# Deep Learning based Speech Disfluency Detection

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science*
*in*
***Electronics and Communication Engineering*** *by Research*

by

Sparsh Garg
20161025
sparsh.garg@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2022

International Institute of Information Technology
Hyderabad, India

# CERTIFICATE

It is certified that the work contained in this thesis, titled **"Deep Learning based Speech Disfluency Detection"** by **Sparsh Garg**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Anil Kumar Vuppala

To My Parents

# Acknowledgments

I want to express my first and foremost gratitude to my advisor Prof. Anil Kumar Vuppala. This work would not have been possible without his support. His approach to do research gradually and one step at a time, not only provided an excellent base for my research but also set a right attitude to do any work. His constant and invaluable feedback helped me do the work in the best possible manner. Next, I would like to thank my mentor, Dr Gurugubelli Krishna, his knowledge in any field is amazing. From providing newer and better ways to solve the problem, to helping in learning how to write a good paper, his guidance is very beneficial to anyone. I would also like to thank my friend and my research buddy Utkarsh Mehrotra for always readily having discussions to solve the problem and for helping me in paper-writing and coding whenever required. I would also like to extend my gratitude to two of my dear friends, Shashwat Khandelwal and Shashwat Shrivastava, for always motivating me to do better work and teaching me how to maintain calm while doing research to get a positive mind. My other friends Saksham, Anshul, Mohit, Nikhil, Madhur, Sachin, always listened to my problems and supported me. They gave me memories to enjoy my entire life.

I want to sincerely thank my parents for providing me with proper education and setting a great platform to achieve what I want. Without their teachings and blessings, this would not have been possible. They always ensured that I remained in the right frame of mind for my work and didn't have to worry about other things. Finally, I would like to thank my elder brother Kushagra for holding my back and making corona times easier.

# Abstract

Spontaneous speech is a particular type of speech setting where a speaker speaks without preparing in advance. This makes the speaker think about what to say on the spot, formulate the utterances and then produce the speech. Such a setting often leads to abrupt breaks or discontinuities in the normal conversation flow called disfluencies. Disfluencies can provide information regarding the speaking style, speaker identity and language fluency, which can be useful for several speech-based applications. For automatic speech recognition (ASR) systems, the presence of these disfluencies leads to a higher word error rate, since most ASR systems are developed on non-spontaneous read speech data. Thus, the detection of disfluencies in spontaneous speech becomes an essential task for many applications. For training any machine learning system, one need data. In this thesis, we introduce IIITH-IED dataset for disfluencies in spontaneous lecture-mode speech, and then use it to develop frame-level automatic disfluency detection systems. Finally, we propose a transfer learning method to detect disfluencies in spontaneous lecture mode speech using three frame-level automatic disfluency detection systems trained on stuttered speech. Compared to baseline systems, proposed method gives an average improvement of 2.25% across all disfluencies and all detection systems.

# Contents

# List of Figures

# List of Tables

*Chapter 1*

# Introduction

## 1.1 Speech Disfluencies

Humans convey their thoughts and emotions to each other by producing a sound that carries meaning with it and which the listener can decode. These sounds produced by humans, which carry significant meaning with them, is called speech. Depending on the amount of preparation and the nature of the event, speeches can be divided into two broad categories:

- Spontaneous Speech - The term "spontaneous speech" refers to a situation in which a speaker is expected to talk without any prior preparation. On the fly, the speaker thinks and formulates sentences. [16] defines a spontaneous utterance as: "a statement conceived and perceived during its utterance". An example of this type of speech can be 2 people talking over a phone call.

- Non-Spontaneous Speech - The term "non-spontaneous speech" refers to a situation in which the speaker has deliberated over and prepared the content of his or her utterance before delivering the speech. An example of this type of speech can be when people read from the prepared text.

Due to the speaker's hesitations, spontaneous speech setting often results in abrupt breaks and discontinuity in normal conversation flow. These breaks in speech occur for a variety of reasons, including language complexity, nervousness while speaking, and the need for time to formulate thoughts while speaking [41, 11]. These abrupt breaks or hesitations in the normal flow of speech are referred to as speech disfluencies. In speech production, fluency refers to the consistency and smoothness involved while speaking. At times, all speakers become disfluent. When speaking, they may pause, utilise filler words (such as "uhh" or "umm"), or repeat a word or phrase.

Disfluencies can take on a variety of forms depending on the type of discontinuity and the speaker's method of overcoming or correcting the discontinuity. Disfluencies in speech include filled pauses, prolongations, repetitions, and revisions. Table 1.1 briefly defines the most frequently encountered types of disfluencies in spontaneous speech, along with examples. A more detailed description of these disfluencies is provided in Chapter 2.

**Table 1.1** Description of different types of disfluencies.

| **Disfluency Type** | **Description** |
|---|---|
| Filled Pause | Pauses in speech with filler words. eg.- uhh, umm |
| Prolongations | Lengthening of a particular sound or syllable. eg.- whoooose book it is |
| Word Repetition | Repetition of complete word. eg.- small small |
| Part-word Repetition | Repetition of particular phoneme. eg.- th-this |
| Phrase Repetition | Repetition of more than one words or phrase. eg.- I am I am |
| Revisions | Amending the original utterance by using the similarly structured phrase. eg.- I went to London uhh I went to Sydney |

## 1.2 Detection of Disfluencies

In recent years, the development of robust speech-based applications, which can be used in multiple settings, has been the focus of a number of studies [14, 6]. The presence of disfluencies in speech can adversely affect the performance of many such applications. For ASR systems, disfluencies lead to higher word error rates (WER) since most ASRs are developed for read and non-spontaneous speech. In the case of a system like speech-to-speech translation, this error is further propagated to downstream applications (machine translation and text to speech systems), increasing the total error of the pipeline by many folds [33]. Hence, detection and removal of disfluencies from speech signal prior to giving it as input to these applications become crucial for getting an appreciable performance.

In general, the majority of methods for detecting speech disfluency fall into one of the following categories: as a post-processing step following ASR output, which employs text-based features along with speech for disfluency detection [49, 31] or as a pre-processing step before giving input to a speech system by using signal level methods to identify them [39, 22]. While the former approach is more effective and produces encouraging results, it is computationally more expensive and prone to errors due to its reliance on ASR performance.

## 1.3 Objective and Scope

The primary objective of this thesis is to investigate and develop models for detecting disfluency in spontaneous English speech by exploring different machine learning techniques. Disfluency detection is performed as a pre-processing step in the current work, using a speech segment as an input to develop frame-level disfluency detection systems that aid in the removal of disfluencies from speech segments. The scope of this thesis is as follows:

- Introduction of the IIITH-IED dataset in order to facilitate studies on speech disfluencies in Indian English.
- Using the IIITH-IED dataset to develop automatic disfluency detection systems by employing a variety of machine learning methods.
- Proposed a transfer learning approach to detect disfluencies in conversational speech using a model trained on stutter data.

## 1.4 Thesis outline

The organization of this thesis is as follows:

- Chapter 2 provides a detail description of speech disfluencies, terminologies used in work, and advancement of the domain through literature review.
- Chapter 3 contains details about the IIITH-IED dataset and baseline disfluency detection systems developed on it.
- Chapter 4 discusses our proposed transfer learning method for detecting disfluencies in spontaneous speech by taking knowledge from stutter data.
- Chapter 5 concludes our work and provides the scope for future works.

*Chapter 2*

# Disfluencies in Speech

## 2.1 Introduction

Disfluency is defined as an interruption in the flow of speech that occurs due to the speaker's hesitations, which can be due to a variety of reasons, as discussed in Section 1.1 briefly. A more detailed insight on causes of disfluencies in spontaneous speech, along with their classification, will be provided in the following section.

## 2.2 Types of Disfluencies

From a standard standpoint, disfluencies can be classified in spontaneous speech as-

- Those which involve a delay in production process and don't carry any linguistic meaning, like filled pauses and prolongation.
- Those in which a part of a utterance is repeated, like part-word repetitions, word repetitions, and phrase repetitions.
- Those in which a part of utterance is altered or rephrased like repairs.

These disfluencies are described in more detailed way in following subsections:

### 2.2.1 Filled Pause

Filled pause, also referred as interjection, is the vocalised form of pause which is used by the speaker to hold back for a time while processing an utterance in spontaneous speech. In English, filled pauses generally fall into one of the two categories, "um" or "uh". But categorising orthographically is much easier than phonetically, as it can take many forms depending on the speaker's accent. In most of the cases, filled pause is used by speakers to buy some time for speaking next utterance, while they are still prepairing it in their mind. It can be seen as an instant when the speaking and the prepairing process for the utterance come at conflict.

