Speech Disfluency Detection with Contextual Representation and Data Distillation

Payal Mohapatra
PayalMohapatra2026@u.northwestern.edu
Northwestern University
Evanston, Illinois, USA

Akash Pandey
AkashPandey2026@u.northwestern.edu
Northwestern University
Evanston, USA

Bashima Islam
bislam@wpi.edu
Worcester Polytechnic Institute
Worcester, USA

Qi Zhu
qzhu@northwestern.edu
Northwestern University
Evanston, USA

ABSTRACT

Stuttering affects almost 1% of the world's population. It has a deep sociological impact and hinders the people who stutter from taking advantage of voice-assisted services. Automatic stutter detection based on deep learning can help voice assistants to adapt themselves to atypical speech. However, disfluency data is very limited and expensive to generate. In this work, we propose a set of preprocessing techniques: (1) using data with high inter-annotator agreement, (2) balancing different classes, and (3) using contextual embeddings from a pretrained network. We then design a disfluency classification network (DisfluencyNet) for automated speech disfluency detection that takes these contextual embeddings as an input. We empirically demonstrate high performance using only a quarter of the data for training. We conduct experiments with different training data size, evaluate the model trained on the lowest amount of training data with SEP-28k baseline results, and evaluate the same model on the FluencyBank dataset baseline results. We observe that, even by using a quarter of the original size of the dataset, our F1 score is greater than 0.7 for all types of disfluencies except one, *blocks*. Previous works also reported lower performance with *blocks* type of disfluency owing to its large diversity amongst speakers and events. Overall, with our approach using only a few minutes of data, we can train a robust network that outperforms the baseline results for all disfluencies by at least 5%. Such a result is important to stress the fact that we can now reduce the required amount of training data and are able to improve the quality of the dataset by appointing more than two annotators for labeling speech disfluency within a constrained labeling budget.

CCS CONCEPTS

• **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*; • **Social and professional topics** → *People with disabilities*; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Contextual Representation, Neural Networks, Deep Learning, Speech Disfluency

IASA'22, July 1, 2022, Portland, OR, USA

ACM ISBN 978-1-4503-9403-1/22/07…$15.00
https://doi.org/10.1145/3539490.3539601

1 INTRODUCTION

Voice-assisted devices (e.g., Amazon Echo, Google Home, Apple's Siri) make it easier to interact with technology hands-free. There is a projection of 8 billion intelligent personal assistants with voice interfaces by 2023 [30]. Such a technology boasts conversational interaction with devices and improves their day-to-day utility. It also benefits many minority user groups, for example, people who are blind or have challenged limbs and cannot use the text mode of communication. However, such technology remains unreachable to another demographic, the people who have atypical speech patterns like stuttering [33]. Stuttering or stammering affects about 70 million people worldwide [8]. It has a deep interference with the social and work life normalcy of people who stutter (PWS). As voice interfaces become more commonplace, it is important to pivot research in the direction of inclusivity of PWS. To make the current voice assistants more inclusive, the first step is to build the capability of automatic speech disfluency identification [7]. This is a stepping stone for voice assistants to then adapt to the speech of PWS. In some literature [31] the terms stuttering and disfluency are used interchangeably, however, in a pathological sense, they are distinguished in some other works [1]. In this work, we use them interchangeably to denote different types of speech disfluencies.

Traditionally, a speech pathologist labels the disfluency of an individual under assessment. As a pathologist cannot be available at all times, we need to develop an automatic speech disfluency detector. In the past, researchers have dominantly used spectral audio features like Mel-frequency cepstral coefficients (MFCC), linear prediction cepstral coefficients, Fourier Transforms, energy peaks, and temporal features like amplitude, zero crossing, etc. to recognize stuttering events. Elmar et. al [21]. have shown that using Hidden Markov Models as a screening tool for patients in a speech therapy session has proven as an effective tool for stuttering identification. Other statistical classifiers like Linear Discriminant Analysis, k-nearest neighbors [5], Gaussian Mixture Models [18], and Support Vector machines [26] are other popular choices for

classifiers. All of these works have focused on very specific kinds of speech disfluencies like prolongation and repetition using data collected under a very controlled setting (in therapy sessions or protocol induced). These models may not translate well if the data is collected using a different microphone or with background noise.

