Distribution Agreement

In presenting this thesis or dissertation 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 or dissertation 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 or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Chen Gong March 12 2023

Evaluating Speaker Diarization in Transcripts: A Text-based Approach with the TDER Metric and the TranscribeView System

By

Chen Gong

Jinho D. Choi
Advisor

Computer Science

Jinho D. Choi, Ph.D.
Advisor

Emily Wall, Ph.D.
Committee Member

Roberto Franzosi, Ph.D.
Committee Member

2023

Evaluating Speaker Diarization in Transcripts: A Text-based Approach with the TDER Metric and the TranscribeView System

By

Chen Gong

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

2023

Abstract

Evaluating Speaker Diarization in Transcripts: A Text-based Approach with the TDER Metric and the TranscribeView System
By Chen Gong

Speaker Diarization (SD), the task of attributing speaker labels to dialogue segments, has traditionally been performed and evaluated at the audio level. The diarization error rate (DER) metric for SD systems measures errors in time but does not account for the impact of automatic speech recognition (ASR) systems on transcript-based performance. Word error rate (WER), the evaluation metric for ASR, only considers errors in word insertion, deletion, and substitution, disregarding SD quality. To better evaluate SD performance at the text level, this paper proposes Text-based Diarization Error Rate (TDER) and diarization F1-score, which jointly assess SD and ASR performance.

To address inconsistencies in token counts between hypothesis and reference transcripts, we introduce a multiple sequence alignment tool that accurately maps words between reference and hypothesis transcripts. Our alignment method achieves 99% accuracy on a simulated corpus generated based on common SD and ASR errors.

Comparisons with DER, WER, and WDER on 10 transcripts from the CallHome dataset demonstrate that TDER and diarization F1-score provide a more reliable evaluation of speaker diarization at the text level. To enable a comprehensive evaluation of transcript quality, we present TranscribeView, a web-based platform for assessing and visualizing errors in speech recognition and speaker diarization. To the best of our knowledge, TranscribeView is the first comprehensive platform that enables researchers to align multi-sequence transcripts and assess and visualize speaker diarization errors, contributing significantly to the advancement of data-driven conversational AI research.

Evaluating Speaker Diarization in Transcripts: A Text-based Approach with the TDER Metric and the TranscribeView System

By

Chen Gong

Jinho D. Choi, Ph.D.
Advisor

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

Computer Science

2023

Acknowledgments

I would like to express my gratitude to my advisor, Dr. Jinho D. Choi, for guiding me through my thesis and providing me with the opportunity to explore various research projects. Joining Emory Natural Language Processing Lab under Dr. Choi's mentorship, I discovered my passion for NLP research and learned how to be a better researcher.

I am also thankful to Dr. Emily Wall and Dr. Roberto Franzosi for their contributions to my academic journey. I took two classes with Dr. Wall and completed a computer vision research project under her guidance. Through these experiences, I learned valuable design and visualization techniques that were helpful in my thesis research. Dr. Franzosi gave me the opportunity to work on the NLP-suite project, where I developed and practiced software development skills. These skills turned out to be essential in the implementation of the TranscribeView system.

I would also like to thank Peilin Wu for designing and implementing the Multiple Sequence Alignment tool, which enabled all the evaluation metrics and the visualization system TranscribeView.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Background and motivation

In recent years, data-driven dialogue systems such as BlenderBot [17] and ChatGPT[1], which utilize large seq-to-seq language models [3, 9, 15], have garnered significant interest from various communities. The applications of these dialogue systems are seemingly endless, with numerous organizations processing years' worth of audio recordings from human-to-human dialogues to train their models. However, these audio recordings were often collected without the intention of data-driven model development, resulting in low-quality audio with considerable background noise that makes automatic speech recognition (ASR) challenging. Moreover, these recordings typically use a single channel for all speakers rather than assigning dedicated channels to individual speakers, necessitating the use of speaker diarization (SD), an additional challenging task.

SD is a speech processing task that identifies the speakers of audio segments extracted from a conversation involving two or more speakers [13]. Despite the excellent performance of ASR models in translating audio into text without recognizing individual speakers [2, 5, 14], unstable SD can have a detrimental effect on developing

[1] `https://openai.com/blog/chatgpt`

robust dialogue models, as any model trained on such data would fail to learn unique languages for distinct speakers. Therefore, analyzing the performance of ASR and SD on specific audio streams is crucial for producing high-quality transcripts. However, there has been a lack of comprehensive approaches to simultaneously evaluate both types of errors.

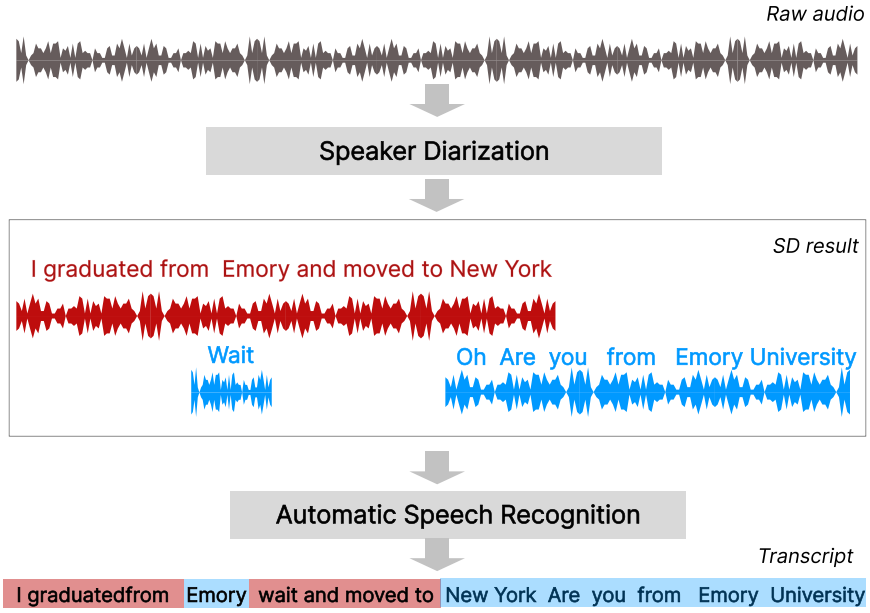


Figure 1.1: Process of generating transcripts

1.2 Thesis Objectives and Contributions

This thesis presents a text-based approach for evaluating speaker diarization quality in transcripts, focusing on the development of the Text Diarization Error Rate (TDER) metric, the Multiple Sequence Alignemnt Algorithm and the TranscribeView system. The main contributions of this work include:

1. The introduction and validation of the TDER metric for text-based speaker diarization evaluation, comparing it with existing metrics such as DER, WDER, and WER.

2. Align4d[2]: The design and implementation of an efficient multi-sequence alignment algorithm based on the Needleman-Wunsch algorithm for token-to-token mapping between generated and reference transcripts.

3. The development of TranscribeView[3], a comprehensive evaluation platform for transcript evaluation and diarization error visualization.