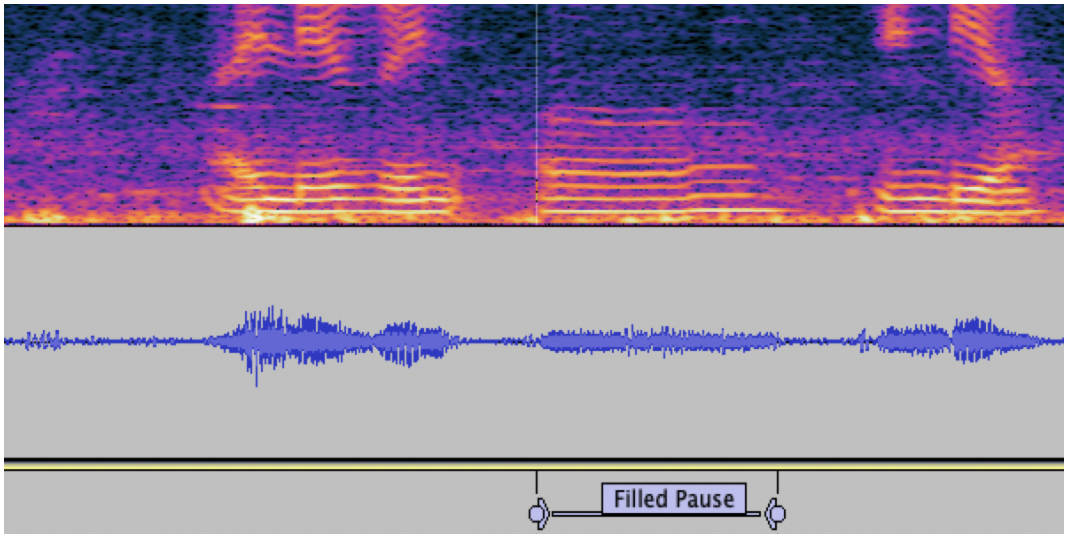


**Figure 2.1** Waveform and Spectogram plot for sentence “I live in uhh India.”

Because such a sound is produced when the vocal cords vibrate with approximately static articulator parameters [20], a filled pause involves a continuous voiced sound of a consistent spectral structure as shown in Figure 2.1

### 2.2.2 Prolongations

Prolongations can be referred to as lengthening of a particular sound or syllable, which tends to make the word longer than a fluent word. Prolongation can be of vocalized as well as non-vocalized sounds. They are often accompanied with an increase in loudness or pitch of the sound.

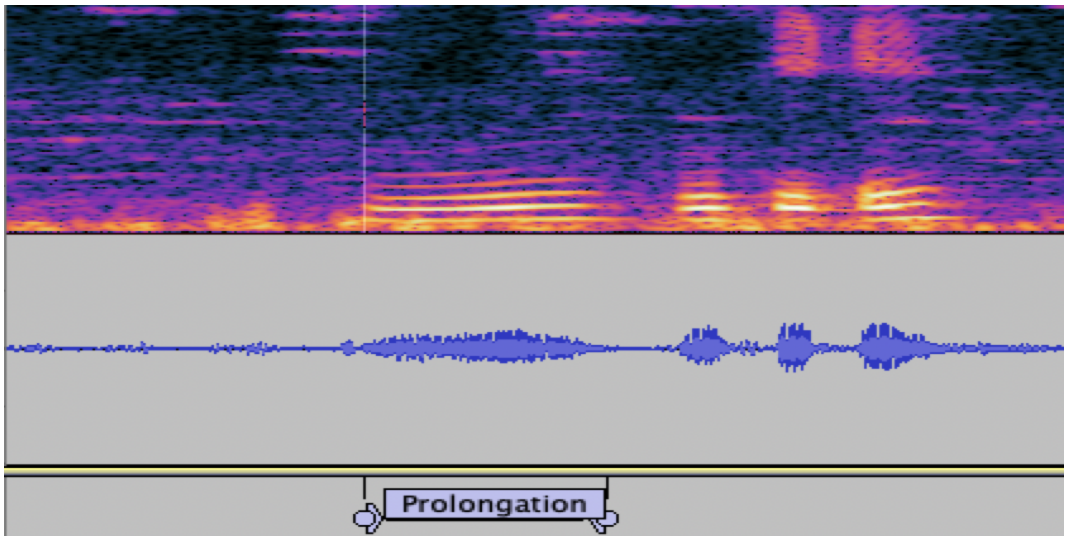


**Figure 2.2** Waveform and Spectogram plot for sentence “whooose book it is?”

One of the differential properties of prolongations that we can see from Figure 2.2 is that throughout the prolonged segment, it exhibits a consistent spectral structure which is due to the continuation of the same sound through the segment.

### 2.2.3 Repetitions

Repetition, as the name suggests, is the type of disfluency that results due to the repetition of a part of an utterance, disrupting the speech flow; it can be at different levels, i.e. syllabic, word level, or phrase level. The basic model of repetition is a unit of speech followed by a pause, followed by another unit of speech, as one can infer from the below figure.

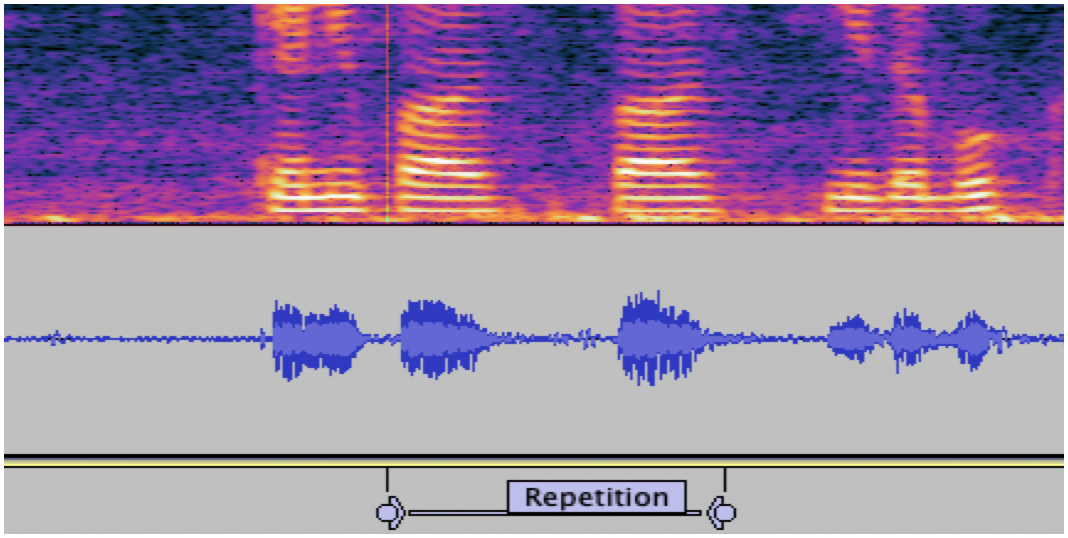


**Figure 2.3** Waveform and Spectogram plot for sentence "Can you pass pass me the book?"

Depending on the repeated unit, repetitions can be further classified into 3 categories:

- Part-Word Repetition - Repetition of particular phoneme. eg.- **Wh-what** is your name ?
- Word Repetition - Repetition of complete word. eg.- Please pass me **the the** book.
- Phrase Repetition - Repetition of more than one words or phrase. eg.- **I like I like** ice cream.

In the case of repetitions, there can also be fluent repetitions that are spoken on purpose. Often, two occurrences of a word are required to convey the intended meaning of an utterance, for example, while strongly stating a view (It's **very very** important). In these cases, the speaker repeats an utterance for a purpose. It is necessary to differentiate fluent repetitions from disfluent ones. One of the properties of disfluent repetitions is that they are often accompanied by a pause or silence region, and there is a repetition of both pitch level and repeated unit. Disfluent repetitions can be distinguished from fluent ones with the use of prosodic information.

In our work, we have taken both fluent and disfluent repetitions in a single class. If we detect a repetition, it can be classified as fluent or disfluent by further processing using prosodic information; it will be taken as a work for the future.

## 2.3 Cause of Disfluencies in Spontaneous Speech

After having a discussion on major types of disfluencies, one should now follow up to discuss why they occur in spontaneous speech.

It takes time for a speaker to prepare what to say and in what manner before speaking. With all the disfluencies, there is one thing similar, that speaker buys some extra time. There seem to be multiple points where problems can arise as speakers proceed through the complex series of tasks required to produce an utterance, such as planning the overall utterance with meaning, the length and complexity of the message that the speaker wants to convey, etc. Planning of these tasks affects the chance of disfluencies in spontaneous speech. Relatively long and complicated utterances are more liable to disfluencies and repairs, as it becomes cognitively more challenging to keep track of all the tasks.

Hesitations such as filled pauses or prolongations can also occur when the speaker is required to access information from long term memory or if the information is hard to retrieve. Speaker then tries to buy some time by using these disfluencies. example - I was saying that uh ⟨information⟩.

Chances of hesitation also depend on the characteristics of words we are planning to speak. Words that we knew for a much longer time and which are frequently used are easier to access while speaking and thus less prone to hesitations. Because less frequently used words require longer to access, they are much more likely to be accompanied by hesitations.

While planning an utterance, other than remembering the information and deciding what to speak, the speaker always goes through correcting errors while speaking. A speaker might decide on a goal during planning, start executing the plan, and then realise that the strategy isn't appropriate and try to make changes. This will induce some kind of repair type disfluencies in speech.

This brief summary of disfluency in spontaneous speech aims to give the reader a quick rundown of the most common types of disfluencies and an understanding of why they happen.

## 2.4 Detection of Disfluencies in Spontaneous speech

In general, most of the detection methods belong to one of the following categories - as a post-processing step after the ASR output by using only text-based features or along with speech [28, 29, 30] or as a pre-processing step before giving input to a speech system by using signal level processing [49, 31]. Though the ASR-based approaches are effective and give encouraging results, they depend on how accurate the ASR is. In this work, we use only speech-based features for the disfluency detection problem.

The general pipeline for the pre-processing approach of detecting disfluency is discussed in detail below.