With the rising popularity of deep-learning methods which have the capability to model very complex non-linearities in decision boundaries, it is a promising direction to explore using them for automatic stutter detection. Shakeel et. al [27] have used a sequence modeling approach using audio features like MFCCs to classify five different types of speech disfluencies. Another common approach is to use speech-to-text encoding for disfluency detection [29]. However, in spite of deep learning tools showing a lot of promise, they need a significant amount of data for training, which is often difficult and expensive to collect.

One of the popular speech disfluency datasets is from the University College London Archive of Stuttered Speech (UCLASS) [12], collected mostly from school children by asking them to read out monologue samples. However, they are not labeled and previous work [15] on stutter-identification using this dataset has not publicly released their annotations. FluencyBank [24] is another popular corpus of disfluency data that has been used to study stuttering [25] in earlier works, but it too does not have publicly available labels. Libristutter [16] is a synthetically generated disfluency dataset from LibriSpeech[22], which is used for some disfluency studies due to lack of labeled data. Recently, Lea et.al [17] have released a labeled dataset, SEP-28k, for speech disfluencies along with an effort to label and release FluencyBank annotations. SEP-28k is one of the first datasets collected from eight openly available online podcasts by PWS which are not recorded by prescription or adhering to any clinical protocol. This is the closest to an *in-the-wild* dataset. Speech disfluency dataset are expensive to create since in a real-world setting a speech pathologist relies on visual cues apart from auditory signatures. This makes labeling of the already collected speech data prone to subjective annotations if multiple trained annotators are not consulted. So far, SEP-28k is the most reliable publicly available speech disfluency dataset which is collected systematically using three annotators for data.

In this work, we focus on addressing speech disfluency detection with *limited data*. One of the very few works in this direction [29] uses 16.8 hours of data while reporting over 70% accuracy. They have used a custom dataset to demonstrate their results. We use the SEP-28k dataset (which is currently a standard dataset for speech disfluencies) to explore the use of limited data for training and evaluating the performance with respect to the baseline results [3, 17] reported in recent works. Although the research literature on speech disfluency is rich, one of the main drawbacks so far has been the unavailability of a reliable public dataset using which previously proposed techniques can be reproduced and juxtaposed. To address this, we use a public dataset and also release our implementation details for easy reproducibility of our results, [1]. Our main contributions are as follows:

- We demonstrate that by using contextual representations and data distillation techniques at the preprocessing stage, our approach can reduce the need for training data and outperform state-of-the-art baselines.
- We demonstrate the efficacy of our method empirically on various dataset sizes. To the best of our knowledge, this is the first study that has used data in the order of minutes for training a disfluency classifier with robust performance across multiple dataset evaluations.
- We develop a DisfluencyNet model that is trained on the SEP-28k dataset and evaluated on the FluencyBank dataset and a portion of SEP-28k that was held out prior to training.

[1]https://github.com/payalmohapatra/Speech-Disfluency-Detection-with-Contextual-Representation-and-Data-Distillation.git

2 DATA DESCRIPTION

SEP-28k (Stuttering Events Podcasts) [17] is an open-source dataset collected from eight online podcasts hosted by PWS, resulting in twenty-eight thousand data points. The data is systematically labeled by three trained annotators. There are five types of disfluencies in this dataset whose definitions are given in Table 1. Lea et. al [17] have also labeled and released a dataset from FluencyBank [24] with corresponding disfluency labels resulting in about four thousand data points. Previous works have shown that using an audio segment of length less than 5 seconds is ideal for stuttering identification [11, 27]. All the audio segments are 3 secs in length and sampled at 16kHz, which make an ideal design choice for disfluency detection. For our study, we have sampled data from the SEP-28k distribution only for training the model and used the data held-out from SEP-28k and data from FluencyBank to evaluate the model. More details about the training and evaluation using different slicing of data are given in Section 4.

