Towards Hierarchical Spoken Language Dysfluency Modeling

Anonymous ACL submission

Abstract

Speech dysfluency modeling is the bottleneck for both speech therapy and language learning. However, there is no AI solution to systematically tackle this problem. We first propose to define the concept of *dysfluent speech* and *dysfluent speech modeling*. We then present *Hierarchical Unconstrained Dysfluency Modeling* (H-UDM) approach that addresses both dysfluency transcription and detection to eliminate the need for extensive manual annotation. Furthermore, we introduce a simulated dysfluent dataset called VCTK^{++} to enhance the capabilities of H-UDM in phonetic transcription. Our experimental results demonstrate the effectiveness and robustness of our proposed methods in both transcription and detection tasks.

1 Introduction

Spoken language dysfluency modeling is the core technology in speech therapy and language learning. According to NIDCD (2016), an estimated *17.9 million* adults and *1.4 percent* of children in the U.S. suffer from chronic communication and speech disorders. Currently, hospitals have to invest substantial resources in hiring speech and language pathologists (SLPs) to manually analyze and provide feedback. More importantly, the cost is not affordable for low-income families. Kids' speech disorders also have a significant connection to the language learning market. According to a report by VCL (2021), the English language learning market will reach an estimated value of *54.8 billion* by 2025. Unfortunately, there is not an AI tool that can effectively automate this problem.

In current research community, there is not a unified definition for dysfluent speech. As such, we *first propose to define* dysfluent speech as any form of speech characterized by abnormal patterns such as repetition, prolongation, replacement, and irregular pauses. Dysfluencies can happen either in speech disorders such as stuttering, aphasia (Brady et al., 2016), and dyslexia (Snowling and Stackhouse, 2013), or in normal conversational speech (Pitt et al., 2005), where individuals may experience hesitations while speaking. Within the domain of *dysfluent speech modeling*, research efforts are conducted both on the speech side and the language side. Whenever dysfluent speech transcription is given(such as *human transcription* in Figure 1), the problem can be tackled by LLMs (ChatGPT, 2022). However, such transcription is not available and current best ASR systems such as Radford et al. (2023) tend to recognize them as perfect speech. Thus, we argue that the bottleneck lies in the *speech side* rather than in language.

Unfortunately, there is also no established definition for the problem of speech dysfluency modeling. We *first propose to define* that speech dysfluency modeling is to detect all types of dysfluencies at both the word and phoneme levels while also providing a time-stamp for each type of dysfluency. In other words, dysfluency modeling should be hierarchical and time-accurate. Previous research has mainly focused on addressing a small aspect of this problem and can be broadly categorized as *transcription* and *detection*.

Current state-of-the-art word transcription models (Radford et al., 2023; Zhang et al., 2023; Pratap et al., 2023; Aghajanyan et al., 2023) can only transcribe certain obvious word-level dysfluency patterns, such as word repetition or replacement. However, the majority of dysfluencies occur at the phoneme-level or subword-level, making them challenging for any ASR system to explicitly detect. As time-accurate detection is required, *phonetic alignment* might be a better representation to capture various dysfluency types. Another requirement is that phonetic alignment should be sensitive to silences as it might indicate a block or poor breath-speech coordination. Kouzelis et al. (2023) recently proposed a time-accurate and silence-aware neural forced aligner, where a weighted finite-state transducer (WFST) is introduced for modeling dysflu-

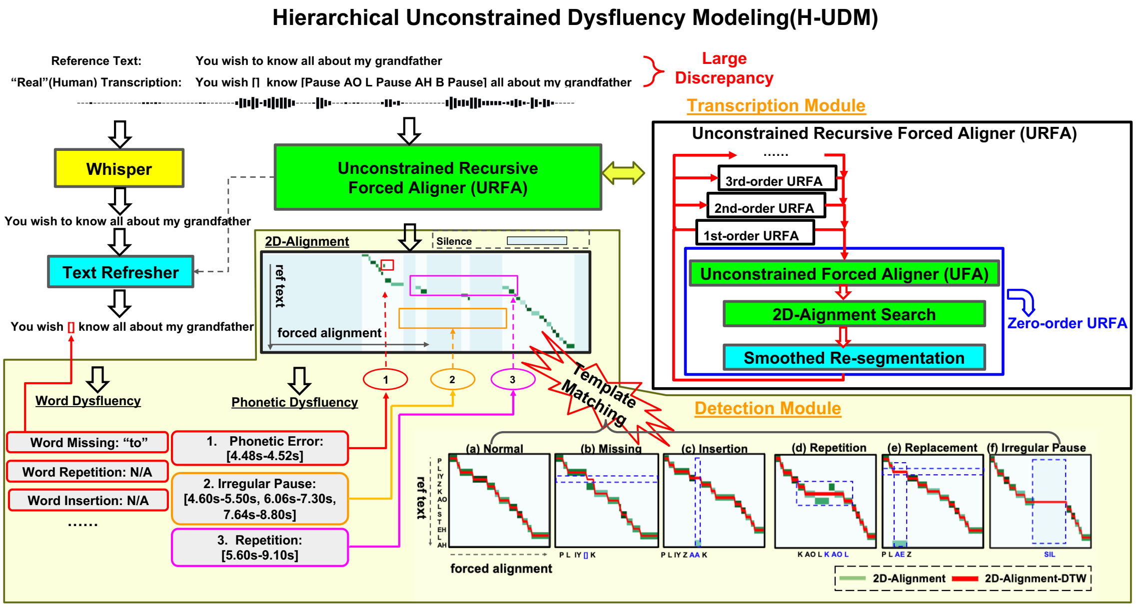

Figure 1: Hierarchical Unconstrained Dysfluency Modeling(H-UDM) consists of *Transcription* module and *Detection* module. Both word-level and phoneme-level dysfluencies are detected and localized. Here is an example of aphasia speech. The reference text is "You wish to know all about my grandfather," while the real/human transcription differs significantly from the reference. Whisper (Radford et al., 2023) recognizes it as perfect speech, while H-UDM is able to capture most of the dysfluency patterns. An audio sample of this can be found here[1].

ency patterns such as repetition. However, this approach assumes there is minimal deviation between the reference and "real" transcribed text. In real-life dysfluent speech, such as the example shown in Figure 1, this assumption may not hold true.

Research on speech dysfluency detection has traditionally been conducted independently of transcription and has recently been dominated by end-to-end methods. These approaches typically focus on either utterance-level detection (Kourkounakis et al., 2021; Alharbi et al., 2017, 2020; Jouaiti and Dautenhahn, 2022), or frame-level detection (Harvill et al., 2022; Shonibare et al., 2022). However, these studies primarily address data-driven classification problems and do not explicitly incorporate dysfluency transcription into their detection methods. More importantly, speech dysfluency detection must be dependent of text. For example, if the reference text is "you wish you wish" and we read that text, there is no dysfluency (stuttering). This crucial aspect has been ignored in all of the previous work. A unified framework that integrates transcription and detection is essential to develop a robust dysfluency modeling system.