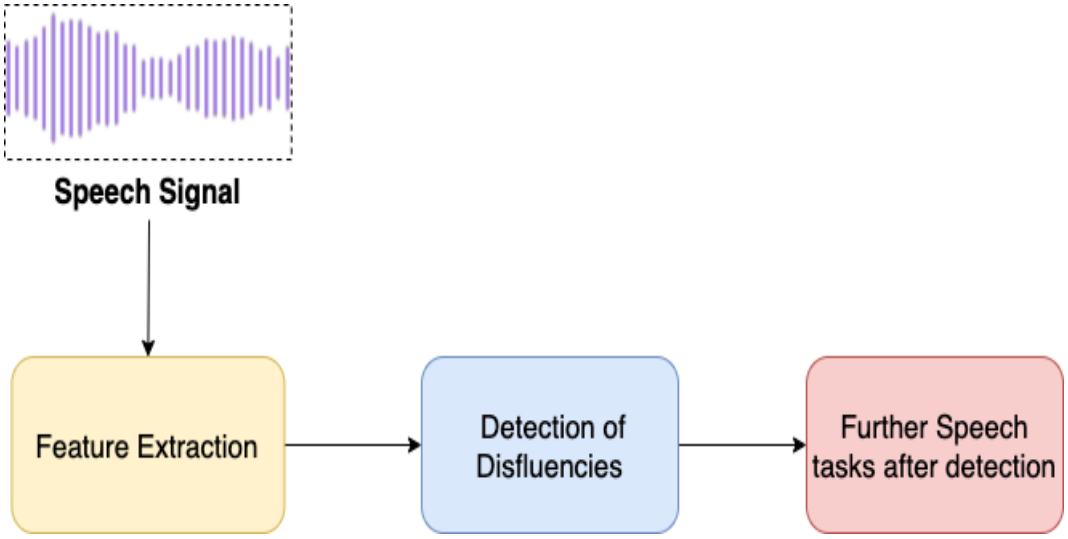


**Figure 2.4** Pre-processing pipeline for disfluency detection

### 2.4.1 Feature Extraction

The ability to represent the input sample with a reduced size while strengthening the distinguishing attributes is a key criterion for a set of features to be valid for any detection or classification task. In the case of speech processing, there are many features contributed by researchers to distinguish the characteristics of different samples of speech. Unlike orthographically, phonetically speech can take various forms, same words or sentences uttered by the same speaker can also have different characteristics. So, it becomes crucial to decide which feature one wants to use according to the speech recognition task. For example, if we want to know about the tone or emotion of the speech, then we'll go towards features that capture long term characteristics of speech. If one wants to use features for the Automatic Speech Recognition(ASR) task, then we'll try to go with segmental features such as MFCC, LPCC, PLP, etc.

In our case of disfluency detection, where we detect the timestamp of disfluencies in an utterance, we have to do framewise analysis, which means to detect if a particular frame is disfluent or not. As frames take up a very short duration of utterance (10ms in our case), and their properties are easier to capture using segmental features, we try to use good segmental features such as MFCC, log-Mel filterbank, etc. which can also be combined with prosodic features to further improve the detection performance.

### 2.4.2 Classfication

In simple words, classification means labelling input data with a class. For example, labelling a speech frame as fluent or disfluent is an example of classification. These classes are sometimes referred

to as labels, targets, etc. Classification is an example of supervised learning, where labels are also provided with input data so that one can train our system for the respective problem. Similar to features, there are numerous classification algorithms also available today, and it is impossible to determine which one is superior to the other. It is dependent on the type of problem we are aiming to solve and the nature of the data supplied.

As the speech domain is very vast, there are different classification methods used for different problems. For example, if we want to predict the next words in our utterance, then sequence prediction algorithms such as RNN, LSTM becomes quite useful. In our case of disfluency detection, where we want to predict if a particular frame is difluent or not, we use binary classifiers. Starting from basic Liner SVM to Multi headed attention, many machine learning techniques have been tested in this work.

We can break our task of disfluency detection in speech signal into classification of speech frames as disfluent or not. If we are able to classify each frame of a speech signal as fluent or disfluent, then all the speech frames can easily be used to get the timestamps of disfluencies in signal.

## 2.5 Performance metrics used in this work

The disfluency detection task can be rephrased as classifying if a particular frame is disfluent or not in a speech signal. We can acquire the estimated timestamp of the disfluency in a signal by labelling frames disfluent or fluent. In this work, we have used three important metrics that are used widely for classification tasks. These metrics are discussed briefly below.

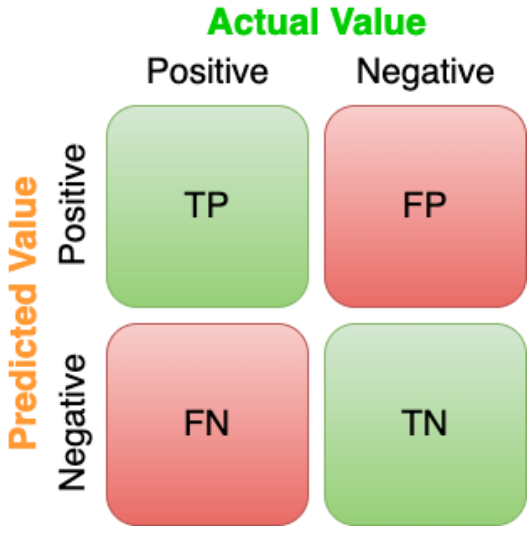


**Figure 2.5** Confusion matrix

### 2.5.1 Accuracy Score

The number of correct predictions divided by the total no. of test samples is the Accuracy. It's a straightforward measurement with a wide range of applications due to its simplicity. From the confusion matrix in Figure 2.5, one can derive the formula for Accuracy as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy used alone is not the best metric for predicting the performance of a classifier. For example, in a situation with a significant class imbalance, a model can predict the value of the majority class for all predictions and obtain high classification accuracy; but, this model is useless in the issue domain. In other terms, Accuracy can still be higher even if all the negative classes are predicted positive in the case of class imbalance. Thus, Accuracy is usually combined with other metrics such as precision, recall, F1 Score to check the authenticity of predictions.

### 2.5.2 Precision

The ratio of correctly predicted positive observations to the total predicted positive observations is known as Precision. It helps in keeping track of how many false positives have been introduced in prediction. To understand it more intuitively, we take the example of predicting a frame as disfluent or not. False positive in this case will be, predicting a fluent frame as disfluent. More no. of false positives is not acceptable for our problem. So to check this, Precision metric is used. From the confusion matrix in Figure 2.5, one can derive the formula for Precision as:

$$Precision = \frac{TP}{TP + FP}$$

### 2.5.3 Recall

The ratio of correctly predicted positive observations to total expected positive observations is known as Recall. It helps in keeping track of false negatives or the number of positive observations that our model failed to predict. Again taking an example of predicting a frame as disfluent or not. False negative, in this case, will be, predicting a disfluent frame as fluent. If Recall is low, then we are missing most of the disfluent frames in a speech utterance. From the confusion matrix in Figure 2.5, one can derive the formula for Recall as:

$$Recall = \frac{TP}{TP + FN}$$

### 2.5.4 F1 Score

Precision takes into account false positives, and recall takes into account false negatives. To get a general idea of both false positive and false negative predicted by the model, we use a metric called F1

Score. F1 score is needed when we want to find a balance between recall and precision. It is represented by the harmonic mean of the model's precision and recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

## 2.6 Recent works on Disfluency detection: Literature Review

### 2.6.1 Databases for disfluencies

For doing any classification or prediction task, a database is needed to extract features of different samples and then train on them.

The Switchboard speech corpora [18] has been used for studying disfluencies in spontaneous conversational speech. It contains 2400 two-sided telephonic conversations in American English, annotated with disfluency events in the transcripts. Speech recordings from informal presentations were the setting for the COPE American English corpus [**?**] where 227-minutes of speech was annotated to study the properties of fillers and their correlation with clause properties.

Speech disfluencies have been the subject of research in languages other than English as well. In [36], read speech data from children was collected in Portuguese, and 9 disfluency events were annotated. The HESITA speech corpus was introduced in [7]. Disfluency events like filled pause, repetitions, substitutions, vocalic extensions and truncated words were identified and annotated in speech recordings from 30 daily news programs in European Portuguese. In [42], segment prolongations were studied in the Hebrew language from spontaneous speech recordings collected from 36 speakers and amounting to 97 minutes. Phonetic and structural properties of segment prolongations in German were studied in [4] using data from 18 speakers, amounting to 4 hours.

In [23], the UCLASS dataset was introduced. This dataset is in British English and deals with disfluencies present in stuttered speech. This dataset is used in our proposed work to develop a pre-trained model on stuttered speech and then predict disfluencies in spontaneous speech. More detail on this dataset will be provided in Chapter 4.

### 2.6.2 Detection of disfluencies in spontaneous speech

Automatic disfluency detection has been the focus of many works [28, 29, 47, 39]. In general, most of the speech disfluency detection methods belong to one of the following categories - as a post-processing step after the ASR output, which use text-based features along with speech for disfluency detection [49, 31] or as a pre-processing step before the ASR. Such methods use only signal-level features to detect disfluencies [39, 22]. In [38], log-energy Mel scale filters and pitch based features were shown to perform well in detecting disfluencies at frame level using SVM and DNN classifiers. Formant information and nasality effect were used as cues in [2, 25] for automatic detection of individual disfluencies. [26] addressed the issue of detecting disfluencies at the utterance level using speech

features. Disfluencies were detected in a 4-second speech file using spectrogram as an input feature to Deep Residual network with Bidirectional LSTM (BiLSTM). Sequence tagging using lexical features is an effective approach to detect disfluencies and has been used in [49, 17]. In [15], disfluency detection based on lexical features was done using a neural machine translation model. In [3], disfluency detection was performed using a noisy training approach, BiLSTM and self-attention model and outperformed the state of the art BERT model [13].

