Interspeech 2024
1-5 September 2024, Kos, Greece

Large Language Models for Dysfluency Detection in Stuttered Speech

Dominik Wagner[1], *Sebastian P. Bayerl*[1], *Ilja Baumann*[1], *Korbinian Riedhammer*[1], *Elmar Nöth*[2], *Tobias Bocklet*[1,3]

[1]Technische Hochschule Nürnberg Georg Simon Ohm
[2]Friedrich-Alexander-Universität Erlangen-Nürnberg
[3]Intel Labs

dominik.wagner@th-nuernberg.de

Abstract

Accurately detecting dysfluencies in spoken language can help to improve the performance of automatic speech and language processing components and support the development of more inclusive speech and language technologies. Inspired by the recent trend towards the deployment of large language models (LLMs) as universal learners and processors of non-lexical inputs, such as audio and video, we approach the task of multi-label dysfluency detection as a language modeling problem. We present hypotheses candidates generated with an automatic speech recognition system and acoustic representations extracted from an audio encoder model to an LLM, and finetune the system to predict dysfluency labels on three datasets containing English and German stuttered speech. The experimental results show that our system effectively combines acoustic and lexical information and achieves competitive results on the multi-label stuttering detection task.

Index Terms: dysfluency detection, stuttering, large language models, wav2vec 2.0, Whisper, pathological speech

1. Introduction

Stuttering is a diverse neurodevelopmental condition that affects an individual's verbal communication ability [1] and can negatively impact the performance of automatic speech and language processing systems [2, 3, 4]. The symptoms of stuttering vary strongly among individuals and depend on psychological influences, conversational factors, and the linguistic complexity of an utterance [1]. The accurate detection of dysfluent speech has implications for stuttering therapy, e.g., in self-therapy software, monitoring of stuttering by clinicians, as well as the development of more inclusive speech technologies in general. Dysfluency detection has traditionally been focused on acoustic features, since the various dysfluency types are often directly visible in audio waveforms and their corresponding spectrograms [5, 6]. Most studies employ spectral features like MFCCs or coefficients derived from linear predictive coding methods [7, 8]. Recent work has explored neural representations obtained from acoustic encoder models [9, 10, 11, 12], in particular wav2vec 2.0 [13]. These representations are then used as input features to various classifiers ranging from support vector machines to neural networks.

Lexical features such as transcriptions generated by automatic speech recognition (ASR) systems are not extensively studied in stuttering-related dysfluency detection, particularly in combination with acoustic features. However, they are commonly used to detect and remove dysfluencies in transcriptions obtained from typical speech [14, 15]. Other studies use different feature types depending on the dysfluency class. For example, in [16], prolongations are detected based on the correlation between successive audio frames and other stuttering-related dysfluencies are detected using word lattices.

While most studies on dysfluency detection focus on using either lexical or acoustic features, it is well known that combinations of the two can improve the detection of speaking style differences in spoken dialog systems [17] and in distinguishing machine-directed speech from background speech [18].

Large language models (LLMs) such as Llama 2 [19], Falcon [20] and GPT-4 [21] exhibit strong language understanding and generation abilities. More importantly, new capabilities have emerged that are not present in smaller-scale language models (e.g., multi-step reasoning, in-context learning, and instruction following) [22]. Furthermore, the potential of LLMs is not limited to text data alone. Recent studies have demonstrated their capability to process and understand non-lexical information such as audio, video, and images [23, 24, 25, 26, 27], bridging the gap between different modalities.

This work is inspired by the recent successful attempts to enable learning from non-lexical features in LLMs and their ability to adapt to new tasks. We explore a system to detect stuttering-related dysfluencies using three datasets of English and German stuttered speech. Our goal is to correctly identify all dysfluency labels in the data (i.e., blocks, interjections, prolongations, sound repetitions, and word repetitions), as well as modified speech in short time intervals. Besides the lexical inputs obtained through ASR transcriptions (either orthographic [28] or phonetic [29]), we employ latent features generated with an acoustic encoder based on wav2vec 2.0. The ASR hypotheses candidates and acoustic features are concatenated and jointly presented to a pretrained Llama 2 model, enabling it to consider both acoustic and lexical content. The system is optimized using low-rank-adaption (LoRA) [30] to generate dysfluency labels based on the acoustic and lexical context. Lexical features represent the characteristics of dysfluent speech at resolutions distinct from those of acoustic features. Acoustic features are fine-grained and able to capture occurrences of dysfluencies in narrow time frames, whereas lexical features are coarse and focus on self-contained entities (e.g. words). Therefore, we hypothesize that lexical and acoustic features are complementary, each offering distinct advantages in enhancing dysfluency detection.