In this study, we propose an *Hierarchical Unconstrained Dysfluency Modeling* (H-UDM) approach that integrates dysfluent speech transcription and detection in an automatic manner with no human effort. It is *unconstrained* because real transcription for dysfluent speech is unknown (as shown in the "Human Transcription" in Figure 1, which is largely different from reference text). In *transcription* module, we first introduce *Unconstrained Recursive Forced Aligner* (URFA) to iteratively generate phoneme alignment (1D) and 2D-Alignment with weak text supervision. We also propose a *Text Refresher* that leverages the 2D-Alignment from URFA to refine the state-of-the-art Whisper (Radford et al., 2023) transcription. In *detection* module, we pre-define 2D alignments for 5 types of phoneme-level dysfluencies (*missing, insertion, replacement, repetition, irregular pause*) and 4 types of word-level dysfluencies (*missing, insertion, replacement, repetition*). We then simply perform the template matching between these templates and the 2D-Alignment from URFA to generate time-accurate detection results. The entire pipeline is shown in Fig 1. To further enhance performance, we curate a dysfluent dataset called VCTK^{++} to boost the capacity of URFA. Experimental results demonstrate the effectiveness of our proposed framework in both dysfluent speech transcription and dysfluency pattern detection.

[1]Fig.1 Audio samples. (1) Aphasia Speech Sample: https://shorturl.at/eTWY1. (2) Template speech samples: https://shorturl.at/bszVX

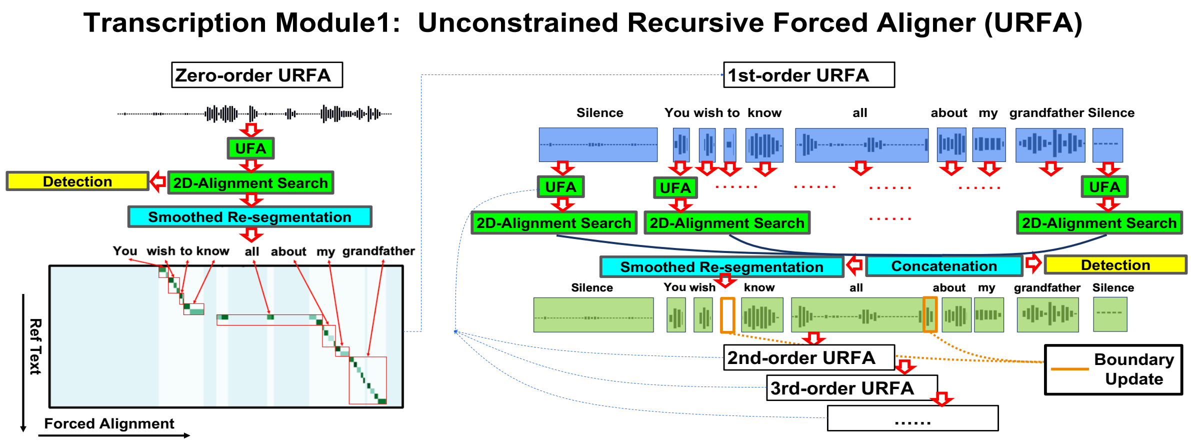

Figure 2: Unconstrained Recursive Forced Aligner consists of three basic modules: *UFA*, *2D alignment Search*, *Smoothed Re-segmentation*. In the first iteration (Zero-order), the entire utterance is taken and 2D alignment is generated. Starting at 2nd iteration (1st-order), the dysfluent speech is segmented at word level and each segment is processed separately and then combined to generate the final 2D alignment for detection.

2 Transcription Module

Our transcription module consists of two core parts: *(1) Unconstrained Recursive Forced Aligner*, which generates phonetic transcriptions (2D-Alignment), and *(2) Text Refresher* which takes both Whisper output and 2D-Alignment to generate word transcription, as shown in Fig. 1.

2.1 Unconstrained Recursive Forced Aligner

The bottleneck for dysfluent speech alignment is that the *real text transcription* is unknown, which is significantly different from the *reference text*, as shown in Fig. 1. However, dysfluency detection relies on the *reference text*. Traditional speech-text aligners (McAuliffe et al., 2017; Kim et al., 2021; Li et al., 2022) assume that the *reference text* is the same as the *real text transcription*, and thus they only work for normal fluent speech. Let's look at a simple example. If the *reference text* is "Y UW W IH SH (You Wish)" and the actual speech (*real text transcription*) is "Y UW W IH W IH SH (You Wi-Wish)," then the alignment from traditional aligners will all be "Y UW W IH SH" as monotonicity is enforced, which is not accurate. For dysfluent speech detection, deriving non-monotonic speech-text alignment is required, and this is achieved through the Unconstrained Forced Aligner (UFA). As dysfluency detection depends on the *reference text*, we also introduce 2D-Alignment to align the non-monotonic phoneme alignment with the *reference text*. Additionally, we deploy our alignment methods recursively, re-segmenting the utterance based on the 2D-Alignment to refine 2D-Alignment itself. The entire paradigm is illustrated in Fig. 2. Each sub-module is detailed in the following.

2.1.1 UFA

Unconstrained forced aligner (UFA) predicts alignment with weak text supervision. The speech segment is passed into WavLM (Chen et al., 2022) encoder which generates latent representations. A conformer module (Gulati et al., 2020) is followed to predict both alignment and boundary information. The alignment and boundary targets used in UFA are derived from the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017). During the inference stage, there is no need for text input, making the alignment process *unconstrained*.Two linear layers are simply applied as phoneme classifier and boundary predictor. For the phoneme classifier, UFA optimizes the softmax cross-entropy objective, while logistic regression is utilized for boundary prediction. Specifically, it predicts floating numbers between 0 (non-boundary) and 1 (boundary). We experimentally found that introducing an additional CTC (Graves et al., 2006) constraint (monotonicity) can enhance the robustness of our non-monotonic alignment. Note that CTC is involved only in training stage. See Appendix A for model details.

Dynamic Alignment Search For the inference of dysfluent speech, *real text transcription* is often not achievable, as discussed Sec. 2.1. Consequently, alignment should be decoded without text supervision. We propose a boundary-aware dynamic alignment search algorithm, which is the extension of Viterbi algorithm while there are two new updates. Firstly, instead of traversing along the monotonic target sequence, we conduct our search across all possible phonemes in the subsequent time step. Secondly, we must consider that the transition probability should be influenced by the boundary information. The intuition is that the transitions between consecutive phonemes near the boundary should be assigned lower importance to mitigate the risk of phoneme omissions. For instance, consider the correct alignment as SIL SIL SIL Y Y Y. In some cases, when the predicted probability for "Y" is low, there is a possibility that the prediction of "Y" might be overlooked due to the higher self-transition probability of SIL. Consequently, the final prediction could erroneously become SIL SIL SIL SIL SIL SIL. The bi-gram phoneme language model is derived by applying maximum likelihood estimation to the VCTK (Yamagishi et al., 2019) forced alignment obtained from MFA (McAuliffe et al., 2017). Details of the search algorithm are outlined in Algorithm 1 in appendix.

