as our aggregation method also for its balance in effectiveness and simplicity.

For the audio-based counterparts, both the recent modular and End-to-end Neural Diarization (EEND) approaches are involved in this experiment. The specific names

[1] https://chat.openai.com/
[2] https://spacy.io/

and types of approaches are listed in Table 5.2:

Model Type	Models
Modular Approach	pyannote [4, 38], x-vector+SC [45, 48], x-vector+AHC [45, 37], ECAPA+SC [9, 48], ECAPA+AHC [9, 37], NeMo-TitaNet [22]
End-to-End Neural Diarization	NeMo-MSDD [34], TOLD [47]

Table 5.2: Categorization of models based on their approach to speaker diarization. In this table, SC means spectral clustering and AHC means agglomerative hierarchical clustering.

5.2.3 Model Evaluation

To evaluate the audio-based SD system on transcripts, the speaker label for each word in the transcript is attributed according to aligning the audio-based model output, which are segments about speaker labels and time period, with the sentence-level timestamp provided by Whisper, which matches the behavior of our text-based sentence-level prediction. Then, all the audio-based SD systems are evaluated on the same ASR-generated transcript, which is also used for text-based SD prediction.

Under the aforementioned evaluation setup, our TDER and WDER are interchangeable as suggested by Equation 4.3. This is because all portion of E_{fa}, E_{ms}, and E_{overlap} are eliminated in time alignment for audio-based SD systems, and our text-based SD systems do not produce such errors as they always assign a speaker label for every sentence. Under this case, the DF1 score is also just 1 - TDER because of the same reason. For simplicity, the WDER is used as the name of our evaluation metric for the rest of SD approach evaluation results and analysis. Other than normal WDER, which is calculated as the average of all single WDER score for each conversation, we also introduce WDER-S, which is the weighted average of WDER using the number of sentences in each transcript as weight, as a supplemental metric, in order to accommodate the difference in length of conversations within our

dataset. Compared with normal WDER, WDER-S reflects the performance on long conversation more accurately.

5.3 Text-based Metrics Experiments

5.3.1 Metrics Behavior

In order to experiment the behavior of metrics on text-based SD and joint ASR and SD systems, Amazon Transcribe[3] and Rev AI[4], two popular speech-to-text systems with speaker label provided are tested with both traditional metrics (DER, WER, WDER) and newly proposed TDER and DF1 metrics. Before calculating the metrics, the hypothesis and reference speaker labels are first applied Hungarian algorithm [23] to find optimal assignment to ensure that the upper limit of the performance is shown. After speaker label assignment, the hypothesis transcripts are evaluated both on the percentage of each type of joint ASR and SD errors (substitution, insertion, deletion, overlapping) and manually inspected to determine the behavior or each joint ASR and SD system. Finally, these systems' behavior are compared with the metrics to see if the metrics behavior correctly reflecting the systems' behavior. For audio-based metrics (DER), the duration in terms of time is extracted from the speaker and time labeled transcript for both hypothesis and reference transcript.

5.3.2 Alignment Efficacy Experiment Setup

To evaluate the effectiveness of our MSA algorithm with other dynamic programming-based alignment algorithms, two baseline approaches are added to the comparison. The first baseline is the original Needleman-Wunsch algorithm. For this approach the algorithm works as pairwise alignment (only two sequences) on the character-level.

[3]https://aws.amazon.com/cn/transcribe/
[4]https://www.rev.ai/

The Microsoft Genalog [15] is used as the actual implementation for this approach. The second approach is pairwise alignment on the token-level, which is accomplished with **align4d** package described in 5.3.3 by merging reference transcript into one sequence according to the token-level timestamp. All alignment approaches are evaluated with the accuracy of alignment, which is the portion of words in the reference transcript that have correct mapping to hypothesis transcript. All the ideal mapping are manually generated.