In the context of Indian English, little work has been done for analyzing disfluencies. In [2], a formant-based thresholding system was discussed, where the first 2 formants and duration information was used to detect filled pause in 96 minutes speech in Indian English. A similar method for filled pause and repetition detection was described in [25]. Here, the stability of the first four formants was used to identify disfluencies in the speech signal. A small dataset of 60 English sentences was considered for this work.

## 2.7 Motivation for current work

Going through existing literature, we found that there are a few notable disfluency databases that are used by researchers, but most of them are not available freely and thus hinder research, which also serves as our motivation for developing a freely available disfluency dataset, introduced in Chapter 3. Many of the above-discussed disfluency detection techniques use lexical features for detecting disfluencies in speech. These lexical features, in most cases, are derived from ASR output, which itself can be corrupt due to the presence of disfluencies in training data. That motivated us to build a pre-processing network that can detect and remove disfluencies in speech before passing it to ASR so as to stop propagating errors due to disfluencies to downstream applications.

For training any system to predict a class, we need enough variance in the samples of class so as to learn the differentiating characteristics of that class in a much better way. Keeping that in mind, another common cause of disfluencies is stuttering. Stuttered speech has disfluencies that are similar to those seen in spontaneous speech, but they occur more frequently. This serves as the motivation for applying transfer learning in the current study, in which we try to detect disfluencies in spontaneous speech using stuttering speech data. Literature review for stutter detection system, and more detailed reasoning behind using transfer learning approach is provided in Chapter 4.

## 2.8 Summary

This chapter gives a brief description of major types of disfluencies in spontaneous speech and some insights on factors that cause them. Different methods for detecting disfluencies in speech and the pre-processing method used in this work has been explained in detail. Finally, the work done by fellow researchers in this domain for providing resources and the novel detection methods have been put forward as a literature review.

*Chapter 3*

# Disfluency detection with machine learning

## 3.1 Introduction

As stated in Section 1.2, the detection and removal of disfluencies from speech is a critical task since disfluencies can impair the performance of speech-based applications such as Automatic Speech Recognition (ASR) systems and speech-to-speech translation systems. From the perspective of Indian languages, there is a lack of studies pertaining to speech disfluencies, their types and frequency of occurrence. Also, the resources available to perform such analysis in an Indian context are limited. Hence, the first step toward studying disfluency detection is to create a database to ease the research of disfluencies, followed by the development of baseline automatic disfluency detection systems using a range of machine learning methods on the acquired dataset. In this chapter, we will start by first giving details about our dataset IIITH-IED and then use our aggregated dataset on considered baseline systems and examine the performance variation.

## 3.2 IIITH-Indian English Disfluency(IIITH-IED) Dataset

When looking through the existing literature on speech disfluencies and their detection, it was found that the majority of the study was done for British and American English [37, 48, 1]. With roughly 83 million individuals using English as a second language, India has one of the world's largest English-speaking populations [34]. Hence, studying disfluencies from an Indian English perspective is essential. However, it was observed that existing studies had not paid as much attention to disfluencies in Indian English as the language deserves. In addition, there are insufficient resources to conduct such a study.

To address this issue of the lack of freely available data for disfluency-based studies in Indian English, we introduce the IIITH-IED dataset. Five major types of speech disfluencies were identified and annotated in 10-hours of lecture mode speech in Indian English for the preparation of this dataset. Since the lecturer prepares some critical points of their lecture in advance, but there are instances where the lecturer has to explain a topic spontaneously, we characterize this type of speech as semi-spontaneous. More detail about this dataset is provided in the below sub-sections.

### 3.2.1 Description

This dataset deals with speech disfluencies in Indian English. India has a large English speaking population, with English being used as the primary teaching medium in most higher education institutions. The lectures delivered in these institutions are an excellent source for studying the characteristics and frequency of occurrence of disfluencies since the lecturer is expected to explain a particular topic, sometimes on the go, during these lectures, which leads to the occurrence of disfluencies along with normal speech. So, to prepare this dataset, freely available lectures under the government of India's NPTEL initiative were used. The lectures used for the preparation of this dataset belonged to the following domains - Computer Science, Artificial Intelligence, Electronics and Communication, and Electrical engineering. Speech recordings from 60 speakers were used in the preparation of the IIITH-IED dataset. Out of the 60 speakers, 30 were male, and 30 were female to minimize any gender-based imbalance that might be present. For every speaker, a 10-minute audio recording was taken. This 10-minute audio recording was further split into smaller audio files of length 8 to 12 seconds. For the IIITH-IED dataset, five different types of speech disfluencies were considered. They were - filled pause, prolongations, part-word repetitions, word repetitions and phrase repetitions.

### 3.2.2 Annotation process

For every speaker, a 10-minute audio recording was taken from a lecture to capture the variability in the speaking style and disfluencies that might be present in the speech. This 10-minute audio recording was further split into smaller audio files of length 8 to 12 seconds, with a sampling rate of 16000 Hz. The smaller audio files were segmented by ensuring that the segmentation did not lead to the chopping of words and phrases. Transcripts provided by NPTEL for audio recordings are without disfluency markings, so for word-level marking, two annotators listened to these audio files and marked the positions of disfluencies in the corresponding transcripts. After marking at the transcript level, the annotation was performed on the speech signal to identify the starting time and ending time of each occurrence of disfluency. Audacity open-source toolkit was used to perform this signal level annotation and generate the corresponding labels for each audio file. This timing information in the generated labels was used later to develop frame-level automatic disfluency detection systems on the IIITH-IED dataset. Fig 3.1 show an example of the annotation performed here. The annotation format used here is similar to [24]. In order to maintain consistency, all the audio files of one particular speaker were annotated at both the word level and the signal level by the same annotator. The other annotator then verified this annotation, and the final timestamps for each disfluency occurrence were obtained.

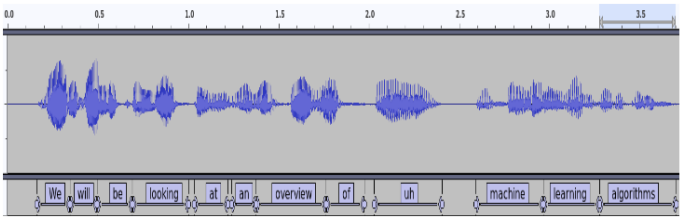


**Figure 3.1** Signal-level Transcription of the following sentence - We will be looking at an overview of uh machine learning algorithms.

### 3.2.3 Statistics

The IIITH-IED dataset includes five different types of speech disfluencies. They are - filled pause, prolongations, part-word repetitions, word repetitions and phrase repetitions. The number of occurrences and the average duration of each type of disfluency present in the IIITH-IED dataset is mentioned in Table 3.1.

**Table 3.1** Number of occurrences of each disfluency type in IIITH-IED Dataset.

| **Disfluency Type** | **# of Occurences** | **Avg. Duration** |
|---|---|---|
| Filled Pause | 1428 | 0.395 sec |
| Prolongation | 71 | 0.553 sec |
| Part-word Repetition | 164 | 0.954 sec |
| Word Repetition | 211 | 0.890 sec |
| Phrase Repetition | 76 | 1.365 sec |

As can be seen, filled pause was the most common type of disfluency occurring in the dataset. It was observed that, like British English, two forms of filled pause, ‘um’ and ‘uh’, were the most common [43]. Out of these two, the number of occurrences of ‘uh’ (1265) were greater than that of ‘um’ (163). There was a gender-based difference here, with the number of ‘um’ filled pause produced by female speakers (106) being much greater than what their male counterparts produced (57). In the case of part-word repetitions, most instances had word-initial repetition in them; that is, the repetition took place at the initial syllable position of the words (for example: w-what). Word repetitions were the second most common type of disfluency in the IIITH-IED dataset. Most instances of word repetitions were for commonly used words like - and, of, the, for, etc. Phrase repetitions and prolongations were the rarest disfluency types, having 76 and 71 occurrences, respectively. The instances of prolongations observed in the dataset were caused due to vowel lengthening. This lengthening occurred either at the word-initial position (for example: lengthening of /o/ in of) or at the word-final position (for example: lengthening of /o/ in to). Middle-word lengthenings leading to prolongation disfluencies were very rare in the dataset.

In total, the IIITH-IED dataset consists of 3386 wavfiles of duration 8-12 seconds. Each wavfile has zero, one, two or more disfluencies in it. Table 3.2 presents the statistics about the number of audio files having a certain number of disfluencies in them.

**Table 3.2** Number of wavfiles corresponding to the number of disfluencies in IIITH-IED Dataset

| **# of Disfluencies** | **Number of wavfiles** |
|---|---|
| Zero | 2013 |
| One | 934 |
| Two | 326 |
| Three | 91 |
| More than three | 22 |

## 3.3 Baseline Systems

### 3.3.1 Linear Support Vector Machine (SVM)

Classification tasks with two class labels are referred to as binary classification. Generalizing the binary classification problem results in a linear equation of the type :

$$y(x) = wT\phi(x) + b$$

where W is the weight vector for each feature, $\phi(x)$ denotes a fixed feature-space transformation, and b denotes the bias. If our data is linearly separable, solving this equation gives us the decision boundary or hyperplane that splits the data into two distinct classes, i.e. samples with y(x) ¿ 0 belong to one class, and samples with y(x) ¡ 0 belong to the other, with y(x) = 0 being the hyperplane.