2. Data

For our experiments, we employ three datasets consisting of audio clips lasting 3 seconds each, annotated with stuttering-related dysfluencies; SEP-28k-Extended, FluencyBank, and the Kassel State of Fluency (KSoF) dataset [11, 7, 31]. The datasets have similar labels with clips containing either no dysfluencies or one or more types of stuttering-related dysfluencies; blocks, prolongations, sound repetitions, word repetitions, and interjections. The clips can be labeled with more than one type of dysfluency, making it a multi-label classification problem.

10.21437/Interspeech.2024-2120

The largest dataset, SEP-28k-E, comprises approximately ∼28k English audio clips sourced from podcasts discussing stuttering. Derived from the SEP-28k dataset, it features semi-automatically generated speaker labels and a speaker-exclusive Train-Dev-Test partition.[1] The original SEP-28k release contains a subset of the adults who stutter dataset of the FluencyBank corpus [32] segmented into 4144 English clips that were annotated to match the annotations used in the larger dataset. In our experiments, we employ the partition outlined in [10].[2]

The KSoF dataset consists of 5597 audio segments extracted from German stuttering therapy recordings. In addition to the five dysfluency labels in SEP-28k-E and FluencyBank, the clips can also contain annotated speech modifications, indicating that a person uses fluency shaping. Fluency shaping is a technique persons who stutter learn in stuttering therapy to help them overcome their stuttering [31]. A detailed description of the datasets and the distribution of the labels can be found in [7, 31, 11].

3. Method

3.1. wav2vec 2.0

Wav2vec 2.0 [13] describes a series of models consisting of a convolutional feature encoder $\mathcal{G} : \boldsymbol{X}_{1:T} \mapsto \boldsymbol{Z}_{1:L}$ with multiple identical blocks using temporal convolution, layer normalization, and GELU nonlinearity. The feature encoder maps a raw audio waveform $\boldsymbol{X}_{1:T} = \{x_1, \ldots, x_T\}$ of length T to hidden representations $\boldsymbol{Z}_{1:L} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L\}$ of length L. These hidden representations $\boldsymbol{Z}$ are then passed to a transformer [33] $\mathcal{T} : \boldsymbol{Z}_{1:L} \mapsto \boldsymbol{C}_{1:L}$, which generates context representations $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L$. During pretraining, a quantization component is used to quantize $\boldsymbol{Z}_{1:L}$, which serve as the targets in a contrastive learning task that requires classifying the true quantized representation within a set of distractors. Latent representations extracted at different transformer layers of wav2vec 2.0 are known to be reliable acoustic features for dysfluency detection [10, 12] and other related tasks such as mispronunciation detection [34]. Pretrained wav2vec 2.0 models are available in two sizes (94M and 315M parameters).

3.2. Whisper

Whisper [28] is a family of sequence-to-sequence transformer [33] models trained to perform multiple tasks such as multilingual ASR, language identification, and speech translation. The models are pretrained on ∼680k hours of data retrieved from the world wide web and are available in five sizes between 39M parameters and 1.55B parameters. All models within the Whisper family share a similar encoder-decoder structure, differing only in parameters such as the number of transformer blocks and hidden layer dimensions.

The Whisper encoder $\mathcal{E}$ maps J log-Mel spectrogram features obtained from the raw audio waveform $\hat{\boldsymbol{f}}(\boldsymbol{X})_{1:J} = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_J\}$ to a sequence of K hidden representations $\boldsymbol{H}_{1:K}$:

$$\mathcal{E} : \hat{\boldsymbol{f}}(\boldsymbol{X})_{1:J} \mapsto \boldsymbol{H}_{1:K}.$$

The decoder autoregressively predicts the probabilities for the next token y_i, given the previous tokens $\boldsymbol{y}_{<i}$ and the hidden representations $\boldsymbol{H}_{1:K}$: $p(y_i \mid \boldsymbol{y}_{<i}, \boldsymbol{H}_{1:K})$. The model is trained on pairs of input spectrograms and target transcriptions using the cross-entropy objective. Due to the unavailability of ground truth transcriptions for all datasets in this study, adapting an ASR system to the domain is not straightforward. Therefore, we rely on Whisper for its strong multilingual performance on standard benchmarks, without the need for task-specific fine-tuning on downstream tasks [28].