3 METHODOLOGY

Table 1: Definition of different speech disfluencies.

Disfluency Type	Definition
Sound Repetition (Snd)	Intra-word phoneme repetition
Word Repetition (WP)	Repetition of any word
Prolongation (Pro)	Extended sounds within a word
Interjection (Intrj)	Filler words or non-words
Blocks (Bl)	Long unnatural pauses

3.1 Preprocessing

Disfluency rating is a subjective score. The Fleiss Kappa [20] inter-rater agreement reported by Lea et. al [17] ranges from 0.11 to 0.62 per disfluency. This motivated us to use a filtering step to only sample the data points where there is no ambiguity of class between the annotators. We use the segments where all annotators agree on the chosen task. This helps us maintain higher quality data for training and evaluation. We had also conducted a pilot study on data where two or more annotators agree on the assigned class for Snd type of disfluency. We achieved an F1 score (refer to Equation (2)) of 0.63, which we will later show in Table 3 is about 25% sub-par than our current predictions with the full dataset. Since the results were not encouraging, we proceed to use only the data points where there is no disagreement between annotators. Table 2 summarises

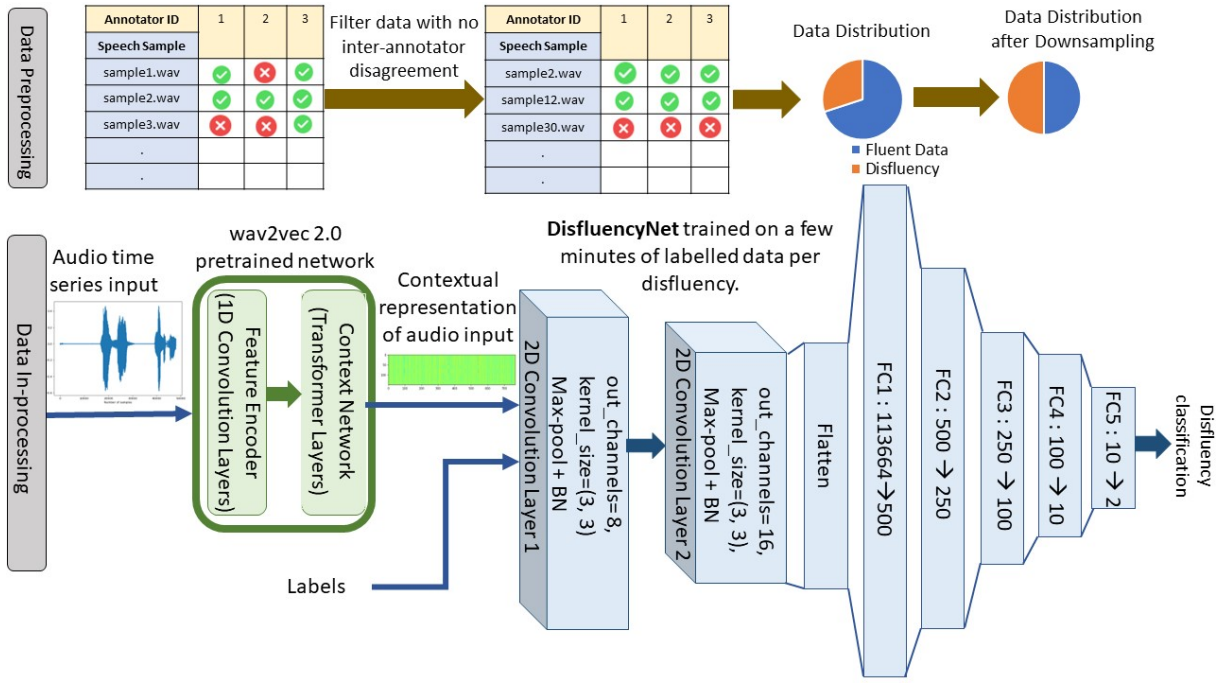

Figure 1: Overall architecture of our approach with 1) a data preprocessing stage with inter-annotator agreement based filtering and down sampling, and 2) an in-processing stage that extracts embeddings from the wav2vec2.0 pretrained network to be used as inputs to DisfluencyNet. Representative specifications of DisfluencyNet for the convolution and fully connected (FC) layers are shown for a model trained on a dataset of quarter size for WP disfluency.

the dataset sizes when 2 or more annotators agree versus dataset size where all three annotators agree. We can observe that in all the classes at least 50% of the data is rejected when three annotators' labels are taken into account. In many classes (Pro, Bl, Intrj), only 30% of the data remains where there is no ambiguity of labels. This distillation significantly improves the quality of the data we use for training. Contrary to the results shown by Lea et.al [17] in their figure titled *Impact of Data Quantity on Dysfluency Detection*, we in fact demonstrate in this study that quality of data has a stronger positive impact than mere quantity. Certainly, higher quality data in greater quantity is naturally beneficial, but data augmentation without quality control is detrimental.