5.3.3 Alignment Package: align4d

As introduced in Section 4.2, our proposed MSA algorithm requires heavy computation for accessing and manipulating large matrices with high dimensions. In order to maximize the computational efficiency of the alignment process, our **align4d** is written in C++20, utilizing only the standard template library, which offers top-tier performance in memory access. On the other hand, considering the usability of such a program, anticipated to have a wide range of uses in evaluating other conversational perspectives, the **align4d** also includes an adaptation layer, enabling easy compilation as a CPython extension and installation as a Python package. This implementation allows for the adjustment of the Levenshtein Distance criterion to determine a full match, partial match, and mismatch, as well as the maximum length of each segment between anchors and the length of the anchor itself, as introduced in Section 4.2. For simplicity and to prevent number overflow in actual computation, in this experiment, the heuristics for scoring function and the Levenshtein Distance criterion d are set as

follows:

$$match(x, y) \leftarrow \begin{cases} S_{fm} = 2 & \text{if } LD(x, y) = 0 \text{ (fully match)} \\ S_{pm} = 1 & \text{if } LD(x, y) \leq 2 \text{ (partial match)} \\ S_{mm} = -1 & \text{if } LD(x, y) > 2 \text{ (mismatch)} \\ S_{gap} = -1 & \text{if only one word is given (gap)} \end{cases}$$

The length of anchor and maximum length of each segment are set to 6 words and 120 words.

The **align4d** is publicly available at `https://github.com/emorynlp/align4d`.

Chapter 6

Results and Analysis

6.1 Model Performance

6.1.1 Conversational Length-based Analysis

Both of our SPM and MPM models are compared with 7 recent audio-based SD systems, including both modular and end-to-end approaches. The results are separated into three groups according to the length of the conversation, with the first two groups containing only a subset of the dataset that meets the length limitation: below and including 15 minutes, above 15 minutes, and no limit.

From Table 6.1, our MPM model's performance is superior than any other SD systems in the experiment for short conversations and comparable to the best audio-based system for the complete dataset. Based on this general comparison, we also compare the best of text-based approach (MPM) with the best of audio-based approach (TOLD) for a more fine-grained analysis on conversational length.

From Figure 6.1, the audio-based system's error rate decreases monotonically as the conversational length increasing, which is potentially because of too less audio information for the system to identify same group of speakers. That of text-based model, on the contrary, generally increases as the conversational length increasing,

Model	≤ 15 Min.		> 15 Min.		Overall	
	WDER	WDER-S	WDER	WDER-S	WDER	WDER-S
pyannote	0.269	0.233	0.137	0.127	0.225	0.187
x-vector+SC	0.378	0.339	0.150	0.175	0.302	0.184
x-vector+AHC	0.298	0.269	0.241	0.268	0.279	0.258
ECAPA+SC	0.402	0.371	0.199	0.152	0.334	0.278
ECAPA+AHC	0.291	0.256	0.166	0.267	0.249	0.239
NeMo-TitaNet	0.233	0.177	0.103	0.088	0.189	0.127
NeMo-MSDD	0.230	0.175	0.085	0.078	0.181	0.123
TOLD	0.206	0.129	**0.080**	**0.069**	0.164	**0.099**
T5-3B SPM	0.312	0.334	0.528	0.563	0.384	0.440
T5-3B MPM	**0.049**	**0.055**	0.114	0.129	**0.101**	0.104

Table 6.1: Performance comparison in terms of WDER and WDER-S with audio-based SD systems (the lower the better) with respect to time.