2.1.2 2D-Alignment Modeling

As dysfluency detection depends in *reference text*, we are going to align the phoneme alignment from UFA to *reference text*, named **2D-Alignment**. We extract the phoneme center embeddings from the phoneme classifier in the Unconstrained Forced Aligner (UFA) (Fig. 4). By obtaining the phoneme embedding sequences for both the reference text and the forced alignment, we compute the dot product between these sequences. As a result, we generate a 2D similarity matrix that serves as the alignment representation. In the forced alignment, each phoneme may align with multiple occurrences of the same phoneme in the reference text, particularly when the reference text contains repeated phonemes. For instance, in the phrase "Please call Stella" represented as "P L IY Z K AO L S T EH L AH," each occurrence of "L" in the forced alignment aligns with all three "L" phonemes in the reference text. To ensure that only one phoneme in the reference aligns with the current phoneme in the forced alignment, we develop *2D-Alignment Search*, which adopts Viterbi Algoithm, on the 2D similarity matrix. This process yields the final 2D alignment, which is primarily monotonic. As illustrated in Figure 1 and Figure 3, the alignment-2d is visualized through green plots, highlighting the relationship between the forced alignment and the reference text. In addition to *Alignment-2d*, we also require a *ground truth 2D-Alignment*, which represents the expected alignment between the forced alignment of nearly perfect speech and the reference text. This ground truth alignment is strictly monotonic. To obtain it, we apply Dynamic Time Warping (DTW) between the forced alignment and the reference text, resulting in the alignment represented by the red plots in Fig. 1 and Fig. 3. We denote this as **2D-Alignment-DTW**, which is used in detection stage only.

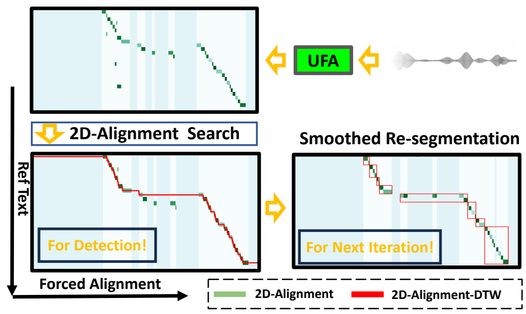

Figure 3: 2D-Alignment Modeling

Smoothed Re-segmentation and Recursive Alignment The generation of non-monotonic alignment inherently introduces variances that can lead to misdetection. To address this issue, we propose segmenting the dysfluent speech by word boundaries and generating alignment for each segment, potentially mitigating the problem. For instance, consider the case illustrated in Fig. 1 and Fig. 2, where the sequence [AO L Pause AH B] actually corresponds to the word "all." Another source of variance arises when individuals utter sequences like "AH, AO, AY," which may indicate the repetition of the phoneme "AH." However, our 2D alignment treats them as distinct phonemes, failing to detect the repetition, which poses a significant challenge. To tackle this issue, we introduce a phoneme smoothing technique. Specifically, at each time step, we calculate the cosine similarity of phoneme embeddings for both 2D-Alignment and 2D-Alignment-DTW. If the similarity falls within a predefined threshold, we merge the 2D-Alignment into 2D-Alignment-DTW, as demonstrated in the final figure of Fig. 3. This process yields a monotonic 2D alignment, allowing us to identify word boundaries by simply locating each word along the "ref text" axis. These segmented results serve as input for 1st-order Unconstrained Forced Aligner (URFA), as depicted in Fig. 2. In 1st-order URFA, we compute a 2D-Alignment for each segment and subsequently concatenate them. This iterative approach can be extended to 2nd-order URFA, 3rd-order URFA, and beyond. It is important to note that the smoothed monotonic 2D-Alignment is exclusively used for segmentation purposes, while the original non-monotonic 2D-Alignment remains in use for detection. This recursive aligner yields improved word boundary detection, as exemplified in Fig. 2, where the boundaries obtained in 1st-order alignment outperform those of zero-order alignment in capturing dysfluencies.

2.2 Text-Refresher

State-of-the-art ASR models (Radford et al., 2023; Zhang et al., 2023; **?**) are commonly trained using a robust language model constraint, ensuring a high level of accuracy in transcribing dysfluent or disordered speech, thereby generating nearly perfect transcriptions. However, to perform word-level dysfluency analysis, it is necessary to introduce imperfections. In this study, we propose *Text Refresher* to achieve this objective.

First, we obtain a perfect transcription using Whisper-large (Radford et al., 2023). We then obtain its corresponding phoneme transcription using CMU dictionary (cmu). Subsequently, in *Text Refresher*, we perform Dynamic Time Warping (DTW) between the phoneme transcription of the Whisper output and the output of the Unsupervised Forced Aligner (UFA). Our primary focus is on identifying *insertions* and *deletions*. If a word (represented as a phoneme sequence) in the Whisper output does not align with the correct word (phoneme sequence) in the UFA output, we remove that word. For example, in the case illustrated in Figure 1, the word "to" is deleted. On the other hand, if a word (phoneme sequence) in the UFA output does not align with any word (phoneme sequence) in the Whisper output, we insert that word. Our observations indicate that in real-life dysfluent speech such as Aphasia speech, most word-level imperfections that Whisper cannot transcribe are primarily from deletions or insertions. It is important to note that URFA also generates word transcriptions. However, based on our findings, it exhibits inferior performance in word-level dysfluency detection compared to *text-refresher*. Therefore, we have opted to employ URFA exclusively for phonetic-level dysfluency detection.

2.3 Transcription Module Evaluation

2.3.1 Phonetic Transcription

In order to evaluate how accurately the speech is transcribed at the frame level, we report **Micro F1 Score** and **Macro F1 Score** (sklearn F1) of phoneme transcription. Note that our F1 scores evaluate how many phonemes are correctly predicted. This is different from (Strgar and Harwath, 2023) which evaluates how many time steps are correctly predicted as phonetic boundaries. In order to evaluate the phoneme segmentation performance within our methods, in additional to phoneme error rate (**PER**), we also propose the duration-aware phoneme error rate (**dPER**). dPER extends Phoneme Error Rate (PER) by weighing each operation (substitution, insertion, deletion) with its duration. See appendix A for details.