1.3 Thesis organization

The rest of the thesis is organized as follows: Chapter 2 provides a literature review on speaker diarization evaluation metrics, transcript alignment methods, and visualization techniques for transcript evaluation. Chapter 3 introduces the TDER metric and its comparison with other metrics. Chapter 4 presents the multiple sequence alignment algorithm and its implementation. Chapter 5 presents the experiment on the correctness of the multiple sequence alignment algorithm and the metrics comparison. Finally, Chapter 6 describes the design and implementation of the TranscribeView system, including its features and user interface.

[2] https://github.com/emorynlp/align4d
[3] https://github.com/emorynlp/TranscribeView

Chapter 2

Background

In this chapter, we introduce the background information necessary for understanding the context and motivation behind this thesis. This chapter provides an overview of various evaluation metrics and alignment algorithms relevant to the field of Automatic Speech Recognition (ASR) and Speaker Diarization (SD). Section 2.1 discusses different types of ASR evaluation metrics, offering insights into their strengths and limitations. Section 2.2 introduces several SD metrics, such as Diarization Error Rate (DER), Jaccard Error Rate (JER), and Word-level Diarization Error Rate (WDER), highlighting their unique characteristics and applications. Finally, Section 2.3 presents a brief overview of alignment algorithms, which are critical for evaluating and comparing transcripts. The information in this chapter lays the foundation for understanding the development and evaluation of the proposed TDER metric and Multiple Sequence Alignment Algorithm.

2.1 ASR Evaluation Metrics

Automatic Speech Recognition (ASR) is the task of converting spoken language into written text using computational algorithms and models. ASR systems play a crucial role in various applications, such as transcription services, voice assistants, real-time

captioning, and more. These systems rely on a combination of acoustic models, which capture the relationship between speech signals and phonemes, and language models, which predict the likelihood of word sequences. The performance of these systems is crucial for various applications, including transcription services, voice assistants, and real-time captioning. To assess the quality and efficiency of ASR systems, researchers and practitioners rely on several evaluation metrics. However, these metrics may have limitations when it comes to evaluating speaker diarization quality.

One of the most widely used metrics for evaluating ASR systems is Word Error Rate (WER). WER measures the similarity between a reference transcript (ground truth) and a hypothesis transcript (ASR output) by calculating the minimum number of edit operations (i.e., insertions, deletions, and substitutions) required to transform the hypothesis transcript into the reference transcript, divided by the total number of words in the reference transcript. The result is expressed as a percentage, with lower WER values indicating better ASR performance.

$$WER = \frac{Insertions + Deletions + Substitutions}{Total\ Reference\ Words} \times 100 \tag{2.1}$$

In addition to WER, other metrics such as Sentence Error Rate (SER) and Character Error Rate (CER) are also used to evaluate ASR systems. SER focuses on sentence-level errors, while CER considers character-level differences between the reference and hypothesis transcripts. Despite their differences, these metrics share common limitations when it comes to speaker diarization quality assessment.

The primary limitation of WER, SER, and CER in evaluating speaker diarization quality is that they do not take into account speaker information. They solely focus on word, sentence, or character-level errors and fail to capture errors related to speaker identification or segmentation. As a result, these metrics are insufficient for evaluating speaker diarization quality, as they cannot provide insights into the performance of systems in accurately identifying and segmenting speakers within a conversation.

In conclusion, although WER, SER, and CER are commonly used metrics for evaluating ASR performance, their limitations in addressing speaker diarization quality necessitate the development of specialized metrics tailored for assessing speaker diarization tasks.

2.2 Speaker Diarization Evaluation Metrics

Speaker Diarization is the task of identifying and segmenting speakers in an audio recording containing multiple speakers. Evaluating the performance of speaker diarization systems is essential to ensure their effectiveness in various applications, such as meeting transcription, broadcast news segmentation, and speaker-specific indexing. Several evaluation metrics have been developed to assess speaker diarization quality, including Diarization Error Rate (DER), Jaccard Error Rate (JER), and Word Diarization Error Rate (WDER).

2.2.1 Diarization Error Rate (DER)

Diarization Error Rate (DER) is the most common metric for the Speaker Diarization task [11, 1]. It measures the fraction of time that the audio segment is not mapped to the correct speaker. To calculate DER score, a one-to-one mapping between speaker IDs in reference and hypothesis is needed so that we can determine the correctness of each labeled segment [1]. DER is computed as the following equation:

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

where S is the total number of audio segments. $dur(s)$ is the time duration of a single segment s. $N_r(s)$ and $N_h(s)$ represent the number of speaker in segment s from reference and hypothesis outputs. $N_{corr}(s)$ gives the number of correct speakers given by the hypothesis output. The denominator $\sum_{s=1}^{S} dur(s) N_{ref}(s)$ gives the total

scoring time.

Equation 2.2 can be decomposed into four parts that represent different aspects of diarization errors:

- **Speaker error**: when speaker ID is incorrect in a segment

$$E_{Spkr} = \frac{\sum_{s=1}^{S} dur(s) \cdot (min(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{\sum_{s=1}^{S} dur(s) N_{ref}(s)} \tag{2.3}$$

- **False alarm speech**: when assigned speaker ID is labeled to non-speech segments

$$E_{FA} = \frac{\sum_{s=1}^{S} dur(s) \cdot (N_{hyp}(s) - N_{ref}(s))}{\sum_{s=1}^{S} dur(s) N_{ref}(s)} \qquad \forall\, (N_{hyp}(s) - N_{ref}(s)) > 0 \tag{2.4}$$

- **Missed speech**: when speech segments is not identified in hypothesis output

$$E_{MISS} = \frac{\sum_{s=1}^{S} dur(s) \cdot (N_{ref}(s) - N_{hyp}(s))}{\sum_{s=1}^{S} dur(s) N_{ref}(s)} \qquad \forall\, (N_{ref}(s) - N_{hyp}(s)) > 0 \tag{2.5}$$

- **Overlap speaker**: percentage of scored time when multiple speakers appear in a segment and some of them are not labeled in hypothesis output. This error is often included in E_{MISS}.

Therefore equation 2.2 can be written as:

$$DER = E_{Spkr} + E_{FA} + E_{MISS} + E_{Overlap} \tag{2.6}$$

Note that many of the recent work has been ignoring the overlap error in their evaluation and only counting three types of errors.

2.2.2 Word-level Diarization Error Rate (WDER)

Recently, more research has been training the speaker diarization and ASR system jointly, where traditional audio-based metrics are no longer applied [16]. Word-level Diarization Error Rate (WDER) has been proposed to evaluate the SD result on the joint SD and ASR system [13, 12]. WDER is a metric designed to evaluate the quality of speaker diarization at the word level. Traditional diarization metrics, such as Diarization Error Rate (DER), often focus on the time-based errors and do not take into account the content of the conversation. WDER provides a more fine-grained evaluation of speaker diarization performance by considering the alignment of words and speaker labels in the transcripts.

$$WDER = \frac{S_{is} + C_{is}}{S + C} \tag{2.7}$$

Equation 2.7 shows the way to compute WDER, where S is the number of ASR substitutions and C is the number of correct ASR words. S_{is} and C_{is} means the corresponding tokens with incorrect speaker labels. To calculate WDER, a word-level alignment between the reference and hypothesis transcripts is required (detail in chapter 4).