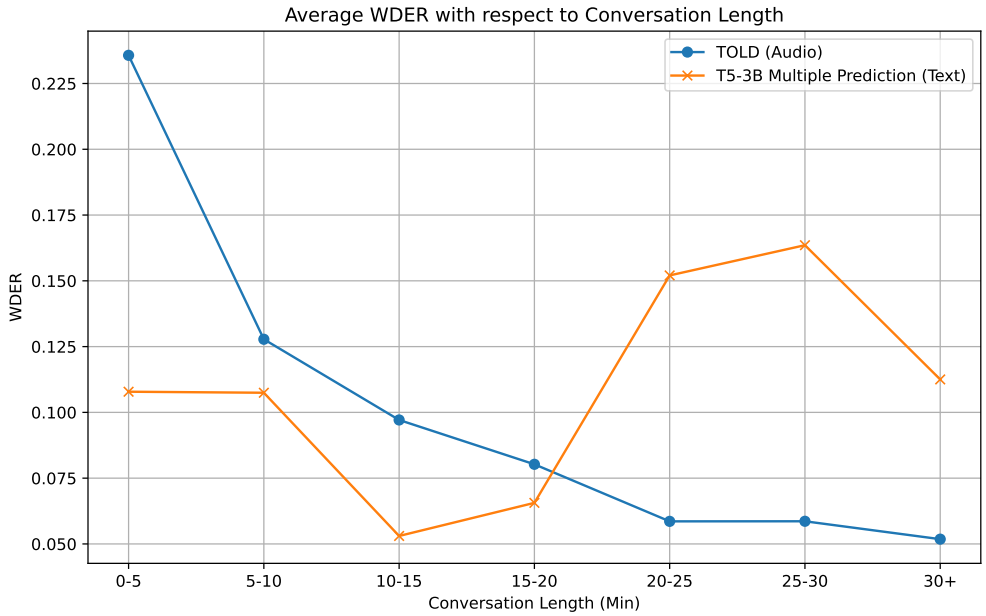

Figure 6.1: Average WDER with respect to conversation length for best audio and text-based SD systems with 5 minute as the interval.

suggesting that while MPM's performance is largely improved over SPM according to Table 6.1, the long-term performance still suffers from speaker label flipping caused by incorrect change predictions, like the example in Section 3.3.

6.1.2 Input Length-based Analysis

Input Sentence	T5-3B SPM		T5-3B MPM	
	WDER	WDER-S	WDER	WDER-S
4	0.428	0.475	0.073	0.277
6	0.388	0.429	**0.056**	0.165
8	0.384	0.440	0.101	**0.104**

Table 6.2: Performance comparison in WDER and WDER-S of SPM and MPM with respect to number of maximum input sentences for each sliding window.

In order to assess the influence of amount of information on the performance, as well as the ability of the model to utilize the information, the text-based models are tested with different length of input for each sliding window. The number of sentences are set to even for avoiding tie situation in majority voting. Also, to fully utilize the pre-training of T5, the total input length should be shorter than the maximum length of T5 pre-training data (512 tokens, which is about 10 sentences). Therefore, we measure the performance on {4, 6, 8} input sentences.

As Table 6.2 shows, the performance of SPM increases marginally and remains unusable. However, for MPM, though the overall performance shown by WDER does not have a general trend of change, the weighted average WDER-S drops consistently, indicating significant performance improvement as the length of input increases.

6.1.3 Text-based Error Types

In order to further determine the types of input where the text-based model makes mistakes, 50 single inputs with at least 1 incorrect prediction and unable to be

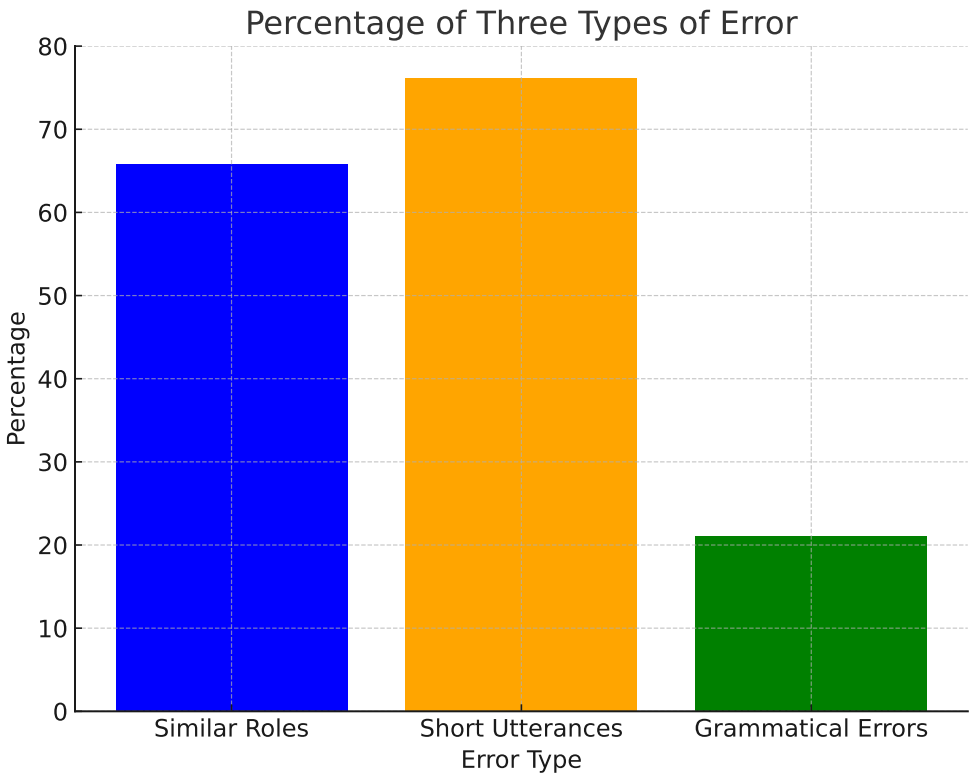

Figure 6.2: Percentage of three error types in the 50 randomly selected input samples.