In this study, we conduct single task learning (STL) for every disfluency type against the fluent class. For every task, we down-sample the dominant class (which is the fluent class in all the cases) to address the class imbalance. In our pilot studies, we found that ensuring class balance showed superior performance over using a weighted loss function for optimization.

In this work, we explore the use of representation learning to provide a more structured prior knowledge to the classification network. We take inspiration from the Automatic Speech Recognition (ASR) methods used in Natural Language Processing (NLP) for speech representations. In particular, the wav2vec 2.0 framework [2] is trained in a task-independent style using self-supervised learning on normal speech data. It consists of temporal convolution layers to generate a latent embedding followed by transformer layers to generate a contextual encoding. wav2vec2.0 is optimized using a contrastive loss function. The positive and negative inputs to the contrastive loss are given by masking the latent embeddings of the raw audio waveform. It is trained on 960 hours of LibriSpeech [22] data. There are two models presented in the wav2vec 2.0 framework. They have the same structure for feature encoder but one, BASE, uses 12 transformers to generate features with 768 dimensions and the other, LARGE, has 24 transformers with features of 1024 dimensions. We choose the BASE model to extract embeddings from our input audio waveform since we want to optimise the downstream task (DisfluencyNet) with limited data. The preprocessing steps are shown in Fig. 1. In our preprocessing stage we work on transforming the dataset to a balanced distribution and convert the time series audio input (of size (1,48000)) to a 2-dimensional(of size (1, 149,768)) feature embedding.

Since the training of wav2vec 2.0 network is done in independent of any task, it is an ideal candidate for many ASR downstream tasks like speech-to-text conversion which can be fine-tuned with limited training data. We identify the efficacy of using such an embedding for speech disfluency representation as well.

3.2 Disfluency Classifier : DisfluencyNet

The inputs to DisfluencyNet is the contextual embedding from the preprocessing stage of dimension (1,149,768). The primary building blocks of DisfluencyNet are 2D convolution layers with max-pooling and fully connected layers as shown in Fig. 1. The convolution layers use a rectified linear unit(ReLU) to model the system's non-linearities and a dropout with a probability 0.5. The outputs

Table 2: Number of samples based on the number of annotators agreement.