2.3.2 Imperfect Word Transcription

In contrast to conventional ASR tasks, evaluating the performance in word-level dysfluency analysis requires the utilization of imperfect word targets. In this study, we employ manual word annotation of disordered speech (Aphasia, Dyslexia) as the target reference and report the *imperfect Word Error Rate* (**iWER**). To evaluate the word segmentation, we calculate the Intersection over Union (IoU) between our predicted time boundaries from URFA and the ground truth boundaries from human annotations. If the IoU is greater than 0.5, the dysfluency is identified as detected. We also report the F1 score for this matching evaluation, referred to as the **Matching Score (MS)**.

3 Detection Module

We develop rule-based methods for detecting time-accurate phonetic-level dysfluencies, including *Phonetic Errors (Missing, Deletion, Replacement), Repetition, and Irregular Pause*. Our methods also cover word-level dysfluencies, including *Missing, Insertion, Replacement, and Repetition*.

3.1 Phonetic-Level Dysfluency Detection

Finally, the detection of phonetic dysfluency becomes straightforward with the availability of the alignment-2D and alignment-2D-DTW. As illustrated in Figure 1-Template, in the case of normal speech, these two alignments align perfectly with each other. However, if there is a significant decrease in the alignment-2D-DTW while lacking any intersection in the corresponding row, it indicates a **missing** phoneme, as depicted in Fig 1-Template-(b). If a row in alignment-2D-DTW encounters multiple columns in alignment-2D, and there are repeated phonemes present, it indicates a **repetition**. This is depicted in Figure 1-template-(d). Conversely, if a row in alignment-2D-DTW already aligns with alignment-2D and simultaneously aligns with the surrounding column in alignment-2D, it signifies an **insertion**. This is illustrated in Figure 1-template(c). If a row in alignment-2D-DTW does not overlap with any horizontal regions in alignment-2D, but only overlaps with a single vertical block in alignment-2D, it is recognized as a **replacement**. This is depicted in Figure 1-template(e). Lastly, any pauses occurring within a complete sentence are identified as **irregular pauses**, as shown in Figure 1-template(f). It should be noted that within this rule-based detection framework, the precise timing of all five types of dysfluencies can be accurately identified with a resolution of 20ms.

3.2 Word-level Dysfluency Detection

We address *missing*, *insertion*, *replacement*, and *repetition* as part of our word-level dysfluency detection. To detect word-level dysfluency, we follow a similar methodology as phonetic-level dysfluency detection, which involves obtaining **2D-Alignment** and **2D-Alignment-DTW**. However, in the case of word-level dysfluency, we do not utilize word embeddings. Instead, we employ perfect matching between the words in the reference and predicted texts, without the need for embedding dot product calculations. Duration, including silence, is not taken into account in this particular analysis as it is already incorporated in the phonetic component.

3.3 Dysfluency Evaluation

We conduct dysfluency evaluation on segments of Aphasia speech. In each Aphasia speech segment, manual annotations are made for all types of dysfluencies, including their accurate timing. For the evaluation of phonetic-level dysfluency, we report the **F1 score (Micro and Macro)** for dysfluency type identification. Additionally, we measure the accuracy of dysfluency detection in terms of time alignment. We apply **Matching Score (MS)**, as defined in Sec. 2.3.2. For the evaluation of word-level dysfluency, we simply report the F1 score (Micro and Macro) without considering the timing aspects.

4 Experiments

4.1 Datasets and Pre-processing

VCTK (Yamagishi et al., 2019) It is a multi-speaker accented corpus containing 44 hours of fluent speech. We randomly select 90% of speakers as training set and the remaining as dev set. VCTK is used to train *UFA*.

VCTK^{++} For each waveform in VCTK and its forced alignment (from MFA (McAuliffe et al., 2017)), we applied simulations regarding the following stutter types. **(i) Repetitions**: Phonemes are randomly sampled within the waveform, appended by a variable-length sample of silence. **(ii) Prolongations**: Phonemes are randomly selected. The sound sample containing the phoneme is then stretched by a random factor. **(iii) Blocks**: Phonemes are selected from a list of commonly blocked sounds, such as consonants. With each simulation, we maintain the alignments such that the phoneme timestamps line up with the individual stutters. See Appendix A for details. VCTK++ is used to train *UFA*.

Buckeye (Pitt et al., 2005) It contains over 40 hours of recordings from 40 speakers of American English. The corpus contains quite a few portions of dysfluent speech with time-accurate annotation. We follow (Strgar and Harwath, 2023) to make the train/test split. Buckeye is used for training *UFA* and for *Phonetic Transcription Evaluation*.

Disorded Speech From our clinical collaborators, our dysfluent data comprises ten participants diagnosed with Aphasia and three kids suffering from Dyslexia. It consists of audio recordings capturing interactions between patients and speech-language pathologists (SLPs). Our primary focus lies in the audio input of patients reading the Grandfather passage, resulting in approximately 20 minutes of speech data. The disordered speech dataset is employed for the evaluation of *Imperfect Word Transcription* and *Dysfluency Detection*.

4.2 Phonetic Transcription Experiments

We train UFA using three types of data: VCTK only, VCTK+Buckeye, and VCTK++. Additionally, we conduct an ablation study to examine the impact of the boundary-aware constraint in the dynamic search algorithm. This is achieved by removing the constraint from the search algorithm. Furthermore, we investigate two alternative

Method	WavLM Size	Training Data	Micro F1 (%, ↑)	Macro F1 (%, ↑)	dPER (%, ↓)	PER (%, ↓)	Micro F1 (%, ↑)	Macro F1 (%, ↑)	dPER (%, ↓)	PER (%, ↓)
			Buckeye Test Set				*VCTK++ Test Set*			
WavLM-CTC-VAD	Large	None	50.1	47.3	86.9	12.0	48.8	45.7	88.0	8.2
WavLM-CTC-MFA	Large	None	49.8	28.7	53.9	12.0	47.6	26.0	54.2	8.2
UFA	Base	VCTK	68.9	55.6	53.3	15.0	78.8	59.5	53.4	11.0
UFA	Base	VCTK+Buckeye	65.9	51.6	63.6	16.3	75.2	56.0	60.0	11.8
UFA	Large	VCTK+Buckeye	70.3	55.0	46.2	13.3	80.7	66.4	45.8	11.0
UFA	Large	VCTK	71.3	60.0	46.0	11.9	81.7	72.0	44.0	10.5
– Boundary-aware	Large	VCTK	68.9	52.0	49.9	12.8	78.4	62.9	47.8	10.7
+ CTC	Large	VCTK	68.9	52.0	49.9	10.2	78.4	62.9	47.8	7.8
UFA	Large	$VCTK^{++}$	73.5	64.0	41.0	11.5	93.6	90.8	38.0	9.2
– Boundary-aware	Large	$VCTK^{++}$	71.0	63.7	44.3	12.2	91.1	90.0	42.1	9.6
+ CTC	Large	$VCTK^{++}$	**77.2**	**68.7**	**40.3**	**9.5**	**92.0**	**90.9**	**39.8**	**6.4**

