

Towards Hierarchical Spoken Language Disfluency Modeling

Jiachen Lian

UC Berkeley

jiachenlian@berkeley.edu

Gopala Anumanchipalli

UC Berkeley

gopala@berkeley.edu

Abstract

Speech disfluency modeling is the bottleneck for both speech therapy and language learning. However, there is no effective AI solution to systematically tackle this problem. We solidify the concept of *disfluent speech* and *disfluent speech modeling*. We then present *Hierarchical Unconstrained Disfluency Modeling* (H-UDM) approach, the hierarchical extension of Lian et al. (2023c) that addresses both disfluency transcription and detection to eliminate the need for extensive manual annotation. Our experimental findings serve as clear evidence of the effectiveness and reliability of the methods we have introduced, encompassing both transcription and detection tasks.

1 Introduction

Spoken language disfluency¹ 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 *disfluent speech*, as mentioned in Lian et al. (2023c). As such, we solidify the definition of *disfluent speech* as any form of speech characterized by abnormal patterns such as repetition, replacement, and irregular pauses, as summarized in Lian et al. (2023c). Within the domain

of *disfluent speech modeling*, research efforts are conducted both on the speech side and the language side. Whenever disfluent speech transcription is given (such as *human transcription* in Figure 1), the problem can be tackled by LLMs (OpenAI et al., 2023). 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 disfluency modeling. We formally define that *speech disfluency modeling* is to detect all types of disfluencies at both the word and phoneme levels while also providing a time-stamp for each type of disfluency. In other words, disfluency modeling should be **hierarchical** and **time-accurate**. Previous research has mainly focused on a small aspect of this problem.

Researchers started by focusing on spotting stuttering using end-to-end methods. They manually tagged each utterance and developed the classification model at the utterance level (Kourkounakis et al., 2021; Alharbi et al., 2017, 2020; Jouaiti and Dautenhahn, 2022). Later on, things got detailed with frame-level stutter detection (Harvill et al., 2022; Shonibare et al., 2022). However, end-to-end methods have their limitations. First, stuttering is just one aspect of disfluency. Current end-to-end models struggle to handle other forms of disfluency effectively. Second, manually labeling data for these methods is a lot of work and not practical for larger-scale projects. Lastly, disfluency modeling depends on the specific text being spoken, a factor that has been overlooked in previous research, as pointed out in Lian et al. (2023c).

It is typically intuitive to consider speech transcription that offers disfluency-specific representations. For a long time, the mainstream of researchers in speech transcription has been focused on word-level automatic speech recognition (ASR),

¹disfluency is interchangeable with dysfluency

Hierarchical Unconstrained Dysfluency Modeling(H-UDM)

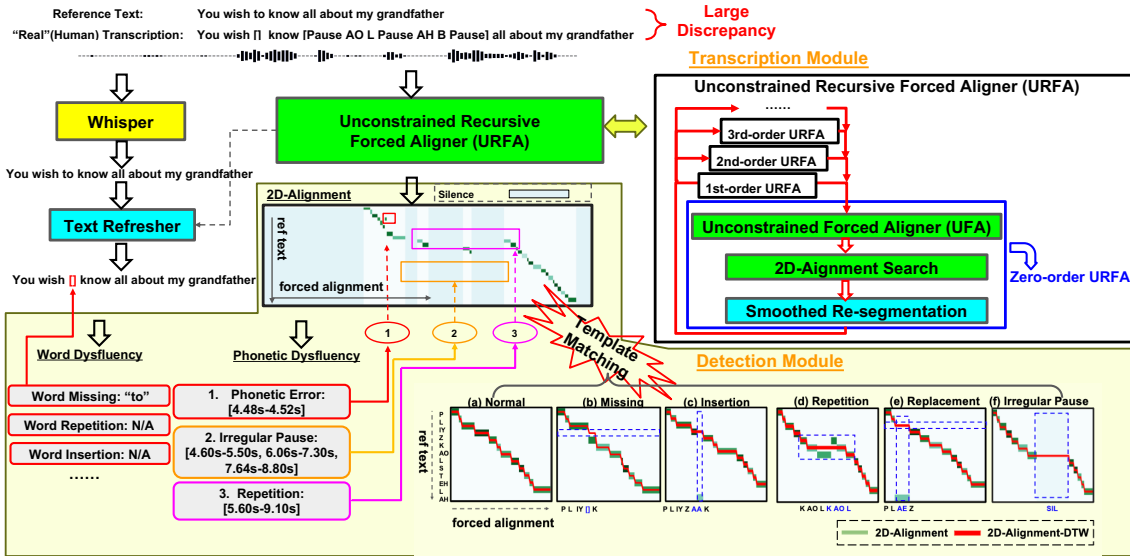


Figure 1: Hierarchical Unconstrained dysfluency Modeling(H-UDM) consists of *Transcription* module and *Detection* module. Both word-level and phoneme-level dysfluencies are detected and localized. Here is an example of aphasia speech. The reference text is "You wish to know all about my grandfather," 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 here².