Disfluency	2 or more annotators agreement	3 annotators agreement(% column 2)
Snd	2342	863 (36.8 %)
WP	2770	1610 (58.1%)
Pro	2812	790 (28.1%)
Intrj	5973	3378 (56.5%)
Bl	3370	528 (15.7%)

of all the layers are batch normalised [13]. The output of the last convolution layer is flattened and fed to fully connected layers which use a leaky ReLU activation function. We use two layers of the 2D convolution with a kernel size of (3,3). The fully connected layers after flattening the output of the second convolution layer's output intuitively follow an encoder like structure to decrease the number of outputs at every subsequent layer. For experiments with different dataset sizes, the fully connected layers' dimensions may vary slightly for better optimization. The output layer uses a softmax function to compute the probability of the input belonging to a given class. For this classification task, cross-entropy loss as shown in Equation (1), where p and q are probability distributions of the target and predicted labels given an input random variable x, is used to optimise the model. We use the Adam optimizer [14] for minimising Equation (1).

$$H(p,q) = -\sum_{x \in X} p(x) log(q(x)) \tag{1}$$

We conduct these experiments on an Ubuntu 20.04 OS server equipped with NVIDIA RTX A5000 GPU cards, Python 3.9.7, and Pytorch 1.11.0 [23]. From the total available data, 20% of data from every class is held out for testing. From the remaining data, 80% is used for training and the rest for validation. We use early stopping based on validation performance on the trained model to avoid overfitting and a learning rate between 1e-4 to 1e-2.

4 EXPERIMENTAL STUDY

4.1 Experimental Setup

To systematically evaluate the various settings, we have randomly shuffled and segregated 20% of data from every disfluency class. This is our testing dataset which is used to compute the performance metrics in all the settings. The split between training and testing data can be a major variance in model evaluations. We have released the testing dataset used for our performance analysis, so that future studies can reproduce our observations as a baseline. We have chosen the performance metrics of precision, recall, F1 score, and accuracy given as:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP}, \; Recall = \frac{TP}{TP+FN} \\ F1\ score &= \frac{2}{(1/Precision) + (1/Recall)} \\ Accuracy &= (TP+TN)/(TP+TN+FN+FP) \end{aligned} \tag{2}$$

where TP, TN, FN, FP are True Positive, True Negative, False Negative and False Positive, respectively. Since we have downsampled the dominant class in the preprocessing step, we now have balanced classes. For test as well the data is sampled from this distribution, hence accuracy and balanced accuracy in fact give similar results as there is no skewed target class.

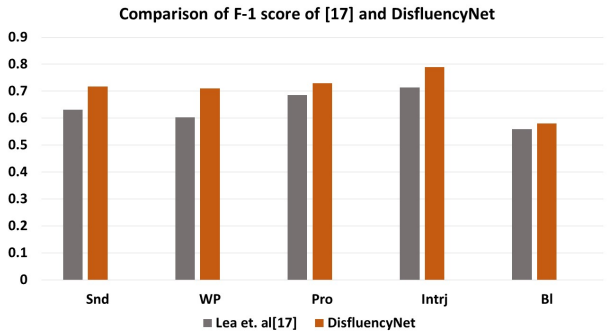

Figure 2: Comparison of F1 scores for our DisfluencyNet trained on a quarter of the data vs. the results reported in [17] on SEP-28k dataset.

4.2 Evaluation on Limited Disfluency Data

As stated, we are interested in evaluating the performance of our approach with smaller dataset sizes. So far only one study [29] has addressed the problem of limited dataset size in speech disfluency domain and reported about 70% accuracy with 16.8 hours of training data. We show that we can achieve high performance using dataset size *in the order of minutes with our approach*. Since the evaluation of [29] is on a custom dataset and the implementation is unavailable, we are not able to juxtapose our approach against theirs.

We consider three divisions of the dataset: sampling all the data, half of the data and, a quarter of the data. This sampling is done after shuffling the data so that data from all podcasts are present in a given distribution. Resultant models from different folds of sampled data are saved and evaluated on the same test dataset. The performance metrics for each disfluency type on models trained with different sizes of data are reported in Table 3. We can observe that even with a few minutes of data (in the results from the quarter dataset), we can achieve accuracies greater than 70% for all the disfluencies except Blocks. Blocks-type of disfluency tend to last longer in a speech segment and more speculation needs to be exercised while working with this label [17]. This might be the underlying reason for the lower performance score of Blocks compared to other disfluencies.