Certainly, there is a chance that numerous such hyperplanes exist that can precisely divide the classes; therefore, the ideal strategy would be to locate the hyperplane with the minimum generalisation error. This is exactly how the Support Vector Machine works. The notion of margin, which is the shortest distance between the decision surface that divides the two classes and any of the samples, is used by the SVM classification model to accomplish this minimal generalisation error. This decision boundary is selected in SVM as the one that maximises the margin. As a result, the samples closest to the decision boundary are the only ones required to determine the decision boundary's equation. These samples that lie on margin boundaries are known as support vectors, from which SVMs also derive their name.

### 3.3.2 Random Forest Ensemble Method

A random forest is an ensemble method based supervised ML algorithm that is used to solve regression and classification problems. A random forest operated by constructing numerous decision trees and then training all of them. The output of the algorithm is the class selected by the majority number of trees. These generally outperform decision trees in terms of accuracy.

It uses techniques like bagging while building individual trees. Different features are used to build each individual tree so that the final output predicted by all of them is more accurate than any that can be predicted by an individual tree. These forests are good in the sense that they nearly have the same hyperparameters as a decision tree or any bagging classifier. Random forests work well with high-dimensional data as only a subset of the original features are chosen to train each individual tree. This property makes it easy for them to work with hundreds of features. Working with a subset of the whole features also helps in bringing down the training time as a whole when the training for each tree is launched in parallel.

### 3.3.3 Deep Neural Network (DNN)

A Deep Neural Network ( DNN ) is an Artificial Neural Network ( ANN ) that contains multiple layers of computing units (usually convolution units) between the input and output layers. All DNN's consist of the following components: neurons, synapses, weights, biases, and functions. DNN perform very well and give very high accuracy when trained on large amounts of data. These networks are trained using supervised machine learning techniques and using methods like backpropagation to compute the weights of the network. Once trained on a given dataset, these models perform with very high accuracy, especially for classification problems. DNN's have the ability to find and model complex non-linear relationships between a given set of features, making them very successful in a variety of applications in today's world.

## 3.4 Experimental setup for baseline systems

The IIITH-IED dataset was used to develop disfluency detection systems in Indian English. The systems developed here are frame-level automatic disfluency detection systems. The models employed are entirely based on acoustic characteristics, i.e., rely only on acoustic features, i.e., the information extracted from the speech signal for disfluency detection. The aim of developing such systems was to detect whether a particular type of disfluency was present in a 10 ms frame of speech or not, as done in [38, 35]. Experiments were performed here for the 5 types of disfluencies making up the IIITH-IED dataset - filled pause, part-word repetitions, word-repetitions, phrase repetitions and prolongation. The detection of every type of disfluency was set up as a binary classification task - the speech frame either belong to that disfluency type or does not. Another set of experiments were performed to decide whether a particular speech frame is disfluent or not. In this experiment, speech frames with all sorts

of disfluency were classified into one class, whereas normal speech frames with no disfluency were classified into another. Binary classification was then performed to decide whether a speech frame is fluent or disfluent.

### 3.4.1 Feature extraction

Two different sets of acoustic features were used for the task of disfluency detection directly from the speech signal. In [38], log Mel-filterbank features were used for disfluency detection and produced high levels of accuracy. Hence, the first set of features used here was a combination of log Mel-filterbank features and the fundamental frequency calculated for each frame. 40 log Mel-filterbank features were extracted by taking a 25 ms frame of the speech signal with a 10 ms shift. The filterbank features were then mean-variance normalized for every audio file, using the VAD information. Besides this, the signal's fundamental frequency in that frame was calculated and used along with the filterbank features, giving a feature vector having 41-dimensions per frame. This set of features will be referred to as the Filterbank features from here on.

The next set of features used here were the MFCC input features. The MFCC features were made up of 13 cepstral coefficients per frame, the 0th cepstral coefficient and the energy of the frame. The delta and delta-delta coefficients were also computed for the MFCC features, giving a 45-dimensional feature vector per frame. Windowed frames of length 25 ms and frame shift of 10 ms were used for computing the MFCC features.

In [35], stacking up features from frames using a context window was shown to produce better detection results. So, for both sets of features here, window lengths of $\pm 1$, $\pm 2$ and $\pm 3$ frames were experimented with. Table 3.3 shows the dimensions of the final feature vectors for every frame obtained using these window lengths.

**Table 3.3** Feature vector dimension per frame for different window sizes

| **Features** | **Dimensions for window sizes** | | | |
|---|---|---|---|---|
| | **0** | **$\pm 1$** | **$\pm 2$** | **$\pm 3$** |
| **Filterbank features** | 41 | 123 | 205 | 287 |
| **MFCC + delta + delta-delta** | 45 | 135 | 225 | 315 |

It was observed that stacking up features from 3 frames before and after every particular frame gave the best classification results. Hence, this configuration was used for reporting the final disfluency detection results.

### 3.4.2 Detection models

Following models were used for disfluency detection task separately, and results were noted for each of them:

- SVM with a linear kernel and penalty term (C) of 0.25 was used for the binary classification task. C effectively sets the tolerance for error in the training data; it is a regularisation parameter that determines the trade-off between attaining a low training error and achieving a low testing error, which refers to your classifier's potential to generalise to new data. In layman terms, a small value of C corresponds to low bias and high variance, whereas the high value of C corresponds to high bias and low variance. We experimented with multiple values of C, and at 0.25, we got the highest performance. Thus, results are reported with C=0.25.

- The Random Forest based ensemble model was taken, with the maximum number of trees being 100 and the maximum depth of every tree being 20 in the ensemble.

- DNN classifier with 2 hidden layers was considered for disfluency detection. The number of hidden units in each layer were 100 and 50, respectively, with sigmoid activation function after each layer. The optimizer used was Adam optimizer. Hyperparameter tuning was performed as well for training the DNN. An optimal learning rate of 0.001 and an optimal batch size of 32 was used here. The DNN architecture used in this work is described in Figure 3.2.

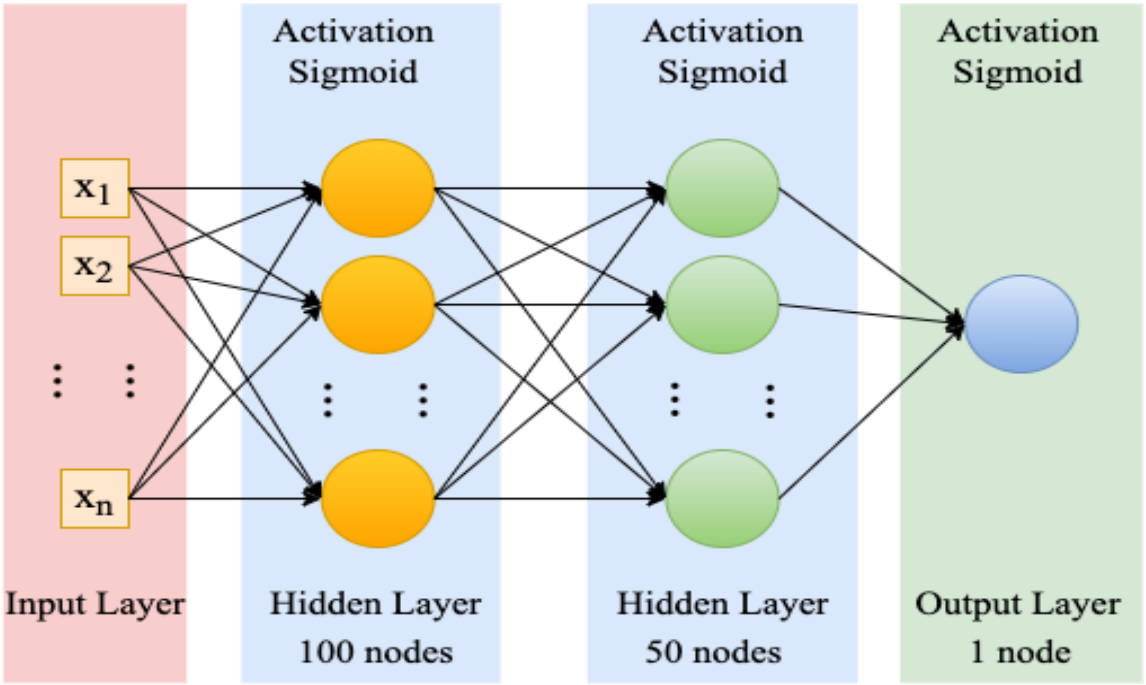


**Figure 3.2** DNN Architecture used in the work.

A train-test split of 80:20 was used for the experiments here. An important point considered was that disfluency datasets tend to be very imbalanced since the number of frames corresponding to disfluencies are much less than fluent speech frames. So, in order to ensure that the disfluency detection models are not biased, random undersampling [27], and SMOTE (Synthetic Minority Oversampling Technique) based oversampling [8] techniques were used while training each model. For random undersampling, a large number of frames belonging to the majority class were removed randomly so that the number of frames of both the classes is equal. On the other hand, using SMOTE, the number of samples of the minority class were made equal to that of the majority class by generating new samples of the minority class using the k-nearest neighbour approach.