which has been further scaled. However, The most advanced word transcription models currently available (Radford et al., 2023; Zhang et al., 2023; Pratap et al., 2023; Aghajanyan et al., 2023; Lian et al., 2023a) 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. Kouzelis et al. (2023) introduced a neural forced aligner that incorporates time accuracy and sensitivity to silence. This aligner employs a weighted finite-state transducer (WFST) to capture dysfluency patterns like repetition. However, it fails on openset dysfluency modeling (Lian et al., 2023c).

The Unconstrained Dysfluency Model (UDM) introduced in (Lian et al., 2023c) was devised to address the aforementioned challenges comprehensively. UDM seamlessly integrates both transcription and detection modules within a unified framework. Within the UDM framework, non-monotonic alignments are acquired through dynamic alignment search, forming the foundation for subsequent template matching algorithms aimed at detecting various dysfluency types. Specifically, distinct templates are tailored for each dysfluency category, encompassing replacements, insertions, deletions,

blocks, and repetitions. Additionally, VCTK++ dataset was introduced to further enhance model performance. In the present study, we extend the capabilities of UDM by incorporating a monotonicity constraint. While non-monotonic alignment is essential for effective dysfluency modeling, our experiments demonstrate that the integration of a simple Connectionist Temporal Classification (CTC) module alongside a phoneme classifier can enhance non-monotonicity. Furthermore, we introduce the *Unconstrained Recursive Forced Aligner* (URFA), which employs an iterative process to generate both phoneme alignments (1D) and 2D alignments with weak text supervision. This recursive modeling significantly enhances detection robustness. Our proposed method, termed *Hierarchical Unconstrained Dysfluency Modeling* (H-UDM), attains state-of-the-art performance in real aphasia speech dysfluency 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

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

Transcription Module1: Unconstrained Recursive Forced Aligner (URFA)

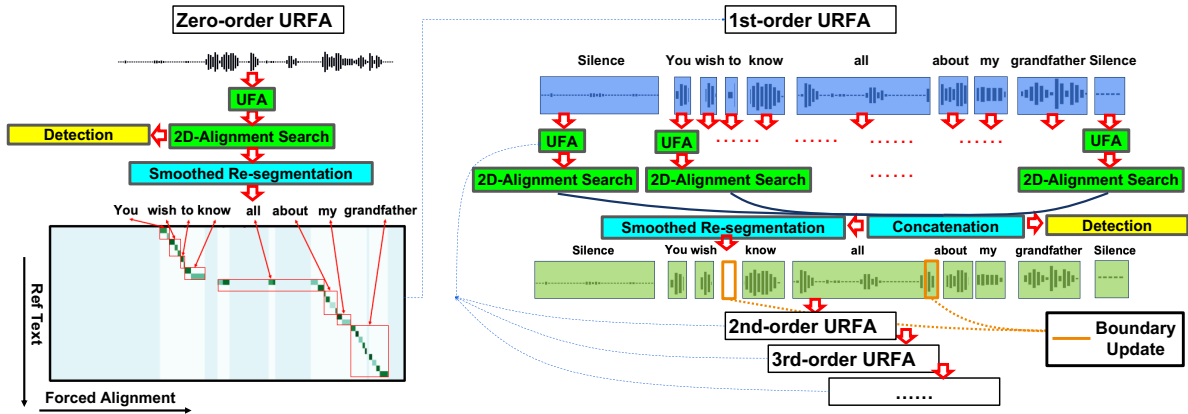


Figure 2: Unconstrained Recursive Forced Aligner consists of three basic modules: *UFA*, *2D alignment Search*, *Smoothed Re-segmentation*. In the first iteration (Zero-order), the entire utterance is taken and 2D alignment is generated. Starting at 2nd iteration (1st-order), the disfluent speech is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

output and 2D-Alignment to generate word transcription, as shown in Fig. 1.