Limitations: It is worth noting that WDER only counts the tokens that are aligned between reference and hypothesis transcripts (substitutions and correct words). Inserted words and deleted words are not considered in the metrics. However, as section 2.2.1 mentioned, diarization errors can be categorized into four parts: speaker error, false alarm speech, missed speech, and overlap speaker. Only speaker errors will be captured in aligned words; other types of diarization errors are reflected in the deleted and inserted tokens in ASR outputs.

2.3 Transcript alignment methods

The token-to-token alignment has always been a difficulty with transcript evaluation. Due to the inconsistency of length and spelling of reference and hypothesis transcripts, a word mapping is required in order to measure each word in the transcript [6].

- **Edit Distance (Levenshtein Distance)**: Edit Distance is often used in ASR evaluation for computing WER score [8]. Edit distance is a measure of the similarity between two strings, defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other. For ASR evaluation, the reference and hypothesis tokens can be treated as strings and aligned using an edit distance algorithm.

- **Needleman-Wunsch Algorithm**: The Needleman-Wunsch algorithm is a global sequence alignment method used in bioinformatics to align protein or nucleotide sequences [10]. It can be adapted for transcript evaluation by treating the reference and hypothesis tokens as sequences and finding the optimal alignment with the highest similarity score.

- **Smith-Waterman Algorithm**: The Smith-Waterman algorithm is a local sequence alignment method that finds the most similar subsequences between two sequences [18]. For transcript evaluation, it can be used to align the reference and hypothesis tokens by identifying the regions with the highest similarity.

Chapter 3

Text-based Diarization Error Rate and F-1 score

Chapter 2 introduces some common metrics used in ASR and speaker diarization evaluation and their limitations in evaluating speaker diarization quality using pure transcripts. To overcome these issues, this chapter introduces our proposed metrics Text-based Diarization Error Rate (TDER) and diarization F-1 score.

3.1 Text-based Diarization Error Rate (TDER)

TDER is an adapted version of the original DER introduced in section 2.2.1. DER measures the fraction of time that the audio segment is not mapped to the correct speaker. In terms of the text-based speaker diarization, the length of the audio can be directly converted to the length of the text sequence in terms of the number of tokens included. Such comparison is only meaningful when comparing within the pair of tokens where a definite one-to-one mapping is established. In this way, the **false alarm** and **miss** is the number of tokens that are aligned to a gap, and the **confusion** is the number of tokens that are aligned to incorrect speakers. Based on the aforementioned adaptation, we present the Text-based Diarization Error Rate

(TDER) that can be described as in equation 3.2:

$$\text{TDER} = \frac{\sum_{u=1}^{U} len(u) \cdot (max(N_{ref}(u), N_{hyp}(u)) - N_{correct}(u))}{\sum_{u=1}^{U} len(u) N_{ref}(u)} \tag{3.1}$$

Where $U = \{u_1, ..., u_i\}$, U is the ground-truth transcript and u_i represent each utterance in the ground-truth transcript. $len(u)$ returns the number of tokens in utterance u. $N_{ref}(u)$ and $N_{hyp}(u)$ represent the number of speaker in utterance u from reference and hypothesis transcripts. $N_{correct}(u)$ is the number of speakers that are correctly matched in two transcripts. Since in the reference transcripts, each utterance is only spoken by one speaker, $N_{ref}(u)$ is always equal to 1. Therefore, we can rewrite TDER as follow:

$$\text{TDER} = \frac{\sum_{u=1}^{U} len(u) \cdot (max(1, N_{hyp}(u)) - N_{correct}(u))}{N} \tag{3.2}$$

N is the total number of tokens in the reference transcripts. This metric captures different aspects of Speaker Diarization errors. When $N_{hyp}(u) = 0$, the numerator part captures the **Missed Speech** errors. When $N_{hyp}(u) > 1$, the hypothesis utterance contains more than one speaker, which contains **Speaker Confusion** and **Overlap** errors. When $N_{hyp}(u) = 1$ and $N_{correct}(u) = 0$, the hypothesis utterance's speaker id is labeled correctly. In this case, the numerator is 0 and therefore is not counted toward the SD errors.

Compared to the previous metrics, TDER is based on text and alignment/gap-aware, which is compatible with situations where the ground-truth text has a different number of tokens than the hypothesis text.

3.2 Diarization F-1 score

In addition to TDER, we also use F1-score to measure the diarization quality on the text level determining the precision and recall as follows:

$$R = \frac{align(T_{ref}, T_{hyp})}{length(T_{ref})} \tag{3.3}$$

$$P = \frac{align(T_{hyp}, T_{ref})}{length(T_{hyp})} \tag{3.4}$$

T_{ref} and T_{hyp} represent the sequence of tokens in the reference transcript and hypothesis transcript (i.e, $T_{ref} = \{t_1, ..., t_n\}$). Each token t_i contains a word and a speaker id. $align(T1, T2)$ aligns sequence T1 onto sequence T2 and returns the number of correctly labeled tokens in T1(will introduce in 3.2). Therefore, R gives the percentage of tokens from ground-truth transcripts that are correctly labeled in the generated transcripts, while P gives the percentage of correctly labeled tokens in the generated transcripts.

Chapter 4

Multiple Sequence Alignment for Transcript Mapping

In this chapter, we discuss the necessity of Multiple Sequence Alignment (MSA) in the context of transcript evaluation and provide an in-depth explanation of our implementation of the MSA algorithm. While the dynamic programming approach for MSA has been previously proposed by Fiscus et al. [4], their work primarily offers a high-level overview of extending the 2-dimensional dynamic programming to higher dimensions. Our contribution in this chapter includes a more comprehensive elucidation of the algorithm and the introduction of a publicly available MSA tool called align4d[1]. The permutation method introduced in Chapter 4.4 and the final tool align4d is implemented by Peilin Wu.

4.1 Limitations for Pair-wise Alignment Algorithms

A hypothesis transcript cannot be evaluated unless its tokens are aligned with the most probable ones in the reference transcript.

In Figure 4.1, the hypothesis (A') has 3 errors against the reference (A, B), which

[1] `https://github.com/emorynlp/align4d`

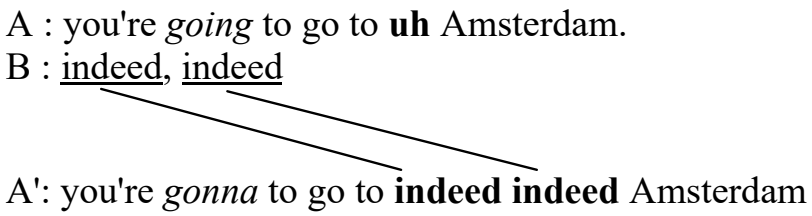


Figure 4.1: Examples of transcript errors, where the reference consists of multiple sequences.

make them difficult to be aligned:

1. A spelling and word recognition error; '*going*' is recognized as '*gonna*' in the hypothesis.
2. A missing word; '*uh*' is not recognized.
3. Overlapped utterances; B's utterance is spoken while A utters '*Amsterdam*', which are merged into one utterance for A'.