**Table 3.4** Results obtained for individual disfluency detection using Filterbank features. Here Acc. shows Accuracy, Rec. shows Recall and F1 shows F1-Score

| **Models** | **Filled Pause** | | | **Prolongations** | | | **Part-Word Repetitions** | | | **Word Repetitions** | | | **Phrase Repetitions** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** |
| **SVM (undersampled)** | 87.12 | 0.842 | 0.866 | 82.77 | 0.828 | 0.821 | 68.23 | 0.621 | 0.638 | 60.94 | 0.547 | 0.585 | 62.95 | 0.563 | 0.615 |
| **RF (undersampled)** | 89.28 | 0.887 | 0.890 | 95.21 | 0.942 | 0.950 | 81.22 | 0.803 | 0.812 | 82.01 | 0.824 | 0.815 | 78.90 | 0.761 | 0.758 |
| **DNN (undersampled)** | 85.83 | 0.866 | 0.859 | 74.65 | 0.693 | 0.712 | 66.43 | 0.610 | 0.632 | 57.88 | 0.556 | 0.535 | 60.85 | 0.531 | 0.594 |
| **SVM (SMOTE)** | 88.91 | 0.902 | 0.892 | 86.24 | 0.848 | 0.865 | 74.14 | 0.773 | 0.797 | 70.12 | 0.686 | 0.690 | 64.58 | 0.580 | 0.626 |
| **RF (SMOTE)** | 91.32 | 0.944 | 0.938 | 95.87 | 0.955 | 0.951 | 84.64 | 0.812 | 0.835 | 86.33 | 0.881 | 0.852 | 79.78 | 0.773 | 0.767 |
| **DNN (SMOTE)** | 86.18 | 0.901 | 0.878 | 81.30 | 0.799 | 0.824 | 79.71 | 0.815 | 0.833 | 65.82 | 0.704 | 0.728 | 64.79 | 0.664 | 0.672 |

**Table 3.5** Results obtained for individual disfluency detection using MFCC features. Here Acc. shows Accuracy, Rec. shows the Recall and F1 shows the F1-Score

| **Models** | **Filled Pause** | | | **Prolongations** | | | **Part-Word Repetitions** | | | **Word Repetitions** | | | **Phrase Repetitions** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** |
| **SVM (undersampled)** | 89.91 | 0.882 | 0.895 | 84.22 | 0.856 | 0.845 | 71.23 | 0.709 | 0.698 | 72.44 | 0.702 | 0.713 | 66.12 | 0.654 | 0.662 |
| **RF (undersampled)** | 89.14 | 0.892 | 0.889 | 94.64 | 0.932 | 0.945 | 82.86 | 0.838 | 0.826 | 82.53 | 0.824 | 0.821 | 79.03 | 0.773 | 0.785 |
| **DNN (undersampled)** | 88.97 | 0.831 | 0.828 | 80.95 | 0.812 | 0.809 | 77.41 | 0.757 | 0.775 | 75.13 | 0.763 | 0.748 | 71.33 | 0.662 | 0.700 |
| **SVM (SMOTE)** | 90.12 | 0.887 | 0.892 | 86.24 | 0.864 | 0.851 | 73.41 | 0.713 | 0.728 | 73.35 | 0.732 | 0.724 | 69.63 | 0.681 | 0.702 |
| **RF (SMOTE)** | 93.84 | 0.936 | 0.937 | 98.21 | 0.978 | 0.979 | 92.95 | 0.900 | 0.927 | 93.32 | 0.913 | 0.931 | 81.02 | 0.791 | 0.814 |
| **DNN (SMOTE)** | 89.94 | 0.894 | 0.898 | 88.35 | 0.891 | 0.883 | 82.33 | 0.809 | 0.819 | 80.86 | 0.799 | 0.803 | 73.52 | 0.722 | 0.736 |

## 3.5 Results and discussion for baseline systems

The first set of results presented here are for the detection of a single disfluency class. TABLE 3.4 shows the results obtained for filled pause, prolongations, part-word repetition, word repetition and phrase repetition detection using Filterbank features. The metrics chosen for the evaluation of classification models here are - accuracy, recall and F1-score. For filled pause, an F1-score of 0.938 was obtained with the Random Forest classifier in the oversampling condition. For prolongation detection, a high recall (0.955) and high F1-score (0.951) was obtained. There was an improvement in the detection accuracy using all three classifiers in the oversampling approach compared to undersampling, which shows that further improvements can be obtained by using the same experimental setup with more extensive datasets so that the variance in samples of each disfluency can be captured effectively. As compared to filled pause and prolongation, the accuracy and F1-score obtained for repetition type disfluencies was lesser. This can be attributed to the longer average duration of these types of disfluencies, which provides scope for greater variance in the samples of these classes. Thus, the best results for part-word and word repetitions are obtained using oversampling, with the best F1-scores obtained being 0.835 for part-word repetition and 0.852 for word repetition. In case of phrase repetitions, a lower recall is obtained in all the experiments. This can be because the number of samples of phrase repetition in the dataset are not enough to efficiently model this class. The highest F1-score obtained for phrase repetition detection was 0.767.

**Table 3.6** Results obtained considering all disfluencies as single class

| **Models** | **Fiterbank + F0** | | | **MFCC** | | |
|---|---|---|---|---|---|---|
| | **Acc.** | **Rec.** | **F1** | **Acc.** | **Rec.** | **F1** |
| **SVM (undersampled)** | 73.28 | 0.755 | 0.743 | 80.34 | 0.805 | 0.812 |
| **RF (undersampled)** | 85.26 | 0.825 | 0.847 | 86.69 | 0.848 | 0.863 |
| **DNN (undersampled)** | 72.86 | 0.768 | 0.738 | 82.25 | 0.814 | 0.820 |
| **SVM (SMOTE)** | 73.57 | 0.753 | 0.757 | 80.94 | 0.810 | 0.816 |
| **RF (SMOTE)** | 88.69 | 0.904 | 0.883 | 89.61 | 0.862 | 0.891 |
| **DNN (SMOTE)** | 73.11 | 0.787 | 0.770 | 83.26 | 0.811 | 0.828 |

Detection results obtained using MFCC features for individual disfluencies are presented in TABLE 3.5. Compared to Filterbank features, MFCC features give better outcomes for all the five disfluencies considered here. With DNN as a classifier, a definite improvement of 3.76% was observed for filled pause detection in the oversampling condition using MFCC compared to Filterbank features. The best F1-score obtained for filled pause and prolongation using MFCC features was 0.937 and 0.978, respectively. The accuracy obtained for repetition type disfluencies was higher as well in this case. The best detection results are obtained using the Random Forest classifier. For word repetitions, a high recall value of 0.913 and high accuracy of 93.3% was obtained in the oversampling condition. In the case of

part-word repetitions, high variability in the samples of this disfluency leads to lower classification accuracy in the undersampled condition. Further, the oversampling scenario improved the results in terms of accuracy and F1-score of 92.9% and 0.927, respectively. For phrase repetitions, using MFCC features with DNN gave an absolute improvement of 8.73% in accuracy. The highest F1-score for phrase repetition was obtained using the Random Forest classifier and was 0.814.

TABLE 3.6 shows the results obtained when all the disfluencies were considered a single class, and binary classification was performed to determine if a speech frame was disfluent or not. In this case, as well, MFCC features outperformed Filterbank features in terms of F1-Score, accuracy and recall.

In the oversampling condition, an absolute difference of 10.15% was observed in the accuracy of the DNN classifier using MFCC features compared to Filterbank features. The best results are obtained with Random Forest classifier using Filterbank and MFCC features in terms of F1-scores of 0.883 and 0.891, respectively in oversampling condition. In addition, high recall values are also obtained using these features with DNN and Random Forest classifier, showing the ability of these models to minimize the number of false alarms obtained in classification.

## 3.6 Summary

In this chapter, we started by giving details about the IIITH-IED dataset for disfluencies in spontaneous speech. Elaboration of annotation process and discussion about disfluencies statistics are followed up. Then, we use the IIITH-IED dataset to develop baseline frame-level disfluency detection systems and investigate their respective performances. For the aforementioned purpose, 3 critical machine learning architectures have been used, namely, Support Vector Machine, Random Forest, and Deep Neural Networks. Experiments were performed on 2 feature sets, MFCC features and log Mel-filterbank features, out of which MFCC features outperformed filterbank features. Thus, we'll be using MFCC features for performing future experiments. Other deep learning architectures will be explored in the next chapter, followed up by our proposed transfer learning method to detect disfluencies in spontaneous speech.

*Chapter 4*

# Transfer Learning based Disfluency Detection

## 4.1 Introduction

In the previous chapter, we have developed baseline systems for disfluency detection in spontaneous speech using the IIITH-IED dataset. Stuttered speech is another primary source of disfluencies. The disfluencies in stuttered speech are similar to those present in conversational speech, but their frequency of occurrence is higher. This serves as the motivation to use transfer learning in the proposed method - using stuttered speech data, we try to detect disfluencies in conversational, lecture-mode speech. In this chapter, we will start by providing a brief overview of stuttering and the nature of disfluencies in stuttered speech. Following that, we'll provide motivation of using transfer learning as an approach to detect disfluencies in spontaneous speech, using stuttered speech. And then use pre-trained models on stuttered speech to detect disfluencies present in the IIITH-IED dataset using the transfer learning approach.

## 4.2 Stuttering

Stuttering is a speech disorder where hesitations that break the flow of speech occur involuntarily [21]. Audible or inaudible prolongation of words, excessive use of fillers (like 'um', 'uh', etc.) and uncontrolled repetitions are some of the main characteristics of stuttered speech. In humans, both physiological and neurogenic causes can lead to stuttering, with some of the reasons being - increased dopamine levels in muscles causing inhibitory effects, problems with auditory processing, and motor disorders pertaining to the basal ganglia [5]. From the perspective of speech, stuttering is sub-divided into various forms, depending on the type of hesitation, which led to the stutter. Some of the different forms of stutter are - interjections (fillers), prolongations, repetitions and revisions.