2.1 Unconstrained Recursive Forced Aligner

The bottleneck for disfluent speech alignment is that the *real text transcription* is unknown, which is significantly different from the *reference text*, as shown in Fig. 1. However, disfluency 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 "K AE Y (Cat)" and the actual speech (*real text transcription*) is "K AE K AE T (Ca-Cat)," then the alignment from traditional aligners will all be "K AE T" as monotonicity is enforced, which is not accurate. For disfluent speech detection, deriving non-monotonic speech-text alignment is required, and this is achieved through the Unconstrained Forced Aligner (UFA) (Lian et al., 2023c). As disfluency 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

The Unconstrained Forced Aligner (UFA) (Lian et al., 2023c) operates by predicting alignments with the guidance of weak text supervision. Initially, the speech segment undergoes encoding by the WavLM (Chen et al., 2022) encoder, which generates latent representations. Subsequently, a conformer module (Gulati et al., 2020) is employed 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 requirement for text input, rendering the alignment process truly "unconstrained." To perform phoneme classification, UFA simply applies two linear layers. For the phoneme classifier, UFA optimizes the softmax cross-entropy objective, while logistic regression is utilized for boundary prediction. Notably, we found through experimentation that introducing an additional Connectionist Temporal Classification (CTC) constraint (Graves et al., 2006) (monotonicity) can enhance the robustness of our non-monotonic alignment. It's important to emphasize that CTC is solely involved in the training stage. For more in-depth model details, please refer to Lian et al. (2023c).

Dynamic Alignment Search We adopt the alignment search methodology proposed by Lian et al. (2023c). It is essential to note that, in the context of disfluency modeling, the alignment must

be non-monotonic. This stands in stark contrast to traditional forced aligners, which typically enforce monotonic alignment based on supervised signals such as text. However, in our case, text supervision is complicated by the substantial divergence between the real transcription and the reference text. Consequently, the reference text becomes an unreliable source for alignment. The process of decoding the alignment sequence from the emission matrix can be accomplished through various methods. In our approach, we follow the methodology outlined in Lian et al. (2023c) and apply the boundary-aware Viterbi algorithm for decoding. It is worth noting that the modified Viterbi algorithm introduces a computational complexity of $O(tN^2)$, where N represents the vocabulary size and t denotes the number of time steps. Given that, in practice, t is typically much larger than N , this computational complexity remains within acceptable bounds. The inclusion of boundary information proves invaluable in handling the ambiguity introduced, particularly by silence. In addition, we trained a phoneme autoregressive language model using the VCTK corpus (Yamagishi et al., 2019). Nevertheless, we did not observe a significant improvement in performance. Therefore, we opted to adhere to the approach outlined in Lian et al. (2023c) and continued to utilize the bi-gram model. For a more comprehensive understanding of the search algorithm, please refer to Lian et al. (2023c).

2.1.2 2D-Alignment Modeling

The concept of *2D-Alignment* was initially introduced in Lian et al. (2023c). The underlying idea revolves around a fundamental question: how accurately does the forced alignment correspond to the reference text? The 2D-Alignment was devised as a metric to assess this alignment. Specifically, the 2D-Alignment represents the temporal alignment between the actual spoken text by the speaker (ground truth text) and the disfluent alignment generated by the dynamic alignment search module. In the work presented in Lian et al. (2023c), this 2D alignment was computed by performing element-wise multiplication between the reference phoneme embeddings and the forced alignment phoneme embeddings. It is important to note that this 2D-Alignment is inherently non-monotonic. However, this approach has significant limitations. Through real speech testing, we observed that in the presence of noise, the noise can become erroneously aligned with parts of the reference text, which is

not desirable. Additionally, using phonemes as the primary units for disfluency modeling may not be optimal. For example, there may be minimal phonetic distinctions between certain phonemes, such as 'AH' and 'AO,' in terms of verbal pronunciation. Nonetheless, in both non-monotonic alignment and 2D-Alignment, they are treated as distinct phonemes and are considered uncorrelated. Despite these limitations, we still retained the ground truth 2D-Alignment for template matching algorithms. This ground truth 2D-Alignment, known as *2D-Alignment-DTW*, is always monotonic in nature. In the following section, we will delve into our strategies for addressing the aforementioned challenges.

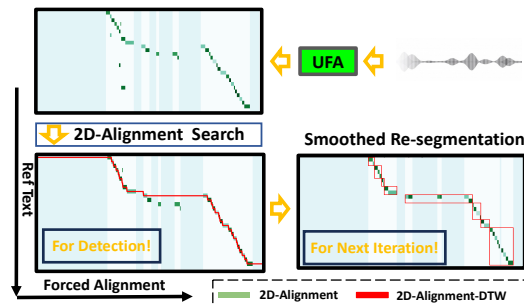


Figure 3: 2D-Alignment Modeling

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 disfluent 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 disfluencies.