recovered after aggregation were randomly selected and manually inspected. Three major types of error-prone input are concluded from this inspection:

1. Different speakers contained in the segment of input have similar roles conversationally or socially.

   - **Dialogue**:

     s_1: They just said it was gonna be recorded whatever.
     s_2: So how's it going?
     s_3: Everything's going cool.
     s_4: When I first got here, things were kind of messed up, but I got your email.

   - **Model Prediction**: [A, **A**, **B**, **B**]
   - **Correct Label**: [A, **B**, **A**, **A**]

   The distinction between s_1, s_2 and s_3, s_4 goes unnoticed by the model due to the speakers being teenage students with shared experiences, resulting in similar patterns of speech.

2. Each speaker's utterance within the segment of input is too short to extract enough useful information.

   - **Dialogue**:

     s_1: Wow.
     s_2: What time is it there?
     s_3: What time is it?
     s_4: It's 3:40.

   - **Model Prediction**: [A, B, **B**, **A**]
   - **Correct Label**: [A, B, **A**, **A**]

   The model is unable to distinguish between s_2 and s_3 due to the brevity of the sentences, which lack sufficient logical or linguistic cues to differentiate two speakers in the dialogue.

3. Input sentences contains grammatical errors.

   - **Dialogue**:

     s_1: How things with you busy?
     s_2: I guess I sent you an email, but I suppose you haven't gotten it.

   - **Model Prediction**: [A, **B**]

   - **Correct Label**: [A, **A**]

   The transition from s_1 to s_2 is not detected by the model due to grammatical inaccuracies present in s_1. This grammatical error in s_1 is likely because the error from ASR system.

Figure 6.2 shows the percentage of each type of error in the 50 selected inputs. Note that one input may contain multiple incorrect predictions and more than once types of error can happen on one incorrect prediction. Also, the criterion for determining these types or errors, especially the similar roles, can be subjective and hard to be counted in a wide range of examples.

6.2 Text-based Metrics

This section introduces the result and analysis for the behavior of text-based metrics in Section 6.2.1 and the efficacy and performance of MSA algorithm in Section 6.2.2. The main content of this section has been published at [14]. The experiments were conducted in collaboration with Chen Gong.

6.2.1 Metrics Behavior Analysis

As described in Section 5.3.1, the behavior of Amazon Transcribe and Rev AI's transcription with diarization is first analyzed.

Based on Table 6.3, Amazon Transcribe primarily ignores words, while Rev AI has a more even distribution of error types. The manual inspection confirms with Table

Transcriber	Deletion	Insertion	Substitution	Overlap	$\sum$
Amazon Transcribe	0.068	**0.015**	**0.028**	**0**	**0.111**
Rev AI	**0.051**	0.027	0.031	0.005	0.114

Table 6.3: Average percentages of the four types of errors over all tokens.

6.3 that Amazon Transcribe tends to directly drop the words when the audio segments suffering from low quality or low loudness. Rev AI, on the other hand, tries to recover as much words as possible under the same situation. This behavior, though results in higher error rate other than word deletion, often provides more useful transcript afterwards, and should be shown in performance metrics.

The results of traditional audio and test based metrics as well as the newly proposed TDER and DF1 are shown in Table 6.4.

Transcriber	DER	WDER	WER	TDER	DF1	P	R
Amazon Transcribe	**0.24**	**0.15**	0.34	0.53	0.79	0.87	0.73
Rev AI	0.26	0.20	**0.29**	**0.50**	**0.84**	**0.88**	**0.81**

Table 6.4: Comparing the traditional metrics (**DER**, **WDER**, **WER**) with our new evaluation metrics (**TDER**, **DF1**). For **DF1** the precision (**P**) and recall (**R**) score are also shown. For **TDER**, lower score means higher performance; For **DF1**, higher score means higher performance.