The first two types are ASR errors that can be handled by most pairwise alignment methods such as the Needleman-Wunsch (NW) algorithm [10]. However, the third type is an SD error involving multiple sequences which occur when utterances by distinct speakers overlap in time. Figure 4.2 describes how the NW algorithm treats them as the insertion and deletion errors and fails to align those tokens completely:

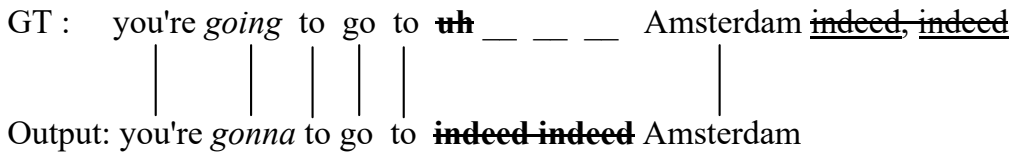


Figure 4.2: The result by the NW algorithm.

To overcome this challenge, a new multi-sequence alignment algorithm is designed by

expanding the dimension of dynamic programming, which takes utterances from all sequences in parallel (Fig. 4.3).

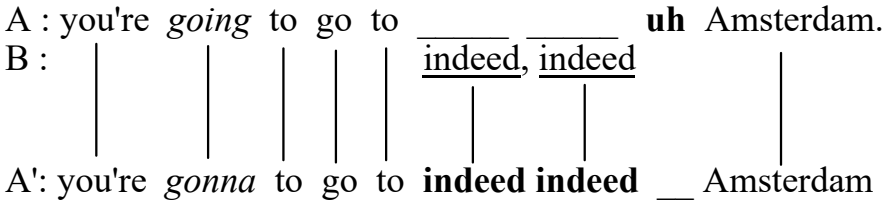


Figure 4.3: The result by our multi-sequence alignment algorithm for the above example.

4.2 Needleman-Wunsch algorithm

We choose the Needleman-Wunsch algorithm as our baseline method. This algorithm is designed to align protein or nucleotide sequences [10]. It finds the best global alignment over the entire input sequences. It allows a gap between tokens when a mismatch happens, and by changing the match metric, we have certain tolerance for misspellings. Similar to the dynamic programming solution to Longest Common Subsequence solutions, this algorithm consists of two parts: **computing a scoring table**, and **backtrack to recover to alignment**. Figure 4.4 gives an example of the scoring table to align two sequences. In this example, the vertical sequence is the reference and the horizontal sequence is the hypothesis.

Compute Score Table takes two sequences $X = \{x_1...x_m\}$ and $Y = \{y_1...y_n\}$ as input. In our case, X, Y are two transcripts, and each x_i, y_i represents a token in the transcript. d is a penalty score indicating that two tokens are not matched. To find the global optimal alignment, a two-dimensional array (or matrix) F is allocated. There is one row for each character in sequence X, and one column for each character in sequence Y. The first two for-loops from line 5 to 10 initialize the first row and

		I	are	fish	am	you	oh	yea
	0	-1	-2	-3	-4	-5	-6	-7
I	-1	2	1	0	-1	-2	-3	-4
am	-2	1	4	3	2	1	0	-1
fish	-3	0	3	6	5	4	3	2
are	-4	-1	2	5	8	7	6	5
you	-5	-2	1	4	7	10	9	8
too	-6	-3	0	3	6	9	12	11
Oh	-7	-4	-1	2	5	8	11	11
yes	-8	-5	-2	1	4	7	10	13

Figure 4.4: Example of score matrix of two sequences of tokens and backtracking following the blue arrow.

Algorithm 1 Needleman-Wunsch Compute Score Table

Input: two sequences X, Y
Require: $n \geq 0$
$m \leftarrow X.length$
$n \leftarrow Y.length$
$d \leftarrow Gap\ penalty\ score$
$F \leftarrow [0..m, 0..n]$
for $i = 1 \rightarrow m$ **do**
    $F[i, 0] \leftarrow d * i$
end for
for $j = 1 \rightarrow n$ **do**
    $F[0, j] \leftarrow d * j$
end for
for $i = 1 \rightarrow m$ **do**
    for $j = 1 \rightarrow n$ **do**
        $M \leftarrow F[i-1, j-1] + Score(X_i, Y_j)$
        $D \leftarrow F[i-1, j] + d$
        $I \leftarrow F[i, j-1] + d$
        $F[i, j] \leftarrow \max(M, I, D)$
    end for
end for

the first column with d multiplied by the index. The third nested for-loop in line 11 updates every cell in the table following the rule:

$$F_{i,j} = max \begin{cases} F_{i-1,j-1} + Score(X_i, Y_j) \\ F_{i-1,j} + d \\ F_{i,j-1} + d \end{cases} \tag{4.1}$$

$Score(X_i, Y_j)$ is a scoring system that evaluates the similarity between two tokens. For word-level comparison, we choose Levenshtein Distance as our $Score$ function to measure the similarity between two words.

Algorithm 2 Backtrack Needleman-Wunsch Alignment

```
1: i ← m
2: j ← n
3: aligned_X ← empty list
4: aligned_Y ← empty list
5: while i > 0 or j > 0 do
6:     if i > 0 and j > 0 and F[i, j] = F[i − 1, j − 1] + Score(X_i, Y_j) then
7:         insert X_i at the beginning of aligned_X
8:         insert Y_j at the beginning of aligned_Y
9:         i ← i − 1
10:        j ← j − 1
11:    else if i > 0 and F[i, j] = F[i − 1, j] + d then
12:        insert X_i at the beginning of aligned_X
13:        insert gap at the beginning of aligned_Y
14:        i ← i − 1
15:    else
16:        insert gap at the beginning of aligned_X
17:        insert Y_j at the beginning of aligned_Y
18:        j ← j − 1
19:    end if
20: end while
```

The backtracking part of the Needleman-Wunsch algorithm aims to recover the optimal alignment between two sequences using the score table computed in the previous step. Starting from the bottom-right corner of the scoring matrix, it iteratively traces back the optimal path until it reaches the top-left corner. The backtracking

procedure follows the scoring rules for matches, mismatches, and gaps to reconstruct the aligned sequences of both input sequences X and Y. This process results in the final global alignment, which is a pair of sequences with the same length containing matched characters, mismatches, and gaps.

4.3 Adaptation to 3-dimension

The Needleman-Wunsch (NW) algorithm is capable of addressing the first three types of errors listed in section 4.1, which include misspellings, missing words, and extra words. To tackle errors arising from overlapping utterances, we initially attempt to extend the NW algorithm to accommodate three dimensions. In the context of a two-speaker conversation transcript, we separate the reference transcript into two sequences based on speaker ID, and subsequently align these two sequences with the hypothesis transcript concurrently.

Algorithm 3 demonstrates the computation of the 3D scoring matrix. The algorithm takes three input sequences $X = x_1, ..., x_n$, $Y = y_1, ..., y_n$, and $Z = z_1, ..., z_n$, where X represents the transcriber's output transcript, Y consists of speaker A's utterances from the ground-truth transcript, and Z contains speaker B's utterances from the ground-truth transcript. Each element x_i, y_i, z_i represents a token. The algorithm aligns tokens from sequences Y and Z with those in sequence X. In this manner, as depicted in Figure 4, we can accurately align overlapping utterances by aligning each speaker's utterances separately.