2.2 ASR Scalability

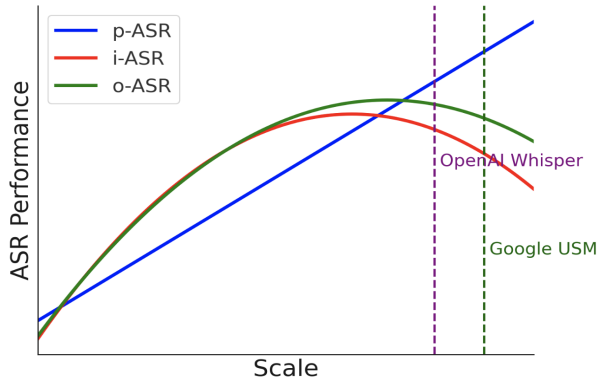


Figure 4: Scaling law for ASR under various conditions. (i) Perfect ASR (p-ASR); (ii) Imperfect ASR(i-ASR); (iii) Overall ASR(o-ASR)

Recent advances in spoken language processing (Pratap et al., 2023; Aghajanyan et al., 2023; Zhang et al., 2023; Lian et al., 2023a) indicate the effectiveness of scaling laws concerning data and model scale. The limit of scaling has not been reached yet. However, the scaling law for ASR is most effective for normal or perfect speech (p-ASR in Fig.4). In real-life settings, things are very different for imperfect speech, such as disfluent speech. Due to the power of language modeling in ASR systems, most imperfect speech is treated as perfect speech, leading to a significant performance drop for imperfect ASR (i-ASR in Fig.4). The overall ASR (o-ASR in Fig. 4), which includes both parts, should also follow the same trend. Lian et al. (2023c) introduced the *text refresher* to introduce imperfections for disfluent speech in an attempt to avoid the aforementioned problems. The

solutions are intuitive. Of all imperfections (disfluencies) at the word level, insertions and deletions are the hardest to detect. However, this can be easily observed on the 2D-Alignment introduced in the previous section. In the 2D-Alignment, we also have 2D-Alignment-DTW as a reference. If the 2D-Alignment does not align with any reference words, then it is likely an insertion, and if the word from the ASR system is redundant in comparison to the 2D-Alignment phoneme sequence, it is likely a deletion. It is important to note that URFA also generates word transcriptions. However, based on our findings, it exhibits inferior performance in word-level disfluency detection compared to the "text refresher." Therefore, we have chosen to employ URFA exclusively for phonetic-level disfluency detection

2.3 Transcription Module Evaluation

2.3.1 Duration-Aware Phonetic Transcription

We follow (Lian et al., 2023c) for phonetic transcription evaluation. Here, we provide more insights for each evaluation metric. First, the transcribed phonemes must be intelligible at the segment level, which is evaluated by the phoneme error rate (*PER*). Second, the transcribed phonemes must be intelligible at the frame-level, which is evaluated by frame-level **Micro F1 Score** and **Macro F1 Score** (*sklearn F1*). Third, the transcribed phonemes must be intelligible at both the segment and frame levels, which is evaluated by the combination of the above metrics. This is also known as *dPER* (Lian et al., 2023c). In more detail, *dPER* is the duration-aware extension of *PER*. For each operation to be counted, we consider the duration for it.

2.3.2 Duration-Aware Imperfect Word Transcription

Disfluent speech is imperfect speech. Traditional ASR systems are typically evaluated by how well the hypothesis matches the ground truth text. In disfluent settings, ASR systems are evaluated based on how well the hypothesis matches the imperfect targets. We start by following (Lian et al., 2023c) to adopt the imperfect word error rate (i-WER) where the disfluent (imperfect) targets are labeled by humans. In our proposed method, we also employ segment-level imperfect ASR evaluation, similar to *dPER* vs *PER*, where duration is also considered. In detail, 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 disfluency is identified as detected. We also report the F1 score for this matching evaluation, referred to as the **Matching Score (MS)**.

3 Detection Module

The reason we adopt a separate design for the detection and transcription modules is that an end-to-end modeling approach for the detection system is not reliable. The transcription module provides us with disfluency-aware representations to optimize the detection module. Here, we can still design learning-based methods (Harvill et al., 2022; Shonibare et al., 2022; Alharbi et al., 2017, 2020) to predict the detection results; however, we don't have human labels for disfluencies, which might be considered for future work. Instead, we have developed a smart label-free system that simply employs the template matching algorithm for each type of disfluency. Template matching is efficient and reliable, eliminating the need for human annotation. We have designed disfluency templates for both word and phoneme levels. These disfluencies include *Phonetic Errors (Missing, Deletion, Replacement)*, *Repetition*, and *Irregular Pause*. Our methods also cover word-level disfluencies, including *Missing*, *Insertion*, *Replacement*, and *Repetition*. The following section details them.