Table 1: Phonetic Transcription Evaluation

forced aligners for comparison purposes: WavLM-CTC-VAD and WavLM-CTC-MFA. In WavLM-CTC-VAD, we combine the CTC phoneme alignment (Kürzinger et al., 2020) obtained from WavLM-CTC (HugginFace-WavLM, 2022) with Voice Activity Detection (VAD) segmentation. By assigning blank tokens and incorporating silence segments identified using online Silero VAD (Team, 2021), we obtain a silence-aware transcription. In WavLM-CTC-MFA, we employ the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to derive silence-aware phoneme alignment. We utilize WavLM-CTC (HugginFace-WavLM, 2022) to generate the initial phoneme transcription, A pronunciation dictionary maps phonemes (as word-level items) to phonemes (as phonemic pronunciation breakdowns). Details can be checked in Appendix A. It is worth noting that UFA remains constant throughout the recursive process. Therefore, our evaluation focuses solely on the alignment produced by UFA rather than that of URFA, as the latter is directly proportional to the former. Phonetic transcription results are shown in Table. 1.

4.3 Imperfect Word Transcription Experiments

	iWER(%, ↓)			
URFA Config	Zero-order	1st-order	2nd-order	3rd-order
Whisper-Large	11.3	-	-	-
+Text Refresher	9.7	9.4	9.2	9.2
$+VCTK^{++}$	9.2	9.0	8.7	8.7
+CTC	**8.8**	**8.6**	**8.4**	**8.4**

Table 2: Word Transcription Evaluation

We utilize Whisper (Radford et al., 2023) as our baseline. We begin by presenting the results obtained directly from Whisper-large. Subsequently, we employ Text Refresher to refine the Whisper transcription and report the updated results. By default, Text Refresher incorporates the UFA-WavLM-Large-VCTK alignment. Additionally, for ablation purposes, we consider the UFA-WavLM-Large-$VCTK^{++}$ alignment as input, which demonstrated superior performance as indicated in Table 1. We also provide a report on various iterations of URFA, including zero-order, 1st-order, 2nd-order, and 3rd-order. The comprehensive transcription results are presented in Table 2.We subsequently select the optimal configuration from Table 2 and present the performance of word segmentation. As a baseline, we employ WhisperX (Bain et al., 2023), which reports the timing information for each word. The results are detailed in Table 3.

	MS(%, ↑)			
URFA Config	Zero-order	1st-order	2nd-order	3rd-order
Whisper-X	42.1	-	-	-
Ours	**77.4**	**79.4**	**81.2**	**81.4**

Table 3: Word Segmentation Evaluation

4.4 Dysfluency Detection

The preliminary experiments presented in Table 1 indicate that both WavLM-CTC-VAD and WavLM-CTC-MFA do not exhibit significant improvements in phonetic transcription performance. Furthermore, the joint training of the VCTK and Buckeye corpora does not enhance the overall performance. Hence, we restrict our evaluation to two variants of the Unconstrained Forced Aligner (UFA): UFA-WavLM-Large-VCTK and UFA-WavLM-Large-$VCTK^{++}$. To assess the efficacy of our rule-based detection algorithm, we also perform manual detection using the predicted alignment from URFA and human-created targets. We also provide a report on various iterations of URFA, including 1st-order, 2nd-order, and 3rd-order. The results are presented in Table 4 and Table 5. MS is *Matching Score*, as stated in Sec. 3.3.

4.5 Results and Discussion

4.5.1 Transcription Analysis

We begin by examining the phonetic transcription results, as presented in Table 1. Both WavLM-CTC-

VAD and WavLM-CTC-MFA demonstrate commendable zero-shot silence-aware phonetic transcription capabilities. However, their performance remains limited and is even inferior to the UFA trained with the WavLM base model. Interestingly, incorporating the Buckeye data during training does not yield any performance improvement. We hypothesize that the presence of noise in the Buckeye corpus, being a dysfluent dataset itself, hinders performance. Additionally, including the LibriSpeech dataset in VCTK training does not lead to performance enhancement. This suggests that UFA has already reached a certain limit in terms of data scalability. Consequently, the subsequent ablations and dysfluency detection experiments are conducted solely using UFA-WavLM-Large-VCTK. During our ablation study, we consistently observed performance improvements by incorporating boundary prediction information in the dynamic alignment search, as described in Section 2.1.1. Moreover, our experiments on VCTK^{++} consistently demonstrated enhanced performance compared to the original VCTK dataset, highlighting the robustness introduced by VCTK^{++}. Ultimately, the inclusion of CTC significantly enhances performance across all metrics. In terms of word transcription results, as shown in Table 2, we found that Whisper-Large exhibited the lowest performance due to its overpowering language modeling. However, with the introduction of Text Refresher and the incorporation of VCTK^{++} and CTC, we observed an improvement in the imperfect Word Error Rate (iWER), further boosting the overall performance. It is noteworthy that the recursive updating of alignment has a notable impact on performance enhancement, with the 3rd-order iteration outperforming the 2nd-order, which, in turn, outperforms the 1st-order iteration. We refrained from exploring additional iterations, as performance tends to approach saturation. This observation aligns with the findings from Fig. 2, where, after the 1st-order URFA iteration, the detection of dysfluent word boundaries surpasses that achieved in the zero-order iteration. The conslusion also holds true for dysfluent word segmentation results, reported in Table. 3. We also provide more examples in Appendix A to illustrate its effectiveness.

4.5.2 Dysfluency Analysis

Since there is no previous work on ierarchical (word/phoneme) and fine-grained (time-accurate) dysfluency detection models like ours, we conducted ablation experiments to compare our proposed rule-based detection methods against ourselves. The results, as shown in Table 4 and Table 5, indicate impressive performance in terms of F1 scores and matching scores (MS), demonstrating the ability of our methods to accurately capture most dysfluencies. In a consistent manner, the iterative update of alignment significantly influences the enhancement of performance for both word-level and phoneme-level detection. However, it is important to note that our methods still fall short of human detection performance, highlighting their inherent limitations.