Instead of generating a 2D table, we need to compute a 3D matrix for three sequences. Line 3 allocates a matrix $F_{m \times n \times d}$ according to the lengths of the input sequences. Lines 4-15 initialize the three surfaces xy, xz, and yz using the scoring table from the Needleman-Wunsch algorithm. Subsequently, the nested for loop in lines 16-22 updates each cell in matrix F following the rules specified in Equation 4.

Algorithm 3 Compute 3D Scoring Matrix

Input: three sequences X, Y, Z
$m, n, d \leftarrow X.length, Y.length, Z.length$
$g \leftarrow Gap\ penalty\ score$
$F \leftarrow [0..m, 0..n, 0..d]$
$table_{xy} \leftarrow$ Pairwise-Align-table(X, Y)
$table_{xz} \leftarrow$ Pairwise-Align-table(X, Z)
$table_{yz} \leftarrow$ Pairwise-Align-table(Y, Z)
for $i = 1 \rightarrow m$ **do**
    for $j = 1 \rightarrow n$ **do**
        for $k = 1 \rightarrow d$ **do**
            $F_{i,j,0} \leftarrow table_{xy}[i, j]$
            $F_{i,0,k} \leftarrow table_{xz}[i, k]$
            $F_{0,j,k} \leftarrow table_{yz}[j, k]$
        end for
    end for
end for
for $i = 1 \rightarrow m$ **do**
    for $j = 1 \rightarrow n$ **do**
        for $k = i \rightarrow d$ **do**
            $updateF_{i,j,k}$
        end for
    end for
end for

$$F_{i,j,k} = max \begin{cases} F_{i-1,j,k} + Score(X_i, -, -) \\ F_{i,j-1,k} + Score(-, Y_j, -) \\ F_{i,j,k-1} + Score(-, -, Z_k) \\ F_{i-1,j-1,k} + Score(X_i, Y_j, -) \\ F_{i-1,j,k-1} + Score(X_i, -, Z_k) \\ F_{i,j-1,k-1} + Score(-, Y_j, Z_k) \\ F_{i-1,j-1,k-1} + Score(X_i, Y_j, Z_k) \end{cases} \tag{4.2}$$

4.4 Multiple Sequence Alignment

In this section, I would like to acknowledge the valuable contributions of Peilin Wu, whose work on the development of the multiple sequence alignment algorithm with permutations has been instrumental in the success of this research.

Section 4.3 introduces an example of expanding the pairwise alignment to align three sequences. However, real-world applications often involve scenarios with an arbitrary number of sequences to be aligned. To accommodate this flexibility, we need a method that can efficiently generate all possible combinations of sequence positions and indices, regardless of the total number of input sequences.

Permutations play a crucial role in this context because they help us explore all potential combinations of tokens from the hypothesis and reference sequences. By systematically generating and evaluating these permutations, we can effectively align multiple sequences without the need to explicitly define the number of sequences or the number of loops required for the alignment process.

Algorithm 4 shows how we used permutations when computing the multi-dimensional scoring table. Algorithm 4 is given a list of sequences S, where S_0 is the hypothesis sequence, and the remaining sequences are reference sequences separated by speaker.

Algorithm 4 Multiple Sequence Alignment with Permutations

Input: A list of sequences S
$n \leftarrow length(S)$
Initialize an empty scoring matrix F with dimensions $(len(S_0) + 1) \times (len(S_1) + 1) \times \cdots \times (len(S_n) + 1)$
$seqCombinations \leftarrow generate\_combinations(n)$
for each $seqComb$ in $seqCombinations$ **do**
    $indexPermutations \leftarrow generate\_index\_permutations(seqComb, S)$
    for each $perm$ in $indexPermutations$ **do**
        Calculate the score $score \leftarrow scoringFunction(perm, S, F)$
        Update the scoring matrix F using $perm$ and $score$
    end for
end for

The length of the list is denoted as n. We first initialize an empty scoring matrix F with dimensions based on the lengths of all the sequences in the list.

The `seqCombinations` variable is calculated using the `generate_combinations` function to store all possible sequence position combinations. For each combination `seqComb`, we generate all possible permutations of indices using the `generate_index_permutations` function with the input parameters `seqComb` and the list of sequences S.

For each permutation `perm` in the generated `indexPermutations`, we calculate the score using a scoring function that takes the permutation, the list of sequences, and the scoring matrix as inputs. Finally, we update the scoring matrix F using the calculated score and the current permutation. This process is repeated for all combinations and permutations, resulting in the final scoring matrix for the multiple sequence alignment with permutations.

Chapter 5

Experiments and Results

Our experiment mainly contains two parts: multi-sequence alignment and Text-based metric evaluation. Two public available transcribers, Amazon and Rev AI, are selected for audio transcript generation. Chapter 5.2 gives the experiment of our alignment algorithm, including the experiment setup and data preparation. Chapter 5.1 shows the experiment and results of comparison between our purposed metrics and other popular metrics.

5.1 Transcribers

For our experiments, we selected two widely-used transcription services: Amazon Transcribe and RevAI.

Amazon Transcribe is an automatic speech recognition (ASR) service developed by Amazon Web Services (AWS). It offers options to include speaker diarization information in the result.

RevAI, developed by Rev.com, is another ASR service that offers transcription capabilities. It employs state-of-the-art artificial intelligence techniques to provide accurate transcriptions across different industries and use cases. RevAI also provides speaker diarization in the output transcripts.

Both Amazon Transcribe and RevAI are publicly available and offer a free tier of usage, making them ideal choices for our experiments due to their accessibility, cost-effectiveness and their ability to perform speaker diarization tasks.

5.2 CallHome Corpus

We use the CABank English CallHome Corpus for this experiment. This corpus is a collection of telephone conversations in English, designed for research purposes. It is part of the larger CallHome project, which includes telephone conversations in various languages. The English CallHome Corpus contains 120 unscripted, informal telephone conversations between native English speakers. Each conversation lasts approximately 30 minutes and covers a range of topics, as participants were free to discuss anything they wished. Most conversations contains 10 minutes manually transcribed text.

The CallHome transcript format is based on the CHAT (Codes for the Human Analysis of Transcripts) format, which is a widely-used standard for transcribing spoken language in research. The format is designed to represent various aspects of spoken language, such as speaker turns, pauses, overlapping speech, and non-verbal cues. This is the reference transcript we used in later experiments. Here is an example of the CallHome transcript:

```
@UTF8
@PID: 11312/t-00001007-1
@Begin
@Languages: eng
@Participants: A Subject, B Subject
@ID: eng|eng|A|||||Subject|||
@ID: eng|eng|B|||||Subject|||
```

```
@Media: 4074, audio
*A: So, anyway, how are you doing these days? 145150_147510
*B: Things are going very well.
*B: I think I had mentioned before that, um, that uh, that uh, that
there's a company now that I'm working with . 147700_154910
*B: um, uh, which is very much just, just myself and Guss. 155500_158710
@End
```

5.3 Evaluation of Multiple Sequence Alignment

5.3.1 Simulated Data

To evaluate the accuracy of our alignment algorithm, it is essential to establish a ground-truth token mapping between reference and hypothesis transcripts. However, manually creating this mapping is labor-intensive. As a result, we opt to simulate hypothesis transcripts based on reference transcripts, allowing us to generate token mappings more efficiently and effectively.