From Table 6.4, it is clear that while DER and WDER are more favor of Amazon Transcribe, WER and our text-based metrics TDER and DF1 are all lean towards Rev AI. Based on the traditional metrics (DER, WDER, WER), we can know that Amazon Transcribe does a better job in SD as it has lower DER and WDER score than Rev AI, but not as good as Rev AI in ASR part due to higher WER score. Through Table 6.3 and manual inspection, we reveal that this is likely because of Amazon Transcribe's tendency in directly dropping words, which, though decreasing the ASR performance scoring, does not affect the audio-based SD-related scoring as the sentence-level timestamp remains. This type of behavior that causes information lost should also be considered and shown in comprehensive evaluation metrics. Both text-

based metrics, especially the DF1 metric, correctly considers the ASR-related errors and give lower scores to Rev AI. For the recall portion of DF1, with same denominator ($len(T_{ref})$), Rev AI's recall is much higher than that of Amazon Transcribe because of higher number of aligned words with correct speaker label, which can only be caused by having more words in hypothesis transcript in general, given that Rev AI's pure audio-based SD performance is weaker. This further proves that the DF1 correctly shows that Rev AI is a better joint ASR and SD system by considering the amount of total useful information provided.

6.2.2 Alignment Algorithm Analysis

By comparing the two baseline approaches (pairwise + character-level, pairwise + word-level) described in 5.3.2, we calculate the alignment accuracy as shown in Table 6.5.

Algorithm	**Accuracy**
Character-level (original NW)	0.92
Token-level w/o Multi-Seq. Support	0.93
Token-level with Multi-Seq. Support (MSA)	**0.99**

Table 6.5: Performance comparison on three types of alignment algorithms on Amazon Transcribe and Rev AI transcripts on CallHome dataset using alignment accuracy.

Based on the accuracy reported in Table 6.5, it is clear that our MSA achieves large improvements over the pairwise alignment algorithms for both character-level (original Needleman-Wunsch algorithm) and token-level (with Levenshtein Distance for fuzzy matching). The improvement brought by token-level matching is marginal (0.92 to 0.93), showing that separating the reference transcript for addressing overlapping issues is the main part that contributes to the overall improvement. This is likely because that the ASR systems generally provide a good quality transcript, especially in terms of spelling errors. Besides, the fundamental mechanism for Needleman-Wunsch algorithm is sufficient for recovering some alignment even if the spelling error occurs.

Chapter 7

Conclusion

This work marks a significant step forward in the exploration of text-based Speaker Diarization (SD), demonstrating the potential of leveraging dialogue transcripts for identifying "who speaks what." Through the deployment of the T5-3B model within Single Prediction Model (SPM) and Multiple Prediction Model (MPM) frameworks, this research has shown that text-based approaches can effectively perform SD, challenging the dependence on audio features. The development and validation of new evaluation metrics, namely Text-based Diarization Error Rate (TDER) and Diarization F1 (DF1), further demonstrate the thesis's contributions to advancing SD methodologies in evaluation. The findings indicate that text-based SD, particularly using the models developed in this study, performs comparably to traditional audio-based methods, especially in short conversations involving two speakers. This not only suggests the viability of text-based diarization as an alternative but also opens up new avenues for research and application in the broader field of conversational AI.

This thesis acknowledges several limitations that highlight areas for further inquiry and improvement. The focus on two-speaker interactions represents a controlled starting point, which already makes substantial progress. However, it limits the generalizability of the findings to more complex, multi-speaker dialogues, which is

often the case in meeting minutes. The heavy reliance on advanced language models and substantial computational resources also poses challenges for scalability and broader application, such as mobile computing. Furthermore, the effectiveness of the alignment tool developed here may vary when encountering highly informal or dialectal speech, indicating a need for ongoing refinement. The efficiency of the alignment algorithm, which has exponential time and space complexity, is still not ideal for long transcripts.

Building on this foundation, future research should aim to extend the methodologies to accommodate multi-speaker conversations, which would significantly broaden the applicability of text-based SD. Exploring more efficient language models could help mitigate computational constraints, making these technologies more accessible. Additionally, enhancing the alignment algorithm to more adeptly handle diverse forms of speech will be crucial for improving the robustness and utility of text-based diarization. Integrating insights from linguistics could also enrich the models, offering more nuanced analyses of speaker dynamics. This can be achieved by reintroducing the machine learning-based approach with consideration for the spelling of words.

Bibliography

[1] Zrar Kh. Abdul and Abdulbasit K. Al-Talabani. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158, 2022. doi: 10.1109/ACCESS.2022.3223444.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[3] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. The fifth 'chime' speech separation and recognition challenge: Dataset, task and baselines. In *Proc. Interspeech 2018*, pages 1561–1565, 2018. doi: 10.21437/Interspeech.2018-1768.