URFA Settings	F1 (%, ↑)	MS (%, ↑)	Human F1 (%, ↑)	Human MS (%, ↑)
UFA-VCTK	62.4	55.2	90.4	85.6
UFA-VCTK^{++}	64.5	60.2	90.6	86.0
+1st-order	65.6	61.0	90.6	86.0
+2nd-order	67.0	62.7	90.6	86.0
+3rd-order	**67.2**	**62.8**	**90.7**	**86.2**

Table 4: Phonetic Dysfluency Detection Evaluation

Methods	F1 (%, ↑)	Human F1 (%, ↑)
Whisper-Large	64.0	86.4
+Text Refresher(VCTK)	66.8	88.0
+Text Refresher(VCTK^{++})	68.4	89.1
+1st-order	70.1	89.1
+2nd-order	73.0	89.3
+3rd-order	**73.1**	**89.3**

Table 5: Word Dysfluency Detection Evaluation

5 Conclusion and Limitations

We propose a hierarchical unconstrained dysfluency modeling (H-UDM) approach that combines transcription and detection, which has been proven effective in both tasks. However, there are several limitations that should be addressed in future research. First, our detection experiments primarily focus on disordered speech, which limits the generalizability. Future work should explore diverse and open-domain dysfluent datasets, which may lack manual annotations. Second, our approach relies on phoneme-level forced alignment as the key representation for detection. However, it is worth investigating alternative speech units such as articulatory units (Lian et al., 2022; Wu et al., 2023), to improve alignment modeling. Lastly, it is worth exploring the application of LLM-guided speech models (Gong et al., 2023) to advance dysfluency modeling in a prompt manner, which remains an open problem.

References

Cmu phoneme dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. 2023. Scaling laws for generative mixed-modal language models. *International Conference on Machine Learning*.

Sadeen Alharbi, Madina Hasan, Anthony JH Simons, Shelagh Brumfitt, and Phil Green. 2020. Sequence labeling to detect stuttering events in read speech. *Computer Speech & Language*, 62:101052.

Sadeen Alharbi, Anthony JH Simons, Shelagh Brumfitt, and Phil D Green. 2017. Automatic recognition of children's read speech for stuttering application. In *6th. Workshop on Child Computer Interaction (WOCCI 2017), eds. K. Evanini, M. Najafian, S. Safavi and K. Berkling*, pages 1–6. International Speech Communication Association (ISCA).

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. *Interspeech*.

Marian C Brady, Helen Kelly, Jon Godwin, Pam Enderby, and Pauline Campbell. 2016. Speech and language therapy for aphasia following stroke. *Cochrane database of systematic reviews*, (6).

ChatGPT. 2022. Chatgpt. https://chat.openai.com/.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass. 2023. Listen, think, and understand. *arXiv preprint arXiv:2305.10790*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

John Harvill, Mark Hasegawa-Johnson, and Changdong Yoo. 2022. Frame-level stutter detection. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2022, pages 2843–2847.

HugginFace-WavLM. 2022. Wavlm-ctc-hugginface. https://huggingface.co/microsoft/wavlm-large.

Melanie Jouaiti and Kerstin Dautenhahn. 2022. Dysfluency classification in stuttered speech using deep learning for real-time applications. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6482–6486.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad. 2021. Fluentnet: End-to-end detection of stuttered speech disfluencies with deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2986–2999.

Theodoros Kouzelis, Georgios Paraskevopoulos, Athanasios Katsamanis, and Vassilis Katsouros. 2023. Weakly-supervised forced alignment of disfluent speech using phoneme-level modeling. *Interspeech*.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. Ctc-segmentation of large corpora for german end-to-end speech recognition. In *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings 22*, pages 267–278. Springer.

Jingbei Li, Yi Meng, Zhiyong Wu, Helen Meng, Qiao Tian, Yuping Wang, and Yuxuan Wang. 2022. Neufa: Neural network based end-to-end forced alignment with bidirectional attention mechanism. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8007–8011. IEEE.

Jiachen Lian, Alan W Black, Louis Goldstein, and Gopala Krishna Anumanchipalli. 2022. Deep Neural Convolutive Matrix Factorization for Articulatory Representation Decomposition. In *Proc. Interspeech 2022*, pages 4686–4690.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

NIDCD. 2016. Nidcd. https://www.nidcd.nih.gov/health/statistics/quick-statistics-voice-speech-language.

Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Olabanji Shonibare, Xiaosu Tong, and Venkatesh Ravichandran. 2022. Enhancing asr for stuttered speech with limited data using detect and pass. *arXiv preprint arXiv:2202.05396*.

sklearn F1. Micro-macro-f1. https://scikit-learn.org/stable/modules/

Margaret J Snowling and Joy Stackhouse. 2013. *Dyslexia, speech and language: a practitioner's handbook*. John Wiley & Sons.

Luke Strgar and David Harwath. 2023. Phoneme segmentation using self-supervised speech models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1067–1073.

Silero Team. 2021. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad.

VCL. 2021. Vcl. https://vclenglish.com/54-8-billion-by-2025-

W. Verhelst and M. Roelands. 1993. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 554–557 vol.2.

Peter Wu, Li-Wei Chen, Cheol Jun Cho, Shinji Watanabe, Louis Goldstein, Alan W Black, and Gopala K. Anumanchipalli. 2023. Speaker-independent acoustic-to-articulatory speech inversion. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.

A Appendix

UFA Unconstrained forced aligner (UFA) predicts alignment with weak text supervision. As shown in Fig. 4, a speech segment is passed into WavLM (Chen et al., 2022) encoder which generates latent representations. A conformer module (Gulati et al., 2020) is followed to predict both alignment and boundary information. The alignment and boundary targets used in UFA are derived from the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017). During the inference stage, there is no need for text input, making the alignment process *unconstrained*. The conformer module comprises of four conformer (Gulati et al., 2020) encoder layers. The hidden size, number of attention heads, filter size, and dropout for each conformer layer are [1024, 4, 5, 0.1], [1024, 8, 3, 0.1], [1024, 8, 3, 0.1], [1024, 4, 3, 0.1] respectively. Two linear layers are simply applied as phoneme classifier and boundary predictor. For the phoneme classifier, UFA optimizes the softmax cross-entropy objective, while logistic regression is utilized for boundary prediction. Specifically, it predicts floating numbers between 0 (non-boundary) and 1 (boundary).

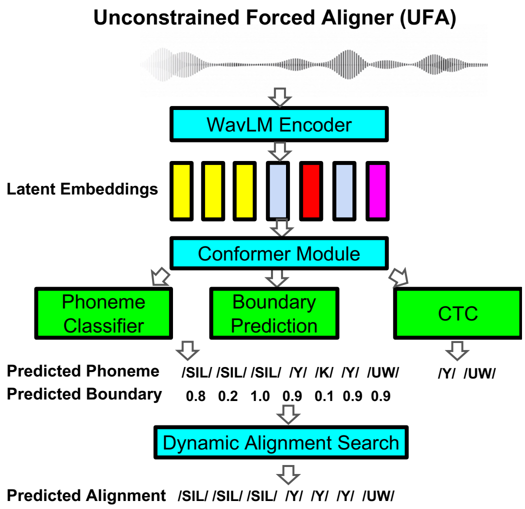

Figure 4: UFA Module

Dynamic Alignment Search We propose a boundary-aware dynamic alignment search algorithm, which is the extension of Viterbi algorithm. Let us denote the phoneme logits as *logits* $\in \mathbb{R}^{B,T,D}$, the boundary predictions as *boundaries* $\in \mathbb{R}^{B,T}$, and the bi-gram phoneme language model as *transition_probs* $\in \mathbb{R}^{D,D}$, where (B, T, D) represents the batch size, time steps, and phoneme dictionary size, respectively. The algorithm is presented as follows.