3.3. Low-rank Adaptation

In low-rank adaptation (LoRA) [30], the pretrained weights of the underlying model are frozen and small trainable adapters are optimized instead. LoRA adapters utilize low-rank decomposition matrices $\boldsymbol{A} \in \mathbb{R}^{r \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times r}$ to define an incremental update $\boldsymbol{\Delta} \in \mathbb{R}^{m \times n}$, which are applied to the pretrained weight matrices of the underlying model. The updates to a weight matrix $\boldsymbol{W}$ are expressed as:

$$\boldsymbol{W}^{'} = \boldsymbol{W} + \boldsymbol{\Delta} = \boldsymbol{W} + \boldsymbol{B}\boldsymbol{A},$$

where the rank $r \ll \{n, m\}$, is a hyperparamter determining the dimensionality of the low-rank decomposition matrices, and thus, the quantity of trainable parameters within the module.

3.4. Minimum Bayes Risk Decoding

Minimum Bayes risk (MBR) decoding has long been used in ASR [35, 36, 37] and machine translation (MT) [38, 39] as a means to improve accuracy of transcriptions or translations by considering not only the most likely hypothesis, but also the potential utility (or risk) associated with alternative hypotheses.

ASR systems typically utilize maximum a posteriori probability (MAP) decoding [40], which maximizes the probability of selecting the correct word sequence [35]. Since exhaustive MAP decoding is intractable, beam search is employed as an approximation instead [41]. However, MAP decoding exhibits several shortcomings that have been observed in both ASR and MT applications, such as a bias towards incomplete transcriptions [42], degrading performance with longer sequences [43] and hallucinations due to low robustness under domain shift [44]. We employ MBR decoding not only in the hope to mitigate some of the shortcomings of MAP decoding, but also to add more lexical diversity, by presenting a sequence of multiple hypothesis candidates to the LLM.

Let $\mathcal{U}(\boldsymbol{y'}, \boldsymbol{y})$ be a utility function that compares an hypothesis string $\boldsymbol{y'} \in \boldsymbol{\mathcal{Y}}$ against a reference string $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$ from the space of all possible hypotheses $\boldsymbol{\mathcal{Y}}$. The optimal decision $\boldsymbol{y}_{opt} \in \boldsymbol{\mathcal{Y}}$ is the one that maximizes the expected utility (or minimizes the expected risk) for data generated under the model $p(\boldsymbol{y} \mid \boldsymbol{x}, \theta)$ [45]:

$$\boldsymbol{y}_{opt} = \underset{\boldsymbol{y'} \in \boldsymbol{\mathcal{Y}}(\boldsymbol{x})}{\operatorname{argmax}} \mathbb{E}_{p(\boldsymbol{y} \mid \boldsymbol{x}, \theta)}[\mathcal{U}(\boldsymbol{y'}, \boldsymbol{y})].$$

Similar to MAP decoding, calculating the expectation over the entire space of all potential hypotheses $\boldsymbol{\mathcal{Y}}$ is often impractical, thus requiring the use of a tractable subset $\hat{\boldsymbol{\mathcal{Y}}} \subset \boldsymbol{\mathcal{Y}}$ of the full hypothesis space. Previous works use n-best lists from beam search to compute a biased estimate of expected utility [37, 36, 39, 46]. More recently, unbiased estimates have been obtained via Monte Carlo (MC) sampling [45, 47, 48].

In MC sampling-based approaches, the set of possible hypotheses is approximated by drawing S independent samples from the model via ancestral sampling. For an hypothesis $\boldsymbol{y'}$, the expectation is then maximized over $\hat{\boldsymbol{\mathcal{Y}}}$:

$$\boldsymbol{y}_{MC} = \underset{\boldsymbol{y'} \in \hat{\boldsymbol{\mathcal{Y}}}(\boldsymbol{x})}{\operatorname{argmax}} \frac{1}{S} \sum_{s=1}^{S} \mathcal{U}(\boldsymbol{y'}, \boldsymbol{y}_s) \text{ with } \boldsymbol{y}_s \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \theta).$$

We employ the MBR method proposed in [45] with $S = 10$ and the negated word error rate for $\mathcal{U}$ to conduct our experiments.