3.1 Phonetic-Level disfluency Detection

We follow the approach outlined in (Lian et al., 2023c) for designing disfluency templates. Instead of directly handling the alignment from dynamic alignment search, we also consider alignment data from the URFA module. We repeat the processes described in (Lian et al., 2023c). In Figure 1-Template, when examining alignments in normal speech, we observe perfect alignment between the two representations. However, closer examination reveals distinctive patterns within these alignments. If we notice a significant drop in alignment-2D-DTW without any overlap in the corresponding row, this signals the presence of a **missing** phoneme, as depicted in Fig 1-Template-(b). When a row in alignment-2D-DTW intersects with multiple columns in alignment-2D and contains repeated phonemes, it indicates a **repetition**, as illustrated in Figure 1-template-(d). Conversely, if a row in alignment-2D-DTW aligns with alignment-2D and simultaneously matches the surrounding column

in alignment-2D, this signifies an **insertion**, as exemplified in Figure 1-template(c). When a row in alignment-2D-DTW fails to overlap with any horizontal regions in alignment-2D but does overlap with a single vertical block in alignment-2D, it is categorized as a **replacement**, as demonstrated in Figure 1-template(e). Lastly, any pauses occurring within a complete sentence are recognized as **irregular pauses**, as shown in Figure 1-template(f).

3.2 Word-level disfluency Detection

We followed the same processes for detecting word-level disfluencies as we did for phoneme-level disfluencies. In line with Lian et al. (2023c), neither duration nor silence were taken into consideration. It's important to note that, unlike Lian et al. (2023c), we select the best results from either URFA or the *text refresher*. We adhere to the evaluation framework proposed by Lian et al. (2023c) for assessing hierarchical disfluency. To provide a more detailed evaluation, we utilize F1 scores and matching scores that consider temporal labels.

4 Experiments

4.1 Datasets and Pre-processing

VCTK (Yamagishi et al., 2019) We utilize VCTK for training the UFA module. We follow the train/test split methodology outlined in Lian et al. (2022d,b, 2023d, 2022c).

VCTK⁺⁺ (Lian et al., 2023c) It is a disfluency-aware simulated speech based on VCTK (Yamagishi et al., 2019). Three types of disfluencies are introduced: repetitions, prolongations, and blocks. For repetitions and prolongations, phonemes are randomly selected and prolonged or repeated for a random duration. These operations are performed in the temporal domain (waveform). VCTK⁺⁺ is utilized for training the UFA.

Buckeye (Pitt et al., 2005) It includes substantial segments of disfluent speech that have been meticulously annotated with precise time markings. To create our training and testing subsets, we adhere to the methodology outlined in (Lian et al., 2023c). Buckeye serves as our primary resource for both training the UFA module and conducting *Phonetic Transcription Evaluation*.

Disordered Speech We utilize the same corpus as Lian et al. (2023c). Collaborating with speech-language pathologists (SLPs), we personally annotate the hierarchical disfluencies. However, since

Method	WavLM Size	Training Data	Micro F1 (% , \uparrow)	Macro F1 (% , \uparrow)	dPER (% , \downarrow)	PER (% , \downarrow)	Micro F1 (% , \uparrow)	Macro F1 (% , \uparrow)	dPER (% , \downarrow)	PER (% , \downarrow)
			<i>Buckeye Test Set</i>				<i>VCTK++ Test Set</i>			
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

this segment consists of only 20 minutes of aphasia/dyslexia speech, it is exclusively employed for inference purposes. It’s important to note that, due to privacy considerations, this particular dataset will not be publicly shared. Please be aware that in the future, we will continue to collect more data from both hospitals and K-5 schools on a larger scale.

4.2 Phonetic Transcription Experiments

Lian et al. (2023c) conducted phonetic experiments on several tasks. First, two baselines were attempted. One is named WavLM-CTC-VAD, where VAD introduces silence into the WavLM-CTC alignment. The other is WavLM-CTC-MFA, where phoneme labels from WavLM-CTC(HugginFace-WavLM, 2022) are set as MFA (McAuliffe et al., 2017) targets. Results from Lian et al. (2023c) indicate that UFA outperforms the baselines under various settings (Buckeye test set and VCTK++ test set). In this work, we explore the role of monotonicity that was introduced. Specifically, we applied the CTC constraint to latent embeddings in the UFA module. An additional phoneme recognition module was applied to introduce such monotonicity. The intuition behind introducing this monotonicity is that the learned phonetic alignment still jumps up and down for disfluent speech and is unstable(Lian et al., 2023c). In this module, we only train UFA without any recursive learning, which will be introduced later on. 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 the Table.1.