Algorithm 1 Boundary-Aware Dynamic Alignment Search

procedure DECODE(*logits*, *boundaries*, *transitional_probs*)
  $B, T, D \leftarrow$ shape of *logits*
  Initialize *trellis* and *backpointers*
  for t in range$(1, T)$ **do**
    for d in range(D) **do**
      trellis$[:, t, d]$, *backpointers*$[:, t, d] \leftarrow$ **MAX_ARGMAX**(*trellis*$[:, t-1, :] + (1 - $ *boundaries*$)[:, t] \times$ *transition_probs*$[d, :]$)
    end for
  end for
  Derive *best_path* from *trellis* and *backpointers*
  return *best_path*
end procedure

VCTK^{++} For each waveform in VCTK and its forced alignment (from MFA (McAuliffe et al., 2017)), we applied simulations regarding the following stutter types. **(i) Repetitions**: Phonemes are randomly sampled within the waveform, appended by a variable-length sample of silence, and inserted into the original sound file. The silence sample is set to vary between 200ms and 500ms in multiples of 20 to match the framerate of the phoneme alignments. **(ii) Prolongations**: Phonemes are randomly selected, excluding phonemes that cannot be reasonably prolonged, such as hard consonants or silence tokens. The sound sample containing the phoneme is then stretched by a random factor anywhere from 5x to 10x using Waveform Similarity Overlap-Add (WSOLA)(Verhelst and Roelands, 1993). The original phoneme is then replaced by the stretched variant in the waveform. **(iii) Blocks**: Phonemes are selected from a list of commonly blocked sounds, such as consonants or combinations of hard phonemes. With each simulation, we maintain the phoneme alignments such that the phoneme timestamps line up with the individual stutters, generating new alignments that act as ground truth for inference. See supplemental material for details. Here is an example of our augmented data. `https://shorturl.at/xBFG7`

Phonetic Dictionary We remove stress-aware phoneme labels (e.g. AE0, AE1→AE). The phoneme dictionary adopted in this paper contains 39 monophones from CMU phoneme dictionary (cmu) along with one additional silence label. For Buckeye corpus, we manually translate the out-of-dictionary phonemes into CMU monophones. Here is the translation paradigm: AEN→AE N, EYN→EY N, IYN→IY N, TQ→T, IHN→IH N, OWN→OW N, NX→N, EHN→EH N, DX→T, EN→AH N, OYN→OY N, EM→EH M, ENG→EH NG, EL→EH L, AAN→AA N, AHN→AH N, AWN→AW N.

Audio Segmentation For VCTK, we train on the entire utterance without segmentation. For Buckeye data, we follow (Strgar and Harwath, 2023) to segment the long utterance by the ground truth transcription. We make sure that the beginning and ending silence length would be no longer than 3s, resulting in the length of all segments ranging from 2s to 17s. Different from (Strgar and Harwath, 2023), we keep all silence labels but still remove the untranscriptable labels such as 'LAUGH', 'IVER', etc. For patient speech, we apply the online Silero VAD (Team, 2021) with a default threshold of 0.5 to make the segments. We keep all of the silences and this results in the length of all segments ranging from 2s to 15s. All audio samples have a sampling rate of 16K Hz.

Human Data Annotation For all disordered speech (aphaisa and dylexia), our co-workers work together to manually label the dysfluencies: types of dysfluency and its time stamp at both word and phoneme level. As the dysfluency patterns are straightforward to observe, each utterance is labelled by only one person.

dPER Definition Denote $\hat{S}, \hat{I}, \hat{D}, \hat{C}$ as the weighted value of substitutions, insertions, deletions, and correct samples. Denote p_i and p_j as the current two phonemes we are comparing in the reference sequence and prediction sequence respectively. Denote $d(p_i)$ and $d(p_j)$ as their durations (number of repetitions). Whatever the error type is detected, we propose the following updating rule: $\hat{S} \to \hat{S} + d(p_i) + d(p_j)$, $\hat{I} \to \hat{I} + d(p_j)$, $\hat{D} \to \hat{D} + d(p_i)$, $\hat{C} \to \hat{C} + |d(p_i) - d(p_j)|$. The ultimate formula is:

$$\text{dPER} = \frac{\hat{S}+\hat{D}+\hat{I}}{\hat{S}+\hat{D}+\hat{C}} \tag{1}$$

Phonetic Transcription Experiements Across all experiments, we utilize the same configuration settings, employing the Adam optimizer with an initial learning rate of 1e-3, which is decayed by 0.9 at each step. Each model converges after approximately 30 epochs, as determined by achieving a 90% phoneme classification accuracy on the development set. Each set of experiment takes about 12 hours on one A6000 GPU.

Configurations for two baseline forced aligner: WavLM-CTC-VAD and WavLM-CTC-MFA. In WavLM-CTC-VAD, we combine the CTC phoneme alignment (Kürzinger et al., 2020) obtained from WavLM-CTC (HugginFace-WavLM, 2022) with Voice Activity Detection (VAD) segmentation. By assigning blank tokens and incorporating silence segments identified using online Silero VAD (Team, 2021), we obtain a silence-aware transcription. The VAD threshold is set to the default value of 0.5, and the minimum and maximum speech durations are defined as 250ms and infinity, respectively. In WavLM-CTC-MFA, we employ the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to derive silence-aware phoneme alignment. We utilize WavLM-CTC (HugginFace-WavLM, 2022) to generate the initial phoneme transcription, and we leverage a pre-trained English ARPA acoustic model. A pronunciation dictionary maps phonemes (as word-level items) to phonemes (as phonemic pronunciation breakdowns). The default beam size of 10 is applied for MFA. In the phoneme-to-phoneme dictionary, the parameters for each phoneme mapping include a pronunciation probability of 0.99, a silence probability of 0.05, and final silence and non-silence correction terms of 1.0. For both methods, no additional training data is needed.

Word Segmentation Examples

GT denotes ground truth. Some samples might have multiple ground truths denoted as GT1, GT2, etc.