### 4.2.1 Nature of disfluencies in Stuttered Speech

In Section 2.3, we've related disfluencies in spontaneous speech to issues with the speaker's ability to buy time during the conception and formulation stages of utterance. But speakers who stutter report that for them issue is not at the conception and formulation stage but at the later stage while executing the articulation plan. The issue stems from the transition from one sound to the next sound due to difficulty in motor programmes coordination.

Stuttering is often accompanied by physical tension, which is a significant aspect of stuttered disfluencies. Other than physical tension, the average duration of stuttered disfluencies is also longer, and they occur more frequently as compared to typical disfluencies in spontaneous speech. From the perspective of speech, stuttering is sub-divided into various forms, depending on the type of hesitation, which leads to the stutter. Some of the different forms of stutter are - interjections (fillers), prolongations, repetitions and revisions. In contrast to spontaneous speech, repetitions found in stuttered speech generally contain multiple instances of the repeated unit, i.e., a word or sound can be repeated multiple times. Prolongation in spontaneous speech is typically only for a fraction of a second, whereas with stuttering, it can last more to a second. Thus, we can say that disfluencies in stuttered speech are not exactly same as those found in spontaneous speech, but rather an extreme case of those.

### 4.2.2 Automatic stutter classification

Various works have been done on automatic stutter classification. Early works focused on extracting acoustic, and signal level features from input audio combined with classical machine learning methods such as GMM, LDA, k-NN, etc. [10, 9, 46]. In [10], Mel Frequency Cepstral Coefficients (MFCC) features were used with LDA and k-NN as classifiers to detect repetition and prolongations in stuttered speech. In [9], the same authors used LPCC based features with LDA and k-NN to detect repetitions and prolongations. But LPCC takes an average of 3.73 sec more than MFCC for giving a decision. To better understand the human perception of speech, both spectral and temporal characteristics of the signal are important. With MFCC, short-term spectral features of speech are captured effectively, but the information about temporal behaviour is not perceived considerably. So, to capture the temporal, instantaneous amplitude and frequency characteristics of signals, in [32], LP-Hilbert Envelope Based MFCC features were used for the detection of prolongations, repetitions and interjections.

In recent studies, deep learning architectures are being used with both text and signal level features for the detection of stuttering events [1, 40, 40]. In [1], a lightly supervised approach was used with task-oriented lattices to recognise stuttering events in children's speech and provide a complete verbatim output of stuttered speech to help diagnose the disorder. In [40], the Sequence labelling approach was employed with Conditional Random Fields (CRF) and BiLSTM for the detection of stuttering events in both manual and automatically generated transcripts (by ASR). In [26], spectrogram features were used with a Deep residual network and BiLSTM to classify a 4-second stutter file into one of the six types of major stuttered disfluencies.

Some recent works have also utilised non-speech related features to detect stuttering events [45, 12]. In [45], based on respiratory biosignal activity, stuttering events were classified into blocks and non-block states of speech using Multi-Layer Perceptron (MLP). In [12], with Artificial Intelligence (AI) aided Convolutional Neural Network (CNN) and facial movement patterns, expected speech is classified as fluent or stuttered.

## 4.3 Transfer Learning Approach - Motivation

On exploring the literature pertaining to stuttered speech and disfluencies in spontaneous speech, we found that the acoustic and linguistic basis by which different forms of stutters are categorised is very similar to the categorisation of disfluencies. Even the set of audio features used for automatic stutter classification and disfluency detection are also overlapping. Also, as summarised in section 4.2.1, one can say that disfluencies in stuttered speech are an extreme case of the typical disfluencies. This observation has led to the hypothesis used in our work that a classifier trained on stuttered data can help in detecting disfluencies in conversational speech. Hence, a transfer learning approach to detect disfluencies is explored.

## 4.4 Deep learning Architectures

Three different deep learning architectures are used for testing the stated hypothesis. First architecture is Deep Neural Network (DNN), which has been explained in Section 3.3. Two other architectures which are to be used are BiLSTM and Multi-head attention based architectures, which are explained in some detail in the below subsections.

### 4.4.1 Bidirectional LSTM

Bi-directional LSTM's are sequence processing models and are an extension of LSTM's. Instead of one, these consist of two LSTM's. Where data from all the time steps for a problem is available, one of the LSTM takes the input in the forward direction while the other takes the input in the backward direction. Bi-LSTM's duplicate the Recurrent neural network ( RNN ) processing chain and can improve the performance of the model in sequence classification problems in comparison to LSTM's. They enable additional training by going through the input data twice in two different directions.

### 4.4.2 Multi-head Attention

In Layman terms, Multi-head Attention as name suggests is a module that goes through multiple computations of attention mechanism [44] in parallel (Each of these is called an Attention Head), and after that, the independent attention outputs are combined and linearly transformed into the dimension

we need for our network. The Attention module separates its Query, Key, and Value arguments N times and sends each part to a different Head. The results of all of these Attention calculations are then pooled to get a final Attention score.

## 4.5 Experimental Setup

In order to test our hypothesis, stutter data from the UCLASS corpus is used in experiments here. A disfluency detection model is trained on data from the UCLASS corpus, and the trained model is then used to detect related disfluencies in the IIITH-IED dataset. Figure 4.5 shows the disfluency detection system using transfer learning used here. Details about the UCLASS and IIITH-IED datasets are presented in the next subsections, followed by a description of the DNN, Bi-LSTM and attention based frame-level disfluency detection models.

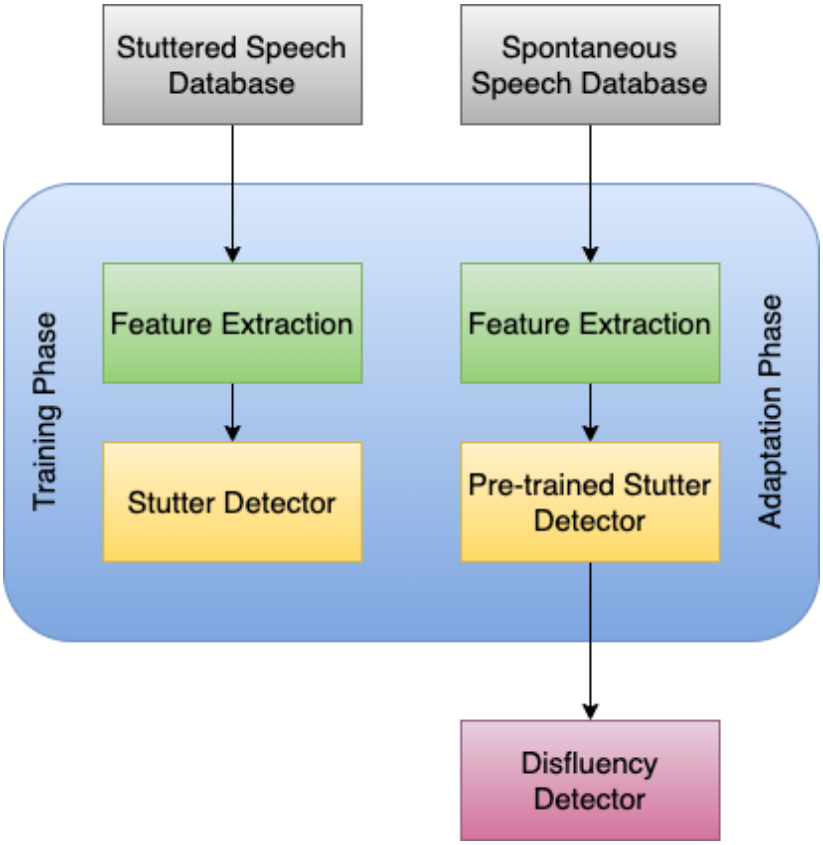


**Figure 4.1** Transfer Learning based disfluency detection pipeline

### 4.5.1 UCLASS Dataset

The University College London's Archive of Stuttered Speech (UCLASS) Dataset is one of the most popular resources for studies on stuttered speech. It was introduced in [23]. The dataset consists of stuttered speech recordings and corresponding annotations in British English. The UCLASS dataset has two main releases. Here we have used speech recordings from Release One of the UCLASS dataset to prepare the pre-trained models for transfer learning experiments. This dataset consists of monologue speech recordings from children of age 8 to 18 years, who were diagnosed with stuttering disorder of varying severity. Out of the 139 recordings available, 25 were used because of the availability of corresponding transcriptions. The transcriptions were forced aligned with the audio to generate timestamps for each word. Each recording was then annotated for 7 types of stutter disfluencies - filled pause, prolongation, sound repetition, part-word repetition, word repetition, phrase repetition and revision, as done in [26]. The annotation was carried out similar to [24]

### 4.5.2 IIITH-IED Dataset

Ten hours of lecture-mode speech in Indian English were transcribed to prepare this dataset. Speech recordings from the freely available lectures under the NPTEL initiative of the Government of India were used to make this dataset. Since lecture-mode speech is prepared, there are instances where the lecturer has to explain a topic spontaneously, this type of speech is categorized as semi-spontaneous. The IIITH-IED dataset consists of speech from 60 speakers - 30 male and 30 female. A 10-minute recording of a lecture from each speaker is annotated for both words as well as disfluencies manually. Each speech recording from a speaker is further segmented into speech files of length 8 to 12 seconds, with a sampling rate of 16000 Hz. After segmentation, annotation is also performed at the signal level to identify the starting and ending time of disfluencies present in each segmented file. The number of occurrences of each disfluency type used from this dataset in our experiments is shown in Table 4.1. More details about the dataset can be found in 3

**Table 4.1** Number of occurrences of each disfluency type in IIITH-IED Dataset

| **Disfluency Type** | **# of Occurences** |
|---|---|
| Filled Pause | 1428 |
| Prolongation | 71 |
| Part-word Repetition | 164 |
| Word Repetition | 211 |

### 4.5.3 Detection Model

In order to test the transfer learning hypothesis, frame-level automatic disfluency detection systems are used. These systems are used to detect whether or not a particular disfluency type is present in a

speech frame of 10 ms. The disfluencies considered for the experiments here are - filled pause, prolongation, part-word repetition and word repetition. For every disfluency type, detection was set up as a binary classification problem - a speech frame either belongs to the disfluency type or it does not.