3.5. Phonetic Transcriptions

Dysfluencies such as sound repetitions may not be visible in orthographic transcriptions and limit the benefit of lexical features based on word-level ASR systems. To analyze the effectiveness

[1] Online: `https://tinyurl.com/yck9fmfv`

[2] Online: `https://tinyurl.com/24vm6dec`

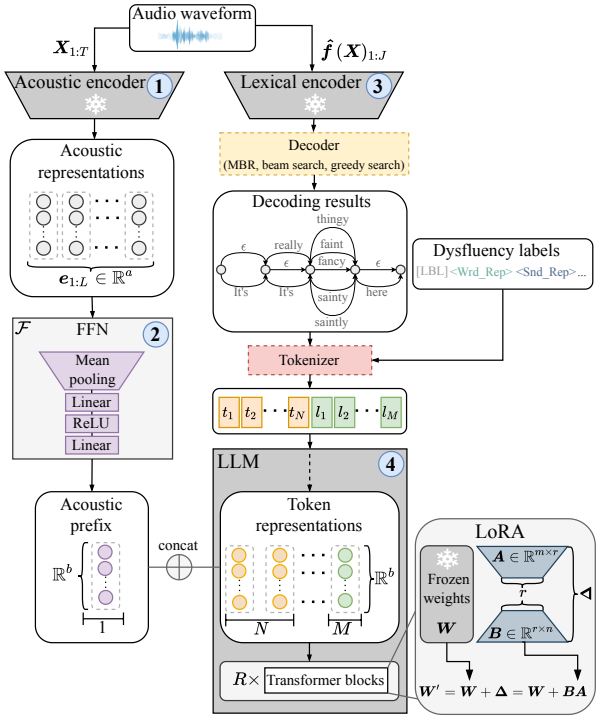

Figure 1: *Overview of the components in our dysfluency detection system. The left segments illustrate the extraction and processing of acoustic features. The right segments demonstrate the retrieval and processing of lexical features, as well as the feature combination within the LLM backbone.*

of more fine-grained units, we explore the phone recognition system described in [29] and predict phonetic transcriptions as an alternative to the orthographic transcriptions generated by Whisper.[3] The phone recognition system is based on the 315M parameter version of wav2vec 2.0 pretrained on data from 53 languages [49] and finetuned on multilingual Common Voice data [50]. It operates on symbols from the International Phonetic Alphabet as modeling units. The phonetic transcriptions are generated via greedy search and treated in the same way as the orthographic transcriptions, i.e., they are mapped to token identifiers and passed to the LLM (cf. Section 3.6).

3.6. Our Approach

The overall composition of our system is inspired by [23, 18, 25] and has four core components (cf. Figure 1). The first component (1) is an acoustic encoder, which generates latent representations from the input utterance. The model yields L acoustic representations $\boldsymbol{e}_{1:L} \in \mathbb{R}^{L \times a}$ for each raw audio waveform input $\boldsymbol{X}_{1:T}$.

The second main component (2) is a small feedforward network (FFN) $\mathcal{F} : \mathbb{R}^{L \times a} \mapsto \mathbb{R}^b$, consisting of mean pooling, two linear layers and ReLU activation. The FFN aggregates the extracted acoustic information along the time dimension and projects the result into the latent token space of a causal LLM. The acoustic projection $\mathcal{F}(\boldsymbol{e}_{1:L})$ is concatenated with the token representations of the LLM. The combined representations are then passed through each of its R Transformer blocks.

The third main component (3) is a pretrained ASR system, which is used to extract lexical features from the audio input by means of three different decoding methods (greedy search, beam search, and MBR decoding). The decoding algorithm generates viable hypothesis candidates, which are then tokenized and passed as inputs to the LLM. Beam search and MBR decoding yield multiple hypotheses, which are flattened to a string of words (i.e., all candidates are concatenated beginning with the most likely one).

Acoustic and lexical features are combined in the fourth main component (4), a pretrained causal LLM. The LLM generates text tokens conditioned on the acoustic representations transformed via $\mathcal{F}$ and the ASR hypotheses candidates. The weights of the LLM remain frozen during training and LoRA modules are optimized instead. During training, the weights of the LLM are held constant, while optimization focuses on the LoRA modules. Components (2) and (4) undergo joint optimization and components (1) and (3) remain unchanged throughout the training process.