Error Distribution

As discussed in section 4.1, we classify auto-transcript errors into four types: Substitutions, Missing tokens, Extra tokens, and Overlapped utterances. To better simulate the hypothesis transcript, we manually labeled the errors from four hypothesis transcripts generated by Amazon and RevAI.

Transcriber	Missing	Extra	Substitution	Overlapping
Amazon	6.8	1.5	2.8	0
RevAI	5.1	2.7	3.1	0.5

Table 5.1: Average percentage of four types of error over all tokens found in RevAI and Amazon.

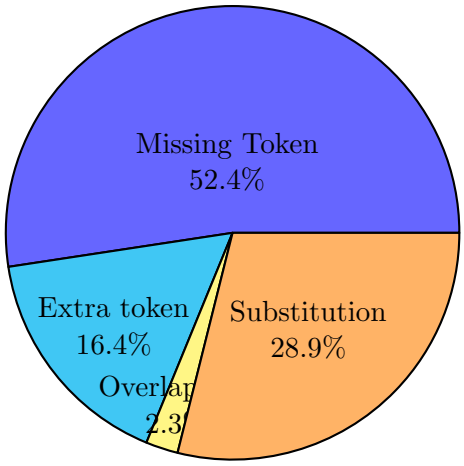


Figure 5.1: Average Error distribution for Rev AI and Amazon transcribers.

Table 5.1 provides a summary of the error distribution for Amazon Transcribe and RevAI. As shown, Amazon Transcribe has a higher rate of Missing token errors, whereas RevAI exhibits a higher error rate in the other three error categories. This suggests that Amazon Transcribe is more likely to omit tokens during the ASR process, possibly skipping segments when the audio is unclear or difficult to recognize.

In contrast, RevAI tends to preserve most of the information during the ASR process, as evidenced by its higher error rates in the other categories. This is also reflected in the Overlapping errors, where Amazon Transcribe does not transcribe any overlapping utterances, while RevAI transcribes these segments, leading to errors.

5.3.2 Result

For each algorithm, we first generate the correct mapping with the produced reference and hypothesis text as the reference mapping result. Then we remove all gaps and let the algorithm align the two texts giving out the generated mappings as the hypothesis mapping result. We measure the accuracy of sequence alignment by calculating the percentage of tokens in reference text that have the same mapping between the reference mapping result and hypothesis mapping result. The final accuracy is calculated as the average among all 10 transcripts.

Algorithm	Avg. accuracy
Pairwise Character	0.92
Pairwise Token	0.93
Multi-sequence	0.99

Table 5.2: Average accuracy for alignment between three proposed alignment algorithm on simulated transcript.

Table 5.2 presents the performance of our multiple sequence alignment algorithm compared to the baseline character-level pairwise alignment and token-level pairwise alignment. As shown, multi-sequence alignment gives a significant improvement comparing to pairwise alignment algorithms, due to the ability of aligning tokens in the wrong orders. We therefore choose multi-sequence alignment to conduct compute evaluation metrics in the next section.

5.4 Evaluation of Proposed Metrics

5.4.1 Data Preparation

We manually selected 10 conversations from the CallHome Corpus based on their audio quality. Each conversation lasts around 30 minutes, but the reference transcript only covers 10 minutes of the audio. Therefore, we cut each of the audio into a 10-minute clip and transcribe it using Amazon and RevAI separately.

The transcription output from both RevAI and Amazon Transcribe is provided in JSON format. For RevAI, the output consists of a list of monologues, with each monologue containing speaker information and a list of elements representing individual tokens, including text, punctuation, and timestamps. In contrast, Amazon's output is structured with separate lists for transcripts, speaker labels, and items. The transcripts list contains the entire transcript as a single string, while the speaker labels list stores diarization results as speaker segments. The items list contains individual tokens with timestamps and confidence scores.

5.4.2 Speaker Alignment

To evaluate TDER, we first need to perform speaker alignment between the reference and hypothesis transcripts. For this purpose, we use the Hungarian algorithm, which is an efficient method to find the optimal assignment in a square cost matrix, minimizing the total cost [7].

In our case, the cost matrix represents the errors of assigning reference speakers to hypothesis speakers. For instance, if we have three reference speakers (R1, R2, and R3) and two hypothesis speakers (H1 and H2), our cost matrix would be:

	Hyp Speakers	
Ref Speakers	H1	H2
R1	C11	C12
R2	C21	C22
R3	C31	C32

Table 5.3: Example cost matrix for speaker alignment

With the cost matrix, the Hungarian algorithm determines the optimal assignment by minimizing the total cost. In our implementation, we use the `linear_sum_assignment` function from the SciPy library, which is based on the Hungarian algorithm, to perform the speaker alignment. Once the speaker alignment is completed, we can proceed to evaluate speaker diarization quality using TDER and other chosen metrics.

5.4.3 Result

Table 5.4 shows the comparison of TDER and F_1 scores with DER, WER, and WDER on the selected 10 conversations. Each entry is the average of 10 conversations score.

Amazon performs better in terms of DER score, which indicates improved segmentation of speaker utterances at the audio level. However, RevAI outperforms Amazon in ASR tasks, as evidenced by a WER of 0.29, which is 0.05 lower than Amazon's WER.

Transcriber	DER	TDER	F_1	P	R	WER	WDER
AMZN	0.24	0.53	0.79	0.87	0.73	0.34	0.15
Rev	0.26	0.50	0.84	0.88	0.81	0.29	0.20

Table 5.4: DER and F1-score metric for ASR provided by Amazon and Rev AI. TDER: Text-based DER. WDER: Word-level DER

Despite Amazon's lower WDER score compared to RevAI, it is important to note that WDER only counts aligned tokens, as mentioned in section 2.2.1. Section 5.3.1 highlights Amazon's tendency to omit tokens during the ASR process. This omission is further exemplified in table 5.1, which shows the manually labeled error distribution for both Amazon and RevAI. Consequently, WDER primarily measures substitution errors, while ignoring the most frequent error made by Amazon—missing tokens.

This propensity for Amazon to drop tokens not only skews the WDER results but also leads to a lower recall score compared to RevAI. The reduced recall score, in turn, contributes to a lower F1 score for Amazon.

When evaluating diarization performance in transcripts, measured by TDER, RevAI demonstrates a slight advantage, aligning with the F1 score results. TDER, F1, and WER collectively reveal a similar trend, with RevAI modestly outperforming Amazon on this corpus. While Amazon exhibits better speaker diarization performance, the ASR quality affects the results as they are reflected in the transcripts. Therefore, TDER demonstrates the ability to reflect the transcript's diarization quality.