URFA Config	iWER(% , \downarrow)			
	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

4.3 Imperfect Word Transcription Experiments

We present results from Whisper (Radford et al., 2023) and zero-order text refresher from (Lian et al., 2023c). In these settings, we conduct recursive word transcription modeling in multiple orders. The recursive process involves the following steps: The default UDM (Lian et al., 2023c) provides zero-order results. After the initial smoothed segmentation, we perform a 2D alignment search and further smoothed segmentation at the segment level. This yields 1st-order word segmentation and 1st-order word transcription. Additionally, we can use the 1st-order 2D-Alignment to guide the text refresher, which also provides us with 1st-order word transcription. We select the better of the two as the final 1st-order transcription, which is used as our final predictions. By repeating this process, we obtain 2nd-order word transcriptions, 3rd-order word transcriptions, and so on. For word segmentation evaluation, we utilize WhisperX (Bain et al., 2023), which provides timing information for each word. The results are detailed in Table 2 for word transcription evaluation and Table 3 for word segmentation evaluation. We also include disfluent speech segmentation results in the appendix A.

URFA Config	MS(% , \uparrow)			
	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 disfluency Detection

We select UFA-VCTK and UFA-VCTK++ as the default phoneme transcriber, as they exhibit the best phonetic transcription performance, as demonstrated in Table 1. In this study, we also aim to investigate whether the proposed recursive inference algorithm can enhance disfluency detection. It’s important to note that the representations used for disfluency detection are always based on the 2D-Alignment, but with different orders of computations, including 1st-order, 2nd-order, and 3rd-order. The results are presented in Table 4 and Table 5. MS refers to the "Matching Score," as explained in Section 2.3.2.

4.5 Results and Discussion

4.5.1 Transcription Analysis

In the phonetic results presented in Table 1, UFA with VCTK/VCTK++ consistently outperforms the other baseline settings. Therefore, we only introduce monotonicity (CTC) to UFA+VCTK/VCTK++. Ultimately, the inclusion of CTC significantly enhances performance across all metrics. Regarding word transcription results, as shown in Table 2, we observe two aspects. First, when examining the default setting (Lian et al., 2023c), which corresponds to the zero-order setting, we can see that CTC improves zero-order transcription results. Second, when we further explore recursive inference experiments, the results for the $(n + 1)$ th order are consistently better than those for the n th order. It’s worth noting that CTC, which introduces monotonicity, further boosts performance. We have not yet explored scaling results, and we are unsure if this could yield a better scaling curve as shown in Fig.4. We leave this as a topic for future work. We refrained from investigating 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 disfluent word boundaries surpasses that achieved in the zero-order iteration. This conclusion also holds true for disfluent word segmentation results, as reported in Table 3. Notably, our methods outperform those of Bain et al. (2023) by a significant margin. Furthermore, we provide more examples in Appendix A to illustrate its effectiveness.

URFA Settings	F1 (% , \uparrow)	MS (% , \uparrow)	Human F1 (% , \uparrow)	Human MS (% , \uparrow)
UFA-VCTK	62.4	55.2	90.4	85.6
UFA-VCTK++	64.5	60.2	90.6	86.0
+CTC	65.0	60.4	90.5	86.2
+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 disfluency Detection Evaluation

4.5.2 disfluency Analysis

We examine both phonetic-level and word-level dysfluencies in Table 4 and Table 5, respectively. It is evident that the introduction of CTC monotonicity consistently enhances performance at both levels. Additionally, when we consider recursive modeling, we can observe progressively improved performance as we increase the number of orders. However, it also reaches a point of saturation when we include further recursions.

Methods	F1 (% , \uparrow)	Human F1 (% , \uparrow)
Whisper-Large	64.0	86.4
+Text Refresher(VCTK)	66.8	88.0
+Text Refresher(VCTK++)	68.4	89.1
+CTC	68.8	89.2
+1st-order	70.1	89.1
+2nd-order	73.0	89.3
+3rd-order	73.1	89.3

Table 5: Word disfluency Detection Evaluation

5 Limitations

We propose a hierarchical unconstrained dysfluency modeling (H-UDM), which is an extension of UDM (Lian et al., 2023c). H-UDM introduces CTC monotonicity, and the incorporation of recursive modeling significantly enhances both transcription and disfluency detection results by a substantial margin. However, there are still several limitations to consider. First, the results on disordered speech are not as satisfactory. This suggests that the inference-only algorithm, the template matching algorithm, may not be sufficient for advanced disfluency modeling. It remains essential to develop end-to-end methods to address this challenge, which, however, presents its own set of difficulties. Second, the current closed-set disfluency classification only includes five types of disfluencies: replacement, insertion, repetition, block, and deletion. However, in an open-domain disfluency modeling system, there are many other complex disfluency patterns, such as syllable swapping and false starts. Designing specific templates

for each type of disfluency is impractical. Third, phoneme units may not be the optimal choice for modeling disfluency. For instance, there may not be significant acoustic differences between "AH" and "AA," yet H-UDM treats them as two distinct phonemes. Although this is partially alleviated by smoothed segmentation, the improvement is limited. Therefore, it is worth exploring alternative speech units, such as articulatory units (Lian et al., 2022a, 2023b; Wu et al., 2023b,a), to enhance alignment modeling.