For a training set containing D acoustic representations, tokenized ASR hypotheses, and dysfluency label tokens $\{(\boldsymbol{e}^{(d)}_{1:L}, \boldsymbol{t}^{(d)}_{1:N}, \boldsymbol{l}^{(d)}_{1:M})\}_{d=1}^{D}$, the training objective for the parameters θ is the autoregressive prediction of the next dysfluency token l_i, given the previous dysfluency tokens $\boldsymbol{l}_{<i}$, the acoustic representations $\boldsymbol{e}_{1:L}$ and the ASR hypothesis tokens $\boldsymbol{t}_{1:N}$ using cross-entropy loss:

$$\mathcal{L}_\theta = -\sum_{i=1}^{M} p_\theta \left(l_i \mid \boldsymbol{l}_{<i}, \boldsymbol{e}_{1:L}, \boldsymbol{t}_{1:N} \right).$$

At inference time, we use greedy search to generate a token sequence for a predefined maximum number of steps or until the `[EOS]` token is encountered. Dysfluency labels are then extracted from the generated text string. Any incomplete or inaccurately generated labels are discarded. We use a maximum number of 20 steps in our experiments.

3.7. Modeling and Architecture Details

Each system is trained until convergence with an effective batch size of 32. Early stopping is applied, when the loss on the development set is not improving for five consecutive epochs. Each model used in Section 4 is selected based on the lowest loss achieved on the development set. For optimization, we use AdamW [51] ($\lambda = 10^{-4}$, $\epsilon = 10^{-8}$, $\beta_1 = 0.99$, $\beta_2 = 0.999$) with an initial learning rate of 2×10^{-4}, a linear schedule and a warm-up phase of 5% of total training steps. The text token sequences are padded to a length of 1024. Token sequences longer than 1024 are truncated. Truncation occurred only in very few cases ($< 1\%$ of the training data). The hidden layers of $\mathcal{F}$ employ 512 units and are trained with a dropout probability of 10%. We set $r = 64$ and the scaling factor for adjusting the magnitude of the adaption $\alpha = 16$ in all our experiments. The LoRA modules are optimized with a dropout probability of 10%. We use the 7B parameter version of Llama 2 as the LLM backbone and a pretrained 315M parameter wav2vec 2.0 model as the acoustic encoder. The experiments on SEP-28k-E and FluencyBank employ an acoustic encoder pretrained on the XLSR-53 [49] dataset, whereas the experiments on KSoF use a model pretrained on XLSR-53 that was subsequently finetuned on the German portion of Common Voice. The lexical encoder is either a pretrained 1.55B parameter Whisper [28] model or the phone recognition system based on wav2vec 2.0 [29].

4. Experiments

4.1. Experimental Setup

We extracted acoustic representations from domain-adapted wav2vec 2.0 models using the method described in [12], as well as their corresponding vanilla (i.e., out-of-domain) equivalents. The acoustic representations were extracted at the 12$^{\text{th}}$ and 24$^{\text{th}}$

[3] We intentionally avoid the widely used but imprecise terms "phonemic transcription" and "phoneme recognizer" here.

Table 1: *Multilabel dysfluency detection results using multilingual acoustic representations. Mod = Modified speech, Blk = Block, Int = Interjection, Pro = Prolongation, Snd = Sound repetition, Wrd = Word repetition. "Finetuned" indicates whether the acoustic features were domain-adapted or not. "Layer" refers to the wav2vec 2.0 transformer layer used for acoustic feature extraction. The column "ASR Decoder" shows the various types of decoding algorithms used to obtain ASR transcriptions.*