Chapter 6

TranscribeView: A System for Transcript Evaluation and Diarization Error Visualization

6.1 Interface

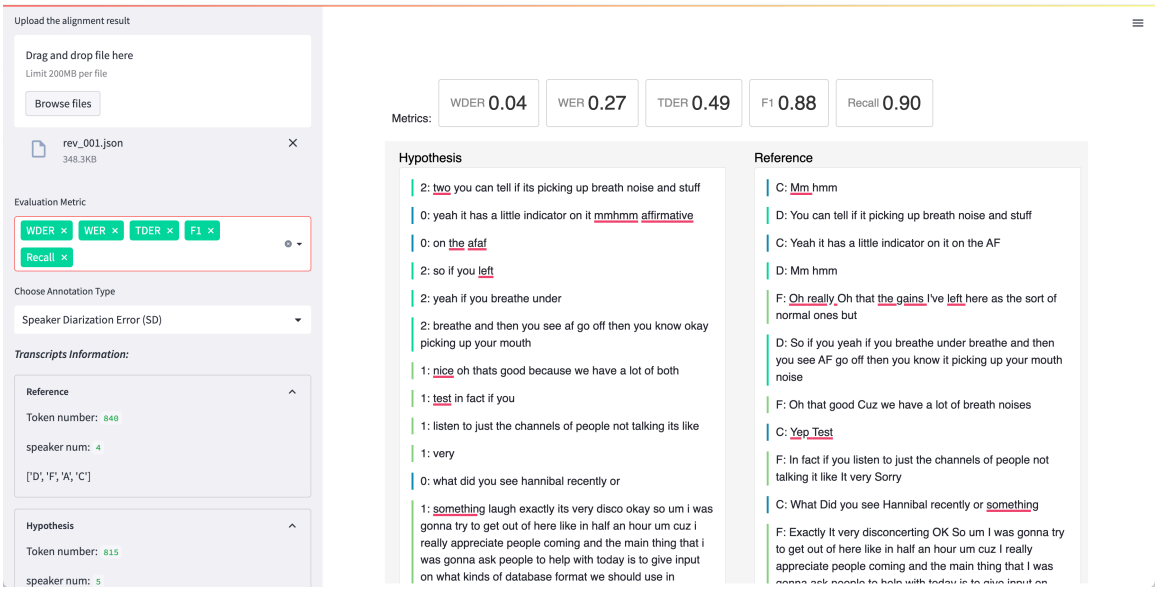


Figure 6.1: Screenshot of system interface

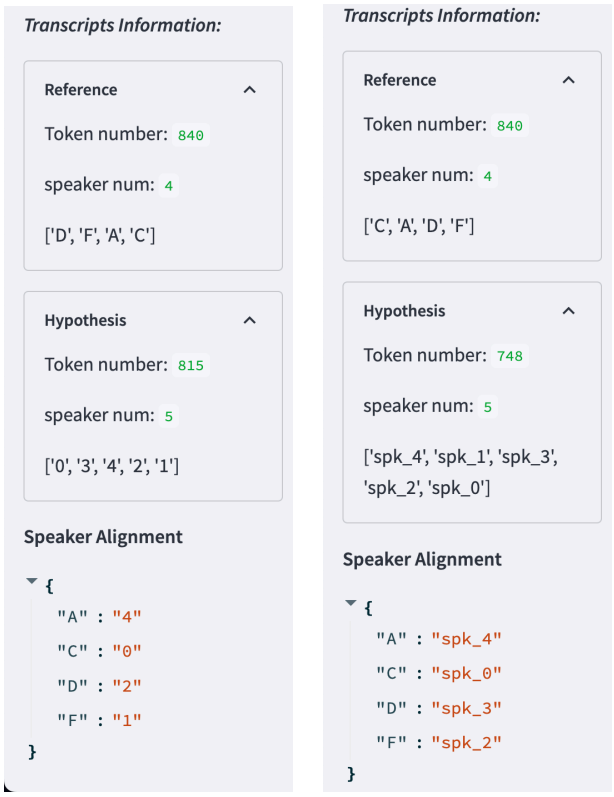
As shown in Figure 6.1, the TranscribeView system interface comprises a left sidebar and a visualization area, working together to deliver a comprehensive evaluation of transcripts. The left sidebar allows users to upload transcripts in JSON format and choose from various evaluation metrics. Upon uploading, the system presents statistical information about the transcript, such as the number of speakers and tokens. Additionally, users have the option to highlight diarization or ASR errors.

The visualization area, situated to the right of the sidebar, showcases the selected metrics' scores at the top. This area is partitioned into two columns: one for the hypothesis transcript and the other for the reference transcript. Each column exhibits its respective transcript, with each utterance accompanied by a colored vertical bar that indicates the speaker ID and the speaker mapping between the two transcripts. Furthermore, users can hover over tokens to view the corresponding aligned tokens in the other transcript.

6.2 Implementation

TranscribeView's implementation relies on the Streamlit framework in conjunction with customized HTML elements. The interface structure and left sidebar section are developed using Streamlit, while the visualization area is created with embedded HTML elements. Streamlit offers an API for adding HTML strings as iFrame elements, enabling the incorporation of CSS and JavaScript elements into the HTML string to enhance interactivity in the visualization area.

TranscribeView's evaluation is based on the alignment results from align4d. Scripts are provided to preprocess the alignment results into JSON format, which is utilized by the visualization area. In addition to the interface, metric APIs are available for users to apply the evaluation metrics independently of the interface.



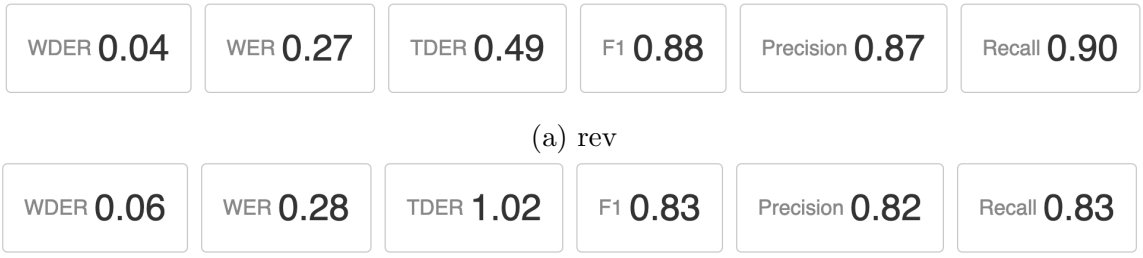
(a) rev (b) amazon

Figure 6.2: Amazon and RevAI's transcripts information after uploading alignment algorithm

6.3 Case Study: Comparing Transcribers

We now demonstrate how we can use TranscribeView to compare the performance of two transcriber systems (Amazon, RevAI) on the same audio data. The case study also shows why TDER and diarization F1 give a more comprehensive evaluation of transcript quality. The audio data and reference transcripts are from ICSI meeting dataset. We randomly cut a 5-min meeting audio clip and input it separately into Amazon and RevAI's transcriber.

Transcript summarization: Upon uploading the alignment output JSON file, the summarization of the transcript's information is shown at the bottom of the sidebar. As in figure 6.2, in the reference transcript, there are 840 tokens with four speakers. RevAI's output contains 815 tokens and Amazon's output only contains



(a) rev

(b) amazon

Figure 6.3: Screenshots for metrics area. Metrics from left to right are: WDER, WER, TDER, diarization F1, Precision, and Recall

748 tokens. Both of the transcribers tend to output fewer tokens, but Amazon drops 11% of the reference tokens. This type of error will be reflected in the metric area. For speaker diarization, both transcribers recognized five speakers although there are only four speakers in the original audio. As a result, shown in the speaker alignment area, one of the speakers in the hypothesis transcript is not aligned to any reference speakers (speaker 3 for RevAI, speaker *spk*_1 for Amazon).