4.3 Evaluation on SEP-28k Data

We compare the performance of our model trained on 25% of the data against the baseline results reported by Lea et. al [17] on the entire SEP-28k dataset as shown in Fig. 2. We observe that the DisfluencyNet trained on a quarter of the dataset consistently outperforms [17] baseline results for all types of disfluency by at least 5%. Lea et al. have not reported their test-train split but their details suggest that testing is not cherry-picked (e.g., leaving one

Table 3: Results for all disfluency on the SEP-28K dataset.

Disfluency	Dataset	Data size in minutes	F1 score	Precision	Recall	Test Accuracy(%)
Snd	Full Dataset	75	0.87	0.78	0.99	86.00
Snd	Half Dataset	37	0.79	0.78	0.80	80.00
Snd	**Quarter Dataset**	**19**	**0.72**	**0.67**	**0.79**	**70.00**
WP	Full Dataset	148	0.87	0.78	0.98	88.90
WP	Half Dataset	74	0.75	0.76	0.74	78.00
WP	**Quarter Dataset**	**37**	**0.71**	**0.75**	**0.66**	**71.00**
Pro	Full Dataset	75	0.95	0.91	0.99	94.90
Pro	Half Dataset	37	0.85	0.85	0.85	85.90
Pro	**Quarter Dataset**	**19**	**0.73**	**0.8**	**0.76**	**75.70**
Intrj	Full Dataset	248	0.88	0.83	0.93	82.70
Intrj	Half Dataset	124	0.81	0.9	0.75	75.60
Intrj	**Quarter Dataset**	**62**	**0.79**	**0.79**	**0.79**	**74.50**
Bl	Full Dataset	45	0.75	0.76	0.73	75.00
Bl	Half Dataset	22	0.68	0.66	0.71	67.30
Bl	**Quarter Dataset**	**11**	**0.58**	**0.54**	**0.61**	**55.00**

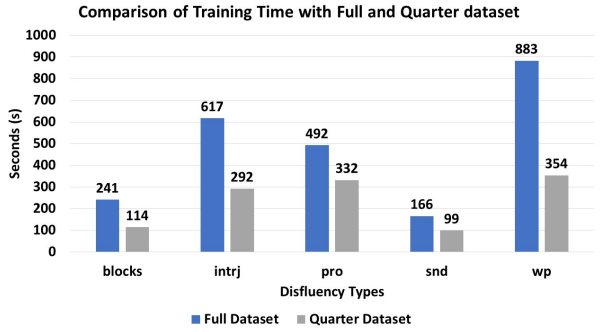

Figure 3: Comparison of training times with full dataset and quarter of the dataset.

speaker out and so). Since the dataset generalizes well, we have assumed that with different splits of test data the performance variation should not be very drastic. We want to revisit Table 2 and draw attention to the fact that data distillation reduces the training set size but does not remove dependency on a larger dataset (for eg. in Snd you need 2000 labelled data by 3 annotators to have an absolute agreement about 36%). We want to emphasize on the fact that by demonstrating successful training using only 25% of the distilled data we reduce the overall dependency on a larger dataset (for eg. for Snd we now only need 600 labeled data). It is an obvious outcome that with a reduced dataset the training time for the respective models decreases as shown in Fig. 3.

4.4 Evaluation on FluencyBank Data

Very recent works by Bayerl et. al [3] and Sheikh et. al [28] have also explored the use of contextual embeddings from the wav2vec2.0 pretrained model. The former [3] focuses on evaluating the importance of internal embeddings of wav2vec2.0 over the embeddings obtained from the final transformer layer for this task. The latter [28] explores the use of embeddings with statistical classifiers and shallow neural networks to conduct a multiclass classification. Our objective is to evaluate the impact of dataset sizes for STL classification of speech disfluencies. However, we compare our results with [3] on STL per disfluency.