Exp.	Acoustic Features		ASR	SEP-28k-E					FluencyBank					KSoF					
	Finetuned	Layer	Decoder	Blk	Int	Pro	Snd	Wrd	Blk	Int	Pro	Snd	Wrd	Mod	Blk	Int	Pro	Snd	Wrd
Baseline (experiments #22-24 in [12])				*0.32*	***0.77***	*0.53*	*0.53*	***0.64***	*0.36*	*0.79*	***0.62***	*0.64*	***0.52***	*0.75*	***0.64***	***0.85***	***0.60***	*0.48*	***0.14***
1	✗	12	1-best	**0.62**	0.62	0.51	0.45	0.31	0.47	0.66	0.49	0.47	0.40	0.74	0.50	0.49	0.34	0.38	0.03
2			N-best	0.61	0.60	0.51	0.44	0.31	0.40	0.65	0.46	0.47	0.36	0.76	0.51	0.45	0.27	0.34	0.04
3			Phon	0.60	0.58	0.50	0.42	0.30	0.53	0.69	0.41	0.50	0.42	0.77	0.47	0.49	0.21	0.29	0.04
4			MBR	0.61	0.62	0.52	0.40	0.31	0.47	0.63	0.46	0.47	0.37	0.78	0.48	0.50	0.18	0.26	0.06
5	✗	24	1-best	0.60	0.52	0.46	0.12	0.13	0.43	0.65	0.28	0.35	0.31	0.70	0.51	0.27	0.15	0.30	0.04
6			N-best	0.59	0.51	0.47	0.16	0.06	0.43	0.59	0.28	0.33	0.30	0.68	0.46	0.31	0.18	0.28	0.00
7			Phon	0.59	0.52	0.43	0.21	0.10	0.49	0.61	0.34	0.39	0.15	0.70	0.46	0.28	0.16	0.30	0.01
8			MBR	0.58	0.52	0.43	0.21	0.05	0.42	0.62	0.26	0.33	0.29	0.69	0.47	0.32	0.10	0.28	0.01
9	✓	12	1-best	0.59	0.59	0.50	0.41	0.31	0.47	0.66	0.35	0.45	0.33	0.73	0.37	0.44	0.30	0.32	0.03
10			N-best	0.61	0.57	0.50	0.40	0.29	0.48	0.64	0.39	0.44	0.33	0.70	0.44	0.39	0.20	0.27	0.00
11			Phon	0.60	0.57	0.51	0.40	0.27	0.52	0.64	0.38	0.46	0.39	0.73	0.45	0.44	0.29	0.31	0.06
12			MBR	0.60	0.58	0.50	0.39	0.29	0.52	0.66	0.42	0.46	0.36	0.72	0.43	0.35	0.20	0.32	0.01
13	✓	24	1-best	0.57	0.74	**0.56**	**0.54**	**0.64**	0.57	**0.81**	0.55	**0.66**	0.43	0.77	0.59	**0.85**	0.52	0.48	0.11
14			N-best	0.57	0.73	**0.56**	0.53	0.61	0.56	0.80	0.56	0.61	**0.52**	0.77	0.55	0.84	0.49	0.43	0.04
15			Phon	0.56	0.72	0.54	0.48	0.58	0.56	0.79	0.52	0.63	0.45	**0.79**	0.56	0.82	0.48	**0.50**	0.08
16			MBR	0.58	0.73	**0.56**	0.49	0.56	**0.58**	0.78	0.48	0.63	0.44	0.78	0.57	0.82	0.52	0.49	0.12

transformer layer of the wav2vec 2.0 system. Lexical features were either, phone-level (*Phon*) or word-level 1-best hypotheses (*1-best*) generated via greedy search, n-best lists generated via beam search (*N-best*), or hypothesis candidates obtained via MBR decoding (*MBR*). Beam search was configured with a beam width 12 and the number of hypotheses in each n-best list was 10. The baseline systems are the multilingual finetuned models from [12]. We report F1-scores balanced between precision and recall for all dysfluency types, as well as modified speech, which is only available for the KSoF corpus.

Preliminary experiments involved utilizing either solely acoustic or solely lexical features for the dysfluency detection task. Lexical features alone underperformed compared to the use of acoustic-only features, and either standalone approach was outperformed by the combination of both acoustic and lexical features. Furthermore, we conducted experiments substituting Llama 2 with Falcon 7B [20] and Mistral 7B [52], but no substantial performance differences were observed. We also examined different methods of aggregating acoustic feature sequences. However, using mean pooling to reduce the sequence length to one and passing a single aggregated acoustic vector as the prefix to the LLM performed comparably to using various aggregation levels that generate longer sequences.

4.2. Results and Discussion

The results are summarized in Table 1. Our LLM-based approach either matched or surpassed the baseline for most dysfluency types, when domain-adapted acoustic features extracted at the last layer of the finetuned wav2vec 2.0 system were used (cf. exp. #13-16). The least overall improvement was observed on the German KSoF corpus, where our system using phonetic transcriptions only marginally surpassed the baseline for modified speech and sound repetitions (cf. exp. #15). This discrepancy may be attributed to the predominance of English pretraining data for the LLM backbone, which may have more difficulty capturing the nuances of the German language.

Our systems exhibit strong performance in identifying blocks within both the SEP-28k-E and FluencyBank datasets with maximum relative improvements over the baseline of $\sim$94% and $\sim$61%, respectively. These improvements are consistent across all different types of acoustic features used. However, the improvement diminished on KSoF (at most $\sim$5% relative), where the baseline performance for blocks is considerably stronger with an F1-score of 0.75.

Generally, n-best lists and MBR hypotheses candidates did not yield significant enhancements over 1-best hypotheses. Only on the FluencyBank dataset, n-best lists exhibited considerably better performance for word repetitions, surpassing phonetic transcriptions by $\sim$15% relative (cf. exp. #14). We initially anticipated that the LLM would benefit from more lexical alternatives and consider their variations. However, it appears that the relevant information extractable from a text sequence is already captured by 1-best hypotheses in most cases.