Metric Comparison: The top of the visualization area shows the selected evaluation metrics. Figure 6.3 shows the selected evaluation metrics for the two transcribers. When looking at WDER and WER, the two transcribers show us a similar performance for both SD and ASR. However, TDER and diarization F1 score indicate there is a significant difference in speaker diarization quality. Having more tokens dropped in Amazon's output causes a significant difference in the Recall score. This error is ignored in WDER as mentioned in section **??** that WDER only accounts for the errors in mapped tokens, but errors of the missing token also bring diarization errors to transcripts. Overall, as shown in metrics, RevAI's transcript seems to have higher quality, especially for the reflected speaker diarization result.

Exploring Visualization: Statistical summarization and metric sometimes may not be intuitive to identify the transcript's quality. Figure 6.4 shows the annotations of hypothesis transcript and reference transcript side by side.

Each colored vertical bar shows the alignment between utterance speakers (*spk*_0

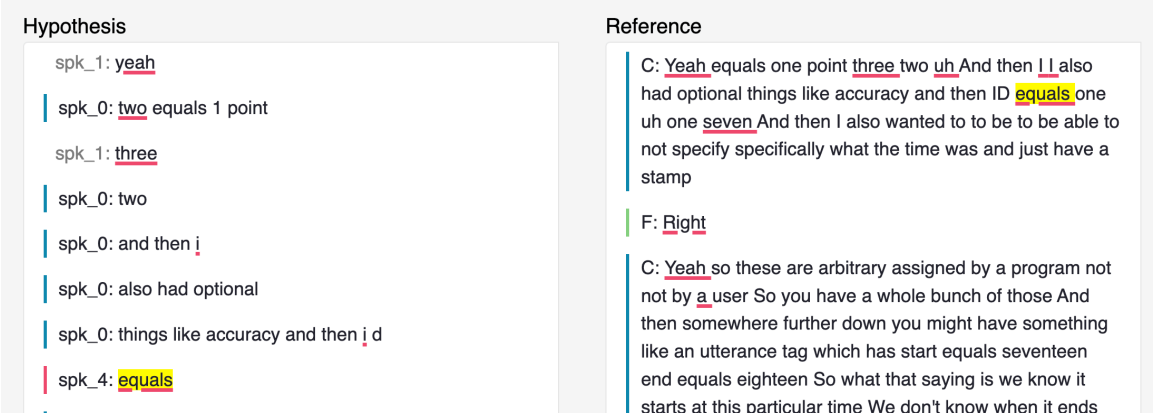


Figure 6.4: Screenshot of visualization area aligning Amazon's output with reference transcript.

is aligned to C). Greyed out speaker label indicates unmapped speakers (spk_1 is not mapped to any speakers in reference). Highlighted tokens are hovered by cursor. The corresponding aligned token is also highlighted. Here red underline indicates diarization errors.

Chapter 7

Conclusion

In conclusion, this thesis presents a novel approach to evaluating speaker diarization performance in text transcripts by introducing the Text-based Diarization Error Rate (TDER) and diarization F1-score metrics. These metrics account for both automatic speech recognition (ASR) and speaker diarization (SD) errors, providing a more comprehensive assessment of transcript quality. To overcome the inconsistency in the number of tokens between hypothesis and reference transcripts, we also developed a multi-sequence alignment tool that enables accurate word-to-word mapping between reference and hypothesis transcripts, achieving a higher accuracy score than pairwise alignment methods on a simulated corpus generated based on common SD and ASR errors.

Our evaluation of TDER, F1-score, DER, WER, and WDER on 10 transcripts from the CallHome dataset demonstrates that TDER and F1-score provide more reliable evaluations of speaker diarization performance at the text level compared to existing metrics. Moreover, we introduced TranscribeView, a web-based platform for evaluating and visualizing errors in speech recognition and speaker diarization. To the best of our knowledge, TranscribeView is the first comprehensive platform that allows researchers to align multi-sequence transcripts, assess, and visualize speaker

diarization errors, which is essential for advancing data-driven conversational AI research.

7.1 Limitations

Despite the contributions and advancements made in this study, it is important to acknowledge certain limitations that may impact the interpretation of the results and the applicability of our provided tool. When evaluating the correctness of our alignment tool, we used simulated data based on the statistical patterns observed in four transcripts. Annotating more transcripts could lead to more accurate summarizations. Additionally, incorporating a language model in the simulation might make the simulated transcript more closely resemble real hypothesis transcripts.

Furthermore, while the improved alignment algorithm increases alignment accuracy and enables text-based speaker diarization, this improvement comes at the cost of increased computational resources. The time and space complexity of the algorithm is $O(n^k)$, where n represents the average number of tokens in one sequence, and k represents the number of sequences, which is equal to the number of speakers plus one. This increased complexity may limit the applicability of the tool in some scenarios, particularly those involving large numbers of speakers or extensive transcripts.

7.2 Future Work

In future work, we aim to find solutions to reduce the runtime of our multiple sequence alignment algorithm. One possible approach is to pre-segment the input sequence into shorter subsequences and align these smaller segments, which could potentially improve computational efficiency. Additionally, we plan to work on enhancing the robustness and design of the TranscribeView system, ensuring it provides an improved

user experience and can handle a wider range of transcript evaluation scenarios.

Bibliography

[1] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. volume 20, pages 356–370, 2012. doi: 10.1109/TASL.2011.2125954.

[2] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf`.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[4] Jonathan G. Fiscus, Jerome Ajot, Nicolas Radde, and Christophe Laprun. Multiple dimension Levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/197_pdf.pdf.

[5] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proceedings of the International Speech Communication Association Conference*, INTERSPEECH'20, pages 5036–5040, 2020. URL http://www.interspeech2020.org/index.php?m=content&c=index&a=show&catid=418&id=1331.

[6] Dietrich Klakow and Jochen Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28, 2002. ISSN 0167-6393. doi: https://doi.org/10.1016/S0167-6393(01)00041-3. URL https://www.sciencedirect.com/science/article/pii/S0167639301000413.

[7] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[8] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710, 1965.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, transla-

tion, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[10] 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.

[11] NIST. Nist fall rich transcription on meetings 2006 evaluation plan, 2006. 2006.

[12] Tae Jin Park and Panayiotis Georgiou. Multimodal Speaker Segmentation and Diarization Using Lexical and Acoustic Cues via Sequence to Sequence Neural Networks. In *Proc. Interspeech 2018*, pages 1373–1377, 2018. doi: 10.21437/Interspeech.2018-1364.

[13] Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J Han, Shinji Watanabe, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning, jan 2021. URL https://arxiv.org/abs/2101.09624.

[14] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision, 2022. URL https://arxiv.org/abs/2212.04356.

[15] 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. *Journal of Machine Learning Research*, 21(1):5485–5551, 2020. URL https://jmlr.org/papers/volume21/20-074/20-074.pdf.

[16] Laurent El Shafey, Hagen Soltau, and Izhak Shafran. Joint speech recognition and speaker diarization via sequence transduction. 2019. doi: 10.48550/ARXIV. 1907.05337. URL https://arxiv.org/abs/1907.05337.

[17] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. BlenderBot 3: A Deployed Conversational Agent that Continually Learns to Responsibly Engage, 2022. URL https://arxiv.org/abs/2208.03188.

[18] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.