To verify the robustness of the trained model, we evaluate it on the data sampled from FluencyBank. We compare the F1 scores of our DisfluencyNet with those from approaches in [3] and [17] in Fig. 4. Only for the disfluency type Interjection, does the model in [3] outperforms the DisfluencyNet trained on the quarter dataset. However, when we compare the F1 score of DisfluencyNet when trained on the full dataset (0.82) against [3] (0.83), the scores are almost equal. For all other disfluencies, the DisfluencyNet trained on only a quarter of the dataset from SEP-28k outperforms the other approaches.

We would like to remark that SEP-28k being collected from open source podcasts brings a domain generalization (recorded on different devices, myriad post-processing, background music, etc.) in the dataset as opposed to other data (UCLASS, FluencyBank, KSoF custom dataset, etc.) being collected in a controlled setting. This helps the model to perform well on data sampled from a different distribution as well (such as the Fluencybank).

5 CONCLUSION AND FUTURE DIRECTIONS

Stuttering is a pathological condition that can affect the overall social and professional well-being of an individual. It hinders their participation in voice-assisted technology services. We need to develop techniques to accommodate for users with speech disfluencies. But data corresponding to speech disfluencies are very expensive and hence, limited. We propose an approach with contextual embeddings and data distillation followed by a DisfluencyNet to use only a few minutes of data for disfluency classification. We further show that our approach outperforms the state-of-the-art reported results in most of the cases by using only a quarter of the data for training compared to the rest. To verify the robustness of our model, we also evaluate our trained network on data sampled from a different dataset. Training on a few minutes of preprocessed data that

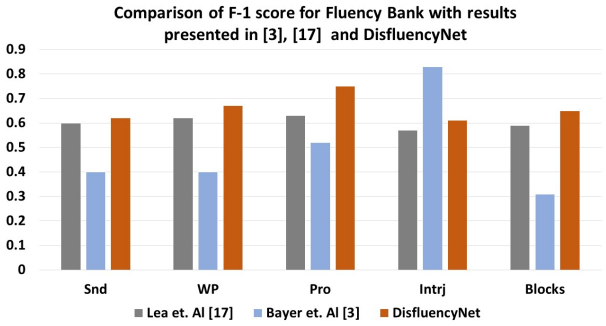

Figure 4: Comparison of F1 score for Fluency Bank with results presented in [3], [17] and our DisfluencyNet trained on a quarter of the data.

contains prior structured information results in a robust model that performs well for classifying different types of disfluency against the fluent class.

Since we have sizeable unlabeled data for speech disfluencies, we are motivated to explore semi-supervised learning under a strict labeling budget with active learning [19], self-supervised learning [4], and weak adaptation [34] techniques to further speech disfluency detection. We are also interested in studying the detection of speech disfluencies under a federated learning [9, 10, 32] setting given the ubiquity of voice assistants in most households and aim at personalizing [6] it for a user.

ACKNOWLEDGEMENT

This work is supported in part by NSF grant 1834701.