Our approach did not lead to improvements on word repetitions. We hypothesize that Whisper's weakly supervised pretraining method may not have been the ideal option for modeling word repetitions in the lexical domain, as it may prioritize preserving the overall meaning of utterances, potentially leading to omissions of tokens with minimal additional information rather than generating exact word-for-word transcriptions.

Employing non-finetuned acoustic features from the last layer of wav2vec 2.0 yielded worse F1-scores compared to using the 12^{th} layer, regardless of the dataset (cf. exp. #1-4 and #5-8). This aligns with previous studies [9, 10], which indicate that transformer layers in the middle are more adept at capturing stuttering patterns. However, acoustic features from domain-adapted models yielded the best performance when extracted at the 24^{th} layer (cf. exp. #13-16). As the model is finetuned, the layers closer to the output become more specialized to the specific task. Finetuning adjusts the parameters of these layers to better fit the characteristics of the dataset, yielding representations better suited to detecting stuttering patterns.

5. Conclusions

Inspired by the recent advancements in LLMs, we approached multi-label dysfluency detection as a language modeling problem. Our system jointly learns from a combination of acoustic and lexical features. Experimental results demonstrated that the Llama 2 backbone effectively combines acoustic and lexical information, matching or outperforming a robust baseline on the majority of dysfluency types across three datasets of English and German stuttered speech. We found that domain-adapted acoustic features from the last layer of a wav2vec 2.0 system yielded the best performance, particularly when combined with 1-best ASR hypotheses generated via greedy decoding. Future work will explore lexical encoder alternatives, the impact of varying LLM sizes and full end-to-end finetuning of both the LLM backbone and the acoustic encoder.

6. References

[1] O. Bloodstein, N. B. Ratner, and S. B. Brundage, *A handbook on stuttering, 7th ed.*, ser. A handbook on stuttering, 7th ed., 2021.

[2] V. Mitra *et al.*, "Analysis and Tuning of a Voice Assistant System for Dysfluent Speech," in *Proc. Interspeech*, 2021.

[3] S. Wu, "The world is designed for fluent people: Benefits and challenges of videoconferencing technologies for people who stutter," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

[4] C. Lea *et al.*, "From user perceptions to technical improvement: Enabling people who stutter to better use speech recognition," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

[5] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Machine learning for stuttering identification: Review, challenges and future directions," *Neurocomputing*, vol. 514, pp. 385–402, 2022.

[6] L. Barrett, J. Hu, and P. Howell, "Systematic review of machine learning approaches for detecting developmental stuttering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1160–1172, 2022.

[7] C. Lea, V. Mitra, A. Joshi, S. Kajarekar, and J. P. Bigham, "Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter," in *ICASSP*, 2021, pp. 6798–6802.

[8] M. Jouaiti and K. Dautenhahn, "Dysfluency classification in stuttered speech using deep learning for real-time applications," in *ICASSP*, 2022, pp. 6482–6486.

[9] C. Montacié, M.-J. Caraty, and N. Lackovic, "Audio features from the wav2vec 2.0 embeddings for the acm multimedia 2022 stuttering challenge," in *Proc. ACM International Conference on Multimedia*, 2022, p. 7195–7199.

[10] S. P. Bayerl, D. Wagner, E. Nöth, and K. Riedhammer, "Detecting Dysfluencies in Stuttering Therapy Using wav2vec 2.0," in *Proc. Interspeech*, 2022.

[11] S. P. Bayerl, D. Wagner, E. Nöth, T. Bocklet, and K. Riedhammer, "The influence of dataset partitioning on dysfluency detection systems," in *Text, Speech, and Dialogue*, 2022, pp. 423–436.

[12] S. P. Bayerl, D. Wagner, I. Baumann, F. Hönig, T. Bocklet, E. Nöth, and K. Riedhammer, "A Stutter Seldom Comes Alone – Cross-Corpus Stuttering Detection as a Multi-label Problem," in *Proc. Interspeech*, 2023.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020, pp. 12 449–12 460.

[14] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency Detection Using a Bidirectional LSTM," in *Proc. Interspeech*, 2016.

[15] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP*, 2020, pp. 8069–8073.

[16] S. Alharbi, M. Hasan, A. J. H. Simons, S. Brumfitt, and P. Green, "A Lightly Supervised Approach to Detect Stuttering in Children's Speech," in *Proc. Interspeech*, 2018.