6 Acknowledgement

Thanks for support from UC Noyce Initiative, Society of Hellman Fellows, NIH/NIDCD and the Schwab Innovation fund.

References

- 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*.
- 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.
- 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.
- HuggingFace-WavLM. 2022. Wavlm-ctc-huggingface. <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*.
- 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, Alexei Baevski, Wei-Ning Hsu, and Michael Auli. 2023a. Av-data2vec: Self-supervised learning of audio-visual speech representations with contextualized target representations. *arXiv preprint arXiv:2302.06419*.
- Jiachen Lian, Alan W Black, Louis Goldstein, and Gopala Krishna Anumanchipalli. 2022a. **Deep Neural Convolutional Matrix Factorization for Articulatory Representation Decomposition**. In *Proc. Interspeech 2022*, pages 4686–4690.
- Jiachen Lian, Alan W Black, Yijing Lu, Louis Goldstein, Shinji Watanabe, and Gopala K Anumanchipalli. 2023b. Articulatory representation learning via joint factor analysis and neural matrix factorization. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

- Jiachen Lian, Carly Feng, Naasir Farooqi, Steve Li, Anshul Kashyap, Cheol Jun Cho, Peter Wu, Robbie Netzorg, Tingle Li, and Gopala Krishna Anumanchipalli. 2023c. Unconstrained dysfluency modeling for dysfluent speech transcription and detection. *arXiv preprint arXiv:2312.12810*.
- Jiachen Lian, Chunlei Zhang, Gopala K. Anumanchipalli, and Dong Yu. 2023d. [Unsupervised tts acoustic modeling for tts with conditional disentangled sequential vae](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2548–2557.
- Jiachen Lian, Chunlei Zhang, Gopala Krishna Anumanchipalli, and Dong Yu. 2022b. [Towards Improved Zero-shot Voice Conversion with Conditional DSVAE](#). In *Proc. Interspeech 2022*, pages 2598–2602.
- Jiachen Lian, Chunlei Zhang, Gopala Krishna Anumanchipalli, and Dong Yu. 2022c. [Utts: Unsupervised tts with conditional disentangled sequential variational auto-encoder](#).
- Jiachen Lian, Chunlei Zhang, and Dong Yu. 2022d. Robust disentangled variational speech representation learning for zero-shot voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6572–6576. IEEE.
- Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kald. In *Interspeech*, volume 2017, pages 498–502.
- NIDCD. 2016. Nidcd. <https://www.nidcd.nih.gov/health/statistics/quick-statistics-voice-speech-language>.
- OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madeleine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret

- Zoph. 2023. [Gpt-4 technical report](#).
- 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/>
- VCL. 2021. Vcl. <https://vclenglish.com/54-8-billion-by-2025->
- Peter Wu, Li-Wei Chen, Cheol Jun Cho, Shinji Watanabe, Louis Goldstein, Alan W Black, and Gopala K. Anumanchipalli. 2023a. [Speaker-independent acoustic-to-articulatory speech inversion](#). In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Peter Wu, Tingle Li, Yijing Lu, Yubin Zhang, Jiachen Lian, Alan W Black, Louis Goldstein, Shinji Watanabe, and Gopala K. Anumanchipalli. 2023b. [Deep Speech Synthesis from MRI-Based Articulatory Representations](#). In *Proc. INTERSPEECH 2023*, pages 5132–5136.
- Junichi Yamagishi, Christophe Veaux, Kirsten MacDonal, 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

Human Data Annotation For all disordered speech (aphasia and dyslexia), 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.