REFERENCES

[1] Nicoline Grinager Ambrose and Ehud Yairi. 1999. Normative disfluency data for early childhood stuttering. *Journal of Speech, Language, and Hearing Research* 42, 4 (1999), 895–909.
[2] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems* 33 (2020), 12449–12460.
[3] Sebastian P Bayerl, Dominik Wagner, Elmar Nöth, and Korbinian Riedhammer. 2022. Detecting Dysfluencies in Stuttering Therapy Using wav2vec 2.0. *arXiv preprint arXiv:2204.03417* (2022).
[4] Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan Raducanu. 2021. Reducing label effort: Self-supervised meets active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1631–1639.
[5] Lim Sin Chee, Ooi Chia Ai, M Hariharan, and Sazali Yaacob. 2009. MFCC based recognition of repetitions and prolongations in stuttered speech using k-NN and LDA. In *2009 IEEE Student Conference on Research and Development (SCOReD)*. IEEE, 146–149.
[6] Huili Chen, Jie Ding, Eric William Tramel, Shuang Wu, Anit Kumar Sahu, Salman Avestimehr, and Tao Zhang. 2022. ActPerFL: Active Personalized Federated Learning. In *ACL 2022 Workshop on Federated Learning for Natural Language Processing*.
[7] Leigh Clark, Benjamin R Cowan, Abi Roper, Stephen Lindsay, and Owen Sheers. 2020. Speech diversity and speech interfaces: Considering an inclusive future through stammering. In *Proceedings of the 2nd Conference on Conversational User Interfaces*. 1–3.
[8] Edward G Conture and Lesley Wolk. 1990. Stuttering. In *Seminars in Speech and Language*, Vol. 11. © 1990 by Thieme Medical Publishers, Inc., 200–211.
[9] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. 2022. Federated Class-Incremental Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[10] Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. 2019. Active federated learning. *arXiv preprint arXiv:1909.12641* (2019).
[11] Peter Howell. 2005. The effect of using time intervals of different length on judgements about stuttering. *Stammering research: an on-line journal published by the British Stammering Association* 1, 4 (2005), 364.
[12] Peter Howell, Stephen Davis, and Jon Bartrip. 2009. The university college london archive of stuttered speech (uclass). (2009).
[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.
[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[15] Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad. 2020. Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6089–6093.
[16] Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad. 2020. FluentNet: end-to-end detection of speech disfluency with deep learning. *arXiv preprint arXiv:2009.11394* (2020).
[17] Colin Lea, Vikramjit Mitra, Aparna Joshi, Sachin Kajarekar, and Jeffrey P Bigham. 2021. Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6798–6802.
[18] P Mahesha and DS Vinod. 2017. LP-Hillbert transform based MFCC for effective discrimination of stuttering dysfluencies. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2561–2565.
[19] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764* (2021).
[20] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.
[21] Elmar Nöth, Heinrich Niemann, Tino Haderlein, Michael Decher, Ulrich Eysholdt, Frank Rosanowski, and Thomas Wittenberg. 2000. Automatic stuttering recognition using hidden Markov models.. In *INTERSPEECH*. 65–68.
[22] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.
[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[24] Nan Bernstein Ratner and Brian MacWhinney. 2018. Fluency Bank: A new resource for fluency research and practice. *Journal of fluency disorders* 56 (2018), 69–80.
[25] Rachid Riad, Anne-Catherine Bachoud-Lévi, Frank Rudzicz, and Emmanuel Dupoux. 2020. Identification of primary and collateral tracks in stuttered speech. *arXiv preprint arXiv:2003.01018* (2020).
[26] Shakeel Ahmad Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2021. Machine Learning for Stuttering Identification: Review, Challenges & Future Directions. *arXiv preprint arXiv:2107.04057* (2021).
[27] Shakeel A Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2021. StutterNet: Stuttering Detection Using Time Delay Neural Network. In *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 426–430.
[28] Shakeel Ahmad Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2022. Introducing ECAPA-TDNN and Wav2Vec2. 0 Embeddings to Stuttering Detection. *arXiv preprint arXiv:2204.01564* (2022).
[29] 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* (2022).
[30] S Smith. 2018. Digital voice assistants in use to triple to 8 billion by 2023, driven by smart home devices.
[31] John A Tetnowski, Kathleen Scaler Scott, and Brittany Falcon Rutland. 2021. Fluency and fluency disorders. *The handbook of language and speech disorders* (2021), 414–444.
[32] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. 2021. Addressing Class Imbalance in Federated Learning. In *AAAI*.
[33] K Wheeler. 2020. For people who stutter, the convenience of voice assistant technology remains out of reach. *USA Today (online)* (2020).
[34] Shichao Xu, Lixu Wang, Yixuan Wang, and Qi Zhu. 2021. Weak Adaptation Learning: Addressing Cross-Domain Data Insufficiency With Weak Annotator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 8917–8926.