[17] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and L. Heck, "Learning when to listen: detecting system-addressed speech in human-human-computer dialog," in *Proc. Interspeech*, 2012.

[18] D. Wagner, A. Churchill, S. Sigtia, P. Georgiou, M. Mirsamadi, A. Mishra, and E. Marchi, "A multimodal approach to device-directed speech detection with large language models," in *ICASSP*, 2024, pp. 10 451–10 455.

[19] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023, arXiv:2307.09288.

[20] E. Almazrouei *et al.*, "The falcon series of open language models," 2023, arXiv:2311.16867.

[21] J. Achiam *et al.*, "GPT-4 technical report," 2023, arXiv:2303.08774.

[22] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024, arXiv:2402.06196.

[23] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," in *NeurIPS*, 2023.

[24] C. Tang *et al.*, "SALMONN: Towards generic hearing abilities for large language models," in *ICLR*, 2024.

[25] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. R. Glass, "Listen, think, and understand," in *ICLR*, 2024.

[26] C. Chen, R. Li, Y. Hu, C.-H. H. Yang, S. M. Siniscalchi, P.-Y. Chen, and E. Chng, "It's never too late: Fusing acoustic information into large language models for automatic speech recognition," in *ICLR*, 2024.

[27] P. K. Rubenstein *et al.*, "Audiopalm: A large language model that can speak and listen," 2023, arXiv:2306.12925.

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," 2022, arXiv:2212.04356.

[29] Q. Xu, A. Baevski, and M. Auli, "Simple and Effective Zero-shot Cross-lingual Phoneme Recognition," in *Proc. Interspeech*, 2022.

[30] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," in *ICLR*, 2022.

[31] S. Bayerl, A. Wolff von Gudenberg, F. Hönig, E. Nöth, and K. Riedhammer, "KSoF: The Kassel State of Fluency Dataset – A Therapy Centered Dataset of Stuttering," in *Proc. LREC*, 2022.

[32] N. Bernstein Ratner and B. MacWhinney, "Fluency Bank: A new resource for fluency research and practice," *Journal of Fluency Disorders*, vol. 56, pp. 69–80, 2018.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *NeurIPS*, 2017.

[34] X. Xu, Y. Kang, S. Cao, B. Lin, and L. Ma, "Explore wav2vec 2.0 for Mispronunciation Detection," in *Proc. Interspeech*, 2021.

[35] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in n-best list rescoring," in *Proc. Eurospeech*, 1997.

[36] V. Goel, W. Byrne, and S. Khudanpur, "LVCSR rescoring with modified loss functions: a decision theoretic perspective," in *ICASSP*, 1998, pp. 425–428.

[37] V. Goel and W. J. Byrne, "Minimum bayes-risk automatic speech recognition," *Computer Speech and Language*, vol. 14, no. 2, pp. 115–135, 2000.

[38] S. Kumar and W. Byrne, "Minimum bayes-risk word alignments of bilingual texts," in *Proc. EMNLP*, 2002, p. 140–147.

[39] ——, "Minimum Bayes-risk decoding for statistical machine translation," in *Proc. HLT-NAACL*, 2004, pp. 169–176.

[40] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 179–190, 1983.

[41] X. L. Aubert, "An overview of decoding techniques for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 89–114, 2002.

[42] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," 2016, arXiv:1612.02695.

[43] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. SSST-8*, 2014, pp. 103–111.

[44] M. Müller, A. Rios, and R. Sennrich, "Domain robustness in neural machine translation," in *Proc. Conference of the Association for Machine Translation in the Americas*, 2020, pp. 151–164.

[45] B. Eikema and W. Aziz, "Is MAP decoding all you need? the inadequacy of the mode in neural machine translation," in *Proc. International Conference on Computational Linguistics*, 2020.

[46] F. Stahlberg, A. de Gispert, E. Hasler, and B. Byrne, "Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices," in *Proc. ACL*, 2017, pp. 362–368.

[47] M. Müller and R. Sennrich, "Understanding the properties of minimum Bayes risk decoding in neural machine translation," in *Proc. ACL*, 2021, pp. 259–272.

[48] B. Eikema and W. Aziz, "Sampling-based approximations to minimum Bayes risk decoding for neural machine translation," in *Proc. EMNLP*, 2022, pp. 10 978–10 993.

[49] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," 2020, arXiv:2006.13979.

[50] R. Ardila *et al.*, "Common Voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2020, pp. 4218–4222.

[51] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[52] A. Q. Jiang *et al.*, "Mistral 7B," 2023, arXiv:2310.06825.