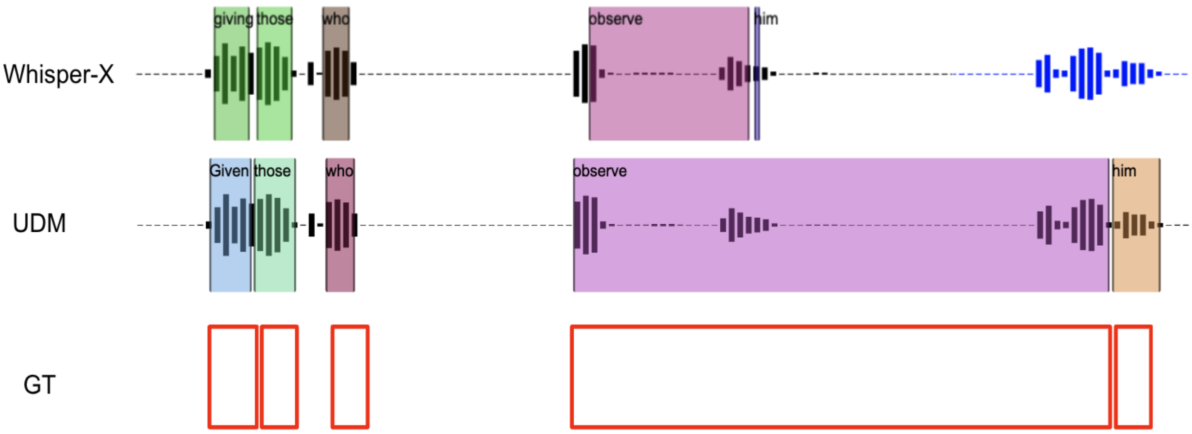

Figure 5: Segmentation-(Dyslexia Sample: Giving those who observe him)

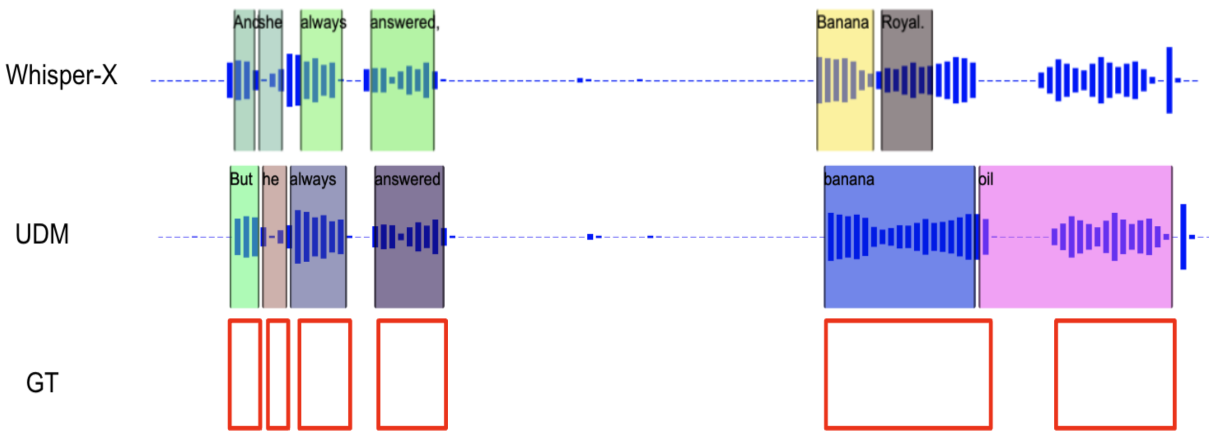

Figure 6: Segmentation-(Dyslexia Sample: But he always answered banana oil.)

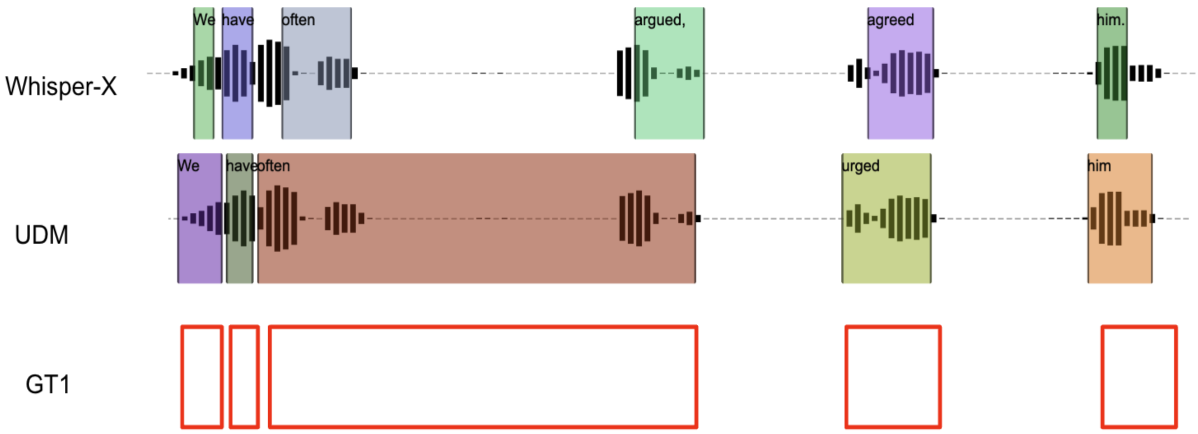

Figure 7: Segmentation-(Dyslexia Sample: We have often urged him)

Figure 8: Segmentation-(Aphasia Sample: Usually several buttons missing.)

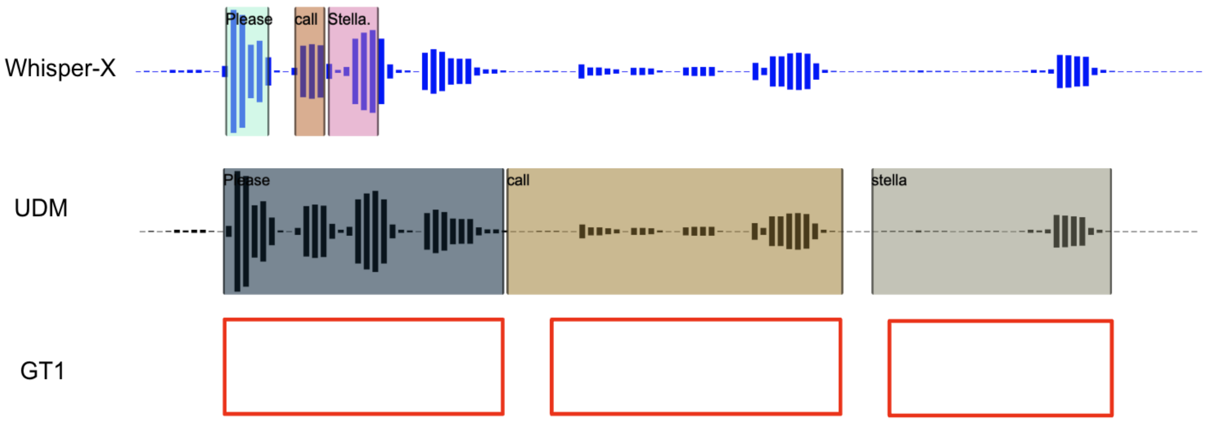

Figure 9: Segmentation-(My stutter sample: Please call stella.)