[4] Hervé Bredin. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. INTERSPEECH 2023*, 2023.

[5] Alexandra Canavan and George Zipperlen. CALLFRIEND American English-Non-Southern Dialect LDC96S46. Web Download, 1996. URL `https://ca.talkbank.org/access/CallFriend/`.

[6] Alexandra Canavan, David Graff, and George Zipperlen. CALLHOME American English Speech LDC97S42. Web Download, 1997. URL `https://ca.talkbank.org/access/CallHome/eng.html`.

[7] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: a pre-announcement. In *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, MLMI'05, page 28–39, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3540325492. doi: 10.1007/11677482_3. URL `https://doi.org/10.1007/11677482_3`.

[8] Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Changhan Wang, Jeff Wang, and Skyler Wang. Seamlessm4t: Massively multilingual & multimodal machine translation, 2023.

[9] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pages 3830–3834, 2020.

[10] John DuBois, Wallace L. Chafe, Charles Meyer, and Sandra A. Thompson. Santa Barbara Corpus of Spoken American English. Web Download, 2000-2020. URL https://ca.talkbank.org/access/SBCSAE.html.

[11] Jonathan G. Fiscus, Jerome Ajot, Martial Michel, and John S. Garofolo. The Rich Transcription 2006 Spring Meeting Recognition Evaluation. In *Proceedings of International Workshop on Machine Learning and Multimodal Interaction*, pages 309–322, 2006. URL https://link.springer.com/chapter/10.1007/11965152_28.

[12] Nikolaos Flemotomos and Shrikanth Narayanan. Multimodal Clustering with Role Induced Constraints for Speaker Diarization. In *Proc. Interspeech 2022*, pages 5075–5079, 2022. doi: 10.21437/Interspeech.2022-814.

[13] Nikolaos Flemotomos, Panayiotis Georgiou, and Shrikanth Narayanan. Linguistically aided speaker diarization using speaker role information. In *The Speaker and Language Recognition Workshop (Odyssey 2020)*, odyssey_2020. ISCA, November 2020. doi: 10.21437/odyssey.2020-17. URL http://dx.doi.org/10.21437/Odyssey.2020-17.

[14] Chen Gong, Peilin Wu, and Jinho D. Choi. Aligning speakers: Evaluating and visualizing text-based speaker diarization using efficient multiple sequence alignment. In *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 778–783, 2023. doi: 10.1109/ICTAI59109.2023.00119.

[15] Amit Gupte, Alexey Romanov, Sahitya Mantravadi, Dalitso Banda, Jianjie Liu, Raza Khan, Lakshmanan Ramu Meenal, Benjamin Han, and Soundar Srinivasan. Lights, camera, action! a framework to improve nlp accuracy over ocr documents. *Document Intelligence Workshop at KDD 2021*, 2021.

[16] Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Naga-

matsu. End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors, 2020.

[17] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460, oct 2021. ISSN 2329-9290. doi: 10.1109/TASLP.2021.3122291. URL `https://doi.org/10.1109/TASLP.2021.3122291`.

[18] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 1, pages I–I, 2003. doi: 10.1109/ICASSP.2003.1198793.

[19] Fei Jia, Somshubra Majumdar, and Boris Ginsburg. Marblenet: Deep 1d time-channel separable convolutional neural network for voice activity detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6818–6822, 2021. doi: 10.1109/ICASSP39728.2021.9414470.

[20] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural CRF model for sentence alignment in text simplification. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.709. URL `https://aclanthology.org/2020.acl-main.709`.

[21] Dietrich Klakow and Jochen Peters. Testing the correlation of word error

rate and perplexity. *Speech Commun.*, 38(1):19–28, sep 2002. ISSN 0167-6393. doi: 10.1016/S0167-6393(01)00041-3. URL https://doi.org/10.1016/S0167-6393(01)00041-3.

[22] Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context, 2021.

[23] Harold W Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[24] Wuwei Lan, Chao Jiang, and Wei Xu. Neural semi-Markov CRF for monolingual word alignment. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6815–6828, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.531. URL https://aclanthology.org/2021.acl-long.531.