Word Segmentation Examples

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

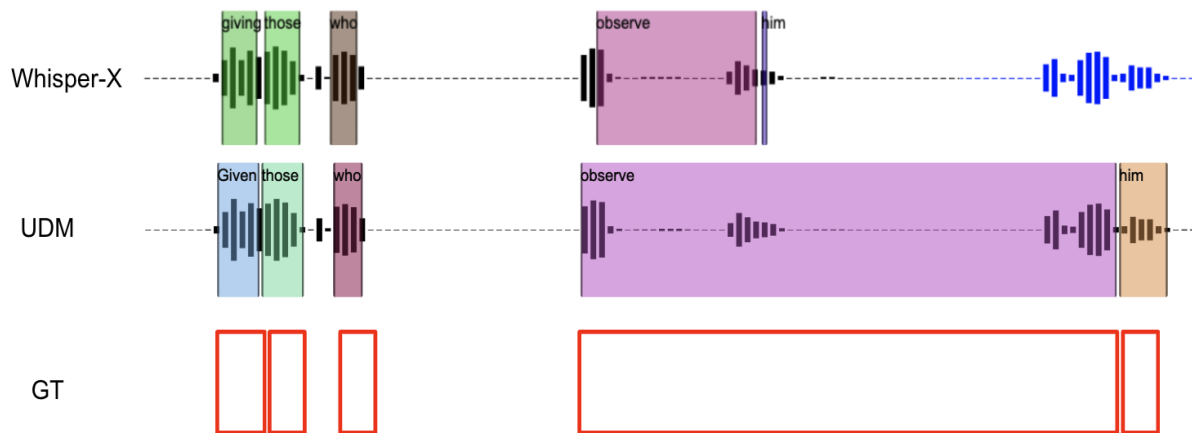


Figure 5: Segmentation-(Dyslexia Sample: Giving those who observe him)



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

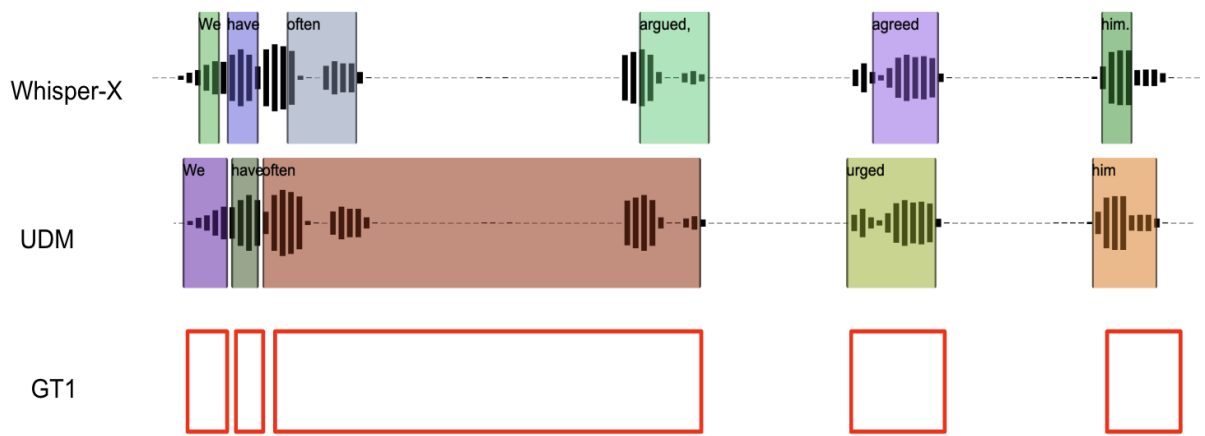


Figure 7: Segmentation-(Dyslexia Sample: We have often urged him)



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

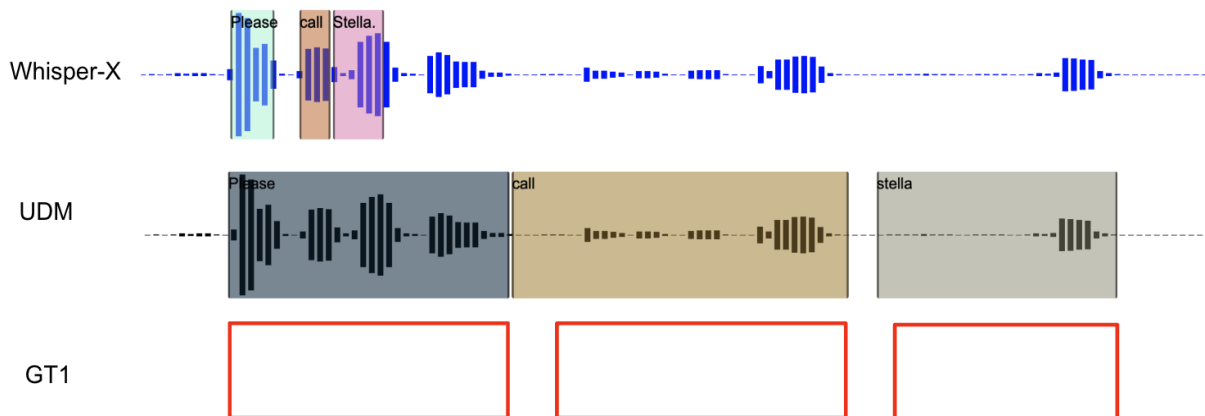


Figure 9: Segmentation-(My stutter sample: Please call stella.)