[25] Federico Landini, Mireia Diez, Themos Stafylakis, and Lukáš Burget. Diaper: End-to-end neural diarization with perceiver-based attractors. *arXiv preprint arXiv:2312.04324*, 2023.

[26] Keon Lee, Kyumin Park, and Daeyoung Kim. Dailytalk: Spoken dialogue dataset for conversational text-to-speech. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095751.

[27] Qiujia Li, Florian L. Kreyssig, Chao Zhang, and Philip C. Woodland. Discriminative neural clustering for speaker diarisation. In *2021 IEEE Spoken Language*

Technology Workshop (SLT), pages 574–581, 2021. doi: 10.1109/SLT48900.2021.9383617.

[28] Runxin Li, Tanja Schultz, and Qin Jin. Improving speaker segmentation via speaker identification and text segmentation. In *Proc. Interspeech 2009*, pages 904–907, 2009. doi: 10.21437/Interspeech.2009-272.

[29] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967. URL https://api.semanticscholar.org/CorpusID:6278891.

[30] Zhao Meng, Lili Mou, and Zhi Jin. Hierarchical rnn with static sentence-level attention for text-based speaker change detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2203–2206, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133110. URL https://doi.org/10.1145/3132847.3133110.

[31] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(70)90057-4. URL https://www.sciencedirect.com/science/article/pii/0022283670900574.

[32] Tae Jin Park and Panayiotis G. Georgiou. Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks. In *Interspeech*, 2018. URL https://api.semanticscholar.org/CorpusID:44061856.

[33] Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J. Han, Shinji Watan-

abe, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72:101317, 2022. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2021.101317. URL `https://www.sciencedirect.com/science/article/pii/S0885230821001121`.

[34] Tae Jin Park, Nithin Rao Koluguri, Jagadeesh Balam, and Boris Ginsburg. Multi-scale speaker diarization with dynamic scale weighting, 2022.

[35] Tae Jin Park, Kunal Dhawan, Nithin Koluguri, and Jagadeesh Balam. Enhancing speaker diarization with large language models: A contextual beam search approach, 2023.

[36] Rohit Paturi, Sundararajan Srinivasan, and Xiang Li. Lexical Speaker Error Correction: Leveraging Language Models for Speaker Diarization Error Correction. In *Proc. INTERSPEECH 2023*, pages 3567–3571, 2023. doi: 10.21437/Interspeech.2023-1982.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[38] Alexis Plaquet and Hervé Bredin. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*, 2023.

[39] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of

transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

[41] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. *Digit. Signal Process.*, 10: 19–41, 2000. URL `https://api.semanticscholar.org/CorpusID:9760419`.

[42] Neville Ryant, Kenneth Ward Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Y. Liberman. The second dihard diarization challenge: Dataset, task, and baselines. In *Interspeech*, 2019. URL `https://api.semanticscholar.org/CorpusID:195069497`.

[43] Laurent El Shafey, Hagen Soltau, and Izhak Shafran. Joint speech recognition and speaker diarization via sequence transduction, 2019.

[44] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(81)90087-5. URL `https://www.sciencedirect.com/science/article/pii/0022283681900875`.

[45] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018. doi: 10.1109/ICASSP.2018.8461375.

[46] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, 2014. doi: 10.1109/ICASSP.2014.6854363.

[47] Jiaming Wang, Zhihao Du, and Shiliang Zhang. Told: A novel two-stage overlap-aware framework for speaker diarization, 2023.

[48] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopez Moreno. Speaker diarization with lstm, 2022.

[49] Quan Wang, Yiling Huang, Guanlong Zhao, Evan Clark, Wei Xia, and Hank Liao. Diarizationlm: Speaker diarization post-processing with large language models, 2024.

[50] Jee weon Jung, Soonshin Seo, Hee-Soo Heo, Geonmin Kim, You Jin Kim, Young ki Kwon, Minjae Lee, and Bong-Jin Lee. Encoder-decoder Multimodal Speaker Change Detection. In *Proc. INTERSPEECH 2023*, pages 5311–5315, 2023. doi: 10.21437/Interspeech.2023-2289.

[51] Juan Pablo Zuluaga, Seyyed Saeed Sarfjoo, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlícek, Karel Ondrej, Oliver Ohneiser, and Hartmut Helmke. Bertraffic: Bert-based joint speaker role and speaker change detection for air traffic control communications. *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 633–640, 2021. URL `https://api.semanticscholar.org/CorpusID:247839114`.