The features used for the task of disfluency detection are MFCC features. The MFCC features consisted of the following - the first 13 cepstral coefficients, the 0th cepstral coefficient and the energy of the frame. Windowed speech frames having a length of 25 ms, with a 10 ms frame shift are used for MFCC feature extraction here. The delta and delta-delta MFCC coefficients are also computed and used. The size of the feature vector obtained then was 45-dimensional per frame.

To produce better disfluency detection results, stacking up features from neighbouring frames using a context window is proven beneficial in [35, 38]. So, window lengths of ±1, ±2 and ±3 frames were used to experiment with the MFCC features extracted above. The dimensions of the final feature vectors for each frame obtained using ±1, ±2 and ±3 frames window lengths were 135, 225 and 315, respectively. As best classification results were obtained by stacking features from 3 frames before and after each individual frame, final disfluency detection results are reported using this configuration only.

Three disfluency detection systems are then trained using the MFCC features extracted. The first system is a DNN-based detector. The network used here has 2 hidden layers. The number of hidden units in the layers are 50 and 100 respectively. The next system is a BiLSTM-based detector. Two bidirectional recurrent layers, each having 7 units are then used to learn temporal dependency between features and then classify whether the disfluency type is present or not. Dropout rates for both the recurrent layers are set at 0.2 and 0.4 to avoid overfitting in the BiLSTM. The last system used for detection is Multi-Head attention based model, in which a multi-head module with number of attention heads as 3, is used before two bidirectional recurrent layers, each having 7 units. Dropout rates are taken same as in BiLSTM based model.

While performing baseline experiments in Chapter 3, we found out that there was a significant improvement while use SMOTE based oversampling approach, when compared to undersampling approach. So, all the transfer learning based experiments are performed with oversampling approach to get the best possible results.

## 4.6 Experiments and Results

Disfluency detection for each of the four disfluencies was setup as a binary classification task. The baseline disfluency detection systems were developed here on the IIITH-IED dataset using 3 deep learning architectures namely DNN, BiLSTM, and attention based as described in previous section. Four binary classifiers were trained for each of the architectures to detect the four disfluencies. The baseline detection accuracy and F1-score obtained on the UCLASS Dataset are shown in Table 4.3 and the results for the IIITH-IED dataset for the four disfluencies are shown in Table 4.4. For all 3 classifiers, a learning rate of $10^{-3}$ was used for training, with the binary cross-entropy loss function and RMSprop optimizer. The number of training epochs used for DNN were 50, while for the BiLSTM-classifiers, the

number of training epochs used were 10, and for multi-head attention based 15 epochs were taken. For every 10 ms speech frame, the MFCC features extracted for that frame were used as input to the models, which then predicted whether that speech frame belongs to the disfluency type or not. *stratified K-fold cross-validation* was used for splitting data in train and test sets so that the ratio of samples from each class is the same in train and test sets. Here, the value of K was set to 10, and 9-folds were used for training, and 1 fold was used for testing the model.

**Table 4.2** Cosine Similarity between a stutter type and the closest related disfluency

| **Disfluency Type** | **Stutter type** | **Cosine Similarity** |
|---|---|---|
| Filled Pause | Filler | 4.23e-2 |
| Prolongations | Prolongation | 3.19e-2 |
| Part-word Repetitions | Sound Repetition | 5.76e-3 |
| Word Repetitions | Word Repetition | 8.55e-4 |

Further, the proposed transfer learning based disfluency detection systems were trained using the UCLASS corpus and the IIITH-IED dataset. Transfer Learning refers to the learning in a target domain by transferring knowledge from another related task [19]. In this approach, we trained the disfluency detection models to detect a particular type of disfluency using the UCLASS corpus and then validated this model for the related disfluency in the IIITH-IED dataset. In order to find how closely related the occurrences of each disfluency type are in the UCLASS and IIITH-IED datasets, the cosine similarity metric was used. Table 4.2 shows the cosine similarity values obtained for each of the pairs. The cosine similarity is calculated by first taking the dot product (similarity measure) between the frame-level features for each pair of frames. The final value is obtained by averaging across all frames. As can be seen from the table, a close correspondence is found between disfluencies in the two datasets, indicating that using stuttered speech data might help in the task of disfluency detection.

While testing, the pre-trained models developed on the UCLASS dataset are evaluated on the IIITH-IED dataset using stratified 10-fold cross-validation so that the network has some amount of learning experience on our data as well, and uniformity is maintained. The learning rate, loss function and the optimizer used in the experiments are the same as in the baseline experiments. A batch size of 32 was taken while training. The performance of the proposed transfer learning based detection systems for all the disfluency types is shown in Table 4.6. The average accuracy and average F1-score obtained across all folds for each disfluency were used as the metrics to compare the performance.

As can be seen from Tables 4.4 and 4.6, using the proposed transfer learning approach, an increase in detection accuracy and F1-score was obtained for all four types of disfluencies, with both the classification methods, i.e. DNN based and BiLSTM based. Especially for filled pause and prolongation disfluencies, the increase in performance is significant. In the filled pause case, the detection accuracy obtained using transfer learning was 94.96% using the Multi-head attention model, with the F1-score being 0.937. An absolute increase of 2.10% is obtained for the attention classifier using the transfer

**Table 4.3** Baseline disfluency detection results for the four types of disfluencies in the UCLASS Dataset. Here F1 refers to the F1-score.

| | DNN | | BiLSTM | | MHA | |
|---|---|---|---|---|---|---|
| **Disfluency Type** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** |
| Filler | 91.37 | 0.912 | 92.54 | 0.924 | 93.92 | 0.932 |
| Prolongations | 89.91 | 0.900 | 91.82 | 0.919 | 96.89 | 0.966 |
| Sound Repetitions | 84.70 | 0.846 | 80.93 | 0.809 | 88.34 | 0.885 |
| Word Repetitions | 85.21 | 0.851 | 83.41 | 0.832 | 87.98 | 0.881 |

**Table 4.4** Baseline disfluency detection results for the four types of disfluencies in the IIITH-IED Dataset. Here F1 refers to the F1-score.

| | DNN | | BiLSTM | | MHA | |
|---|---|---|---|---|---|---|
| **Disfluency Type** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** |
| Filled Pause | 89.92 | 0.892 | 90.17 | 0.891 | 92.86 | 0.923 |
| Prolongations | 88.26 | 0.887 | 91.07 | 0.911 | 95.77 | 0.957 |
| Part-word Repetitions | 82.47 | 0.817 | 78.72 | 0.778 | 87.48 | 0.872 |
| Word Repetitions | 80.73 | 0.805 | 77.97 | 0.774 | 87.45 | 0.867 |

learning approach compared to the baseline result. Also, for all three architectures, an average improvement of 2.11% is seen. This increase can be attributed to the fact that two forms of filled pause are most common in the UCLASS and IIITH-IED datasets - 'um' and 'uh'. The increase in the number of samples of these two types of filled pause in the training set leads to an increase in performance. The highest detection F1-score is obtained for prolongations using the transfer learning method, with 0.969 being the F1-score using the attention-based model. The absolute increase in detection accuracy is also the highest for prolongation when compared to the baseline results. This is because the majority of occurrences of prolongations in both datasets correspond to the lengthening of vowels ( especially vowels /o/ and /a/). Also, the number of occurrences of prolongation in the UCLASS dataset are a lot more than the IIITH-IED dataset, which leads to significant improvements using the transfer learning approach.

In the case of part-word and word repetitions, the improvements obtained using the transfer learning setup are much less than filled pause and prolongation. This is because the occurrences of these two types of disfluencies can take up many forms, leading to high intra-class variance. This makes it difficult to model the samples belonging to part-word repetition and word repetition using the transfer learning setup. Hence, marginal average improvements of 1.5% and 1.4% are obtained in the detection performance for part-word repetition and word repetition, respectively, using all three disfluency detection systems.

From Table 4.3 and Table 4.6, we can see clearly that the knowledge is being transferred from the stutter domain to spontaneous speech disfluency domain, as should be the case in transfer learning. To

**Table 4.5** Detection results of the pre-trained model on IIITH-IED dataset without domain adaptation/retraining. Here, Acc. refers to Accuracy and F1 refers to the F1-score.

| | DNN | | BiLSTM | | MHA | |
|---|---|---|---|---|---|---|
| Disfluency Type | Accuracy | F1 | Accuracy | F1 | Accuracy | F1 |
| Filled Pause | 87.83 | 0.871 | 86.58 | 0.869 | 89.21 | 0.907 |
| Prolongations | 88.02 | 0.879 | 89.78 | 0.898 | 91.74 | 0.919 |
| Part-word Repetitions | 78.43 | 0.786 | 77.31 | 0.772 | 83.35 | 0.836 |
| Word Repetitions | 76.59 | 0.768 | 76.82 | 0.764 | 83.19 | 0.832 |

**Table 4.6** Detection results obtained using the proposed transfer learning approach. Here, Acc. refers to Accuracy and F1 refers to the F1-score.

| | **DNN** | | **BiLSTM** | | **MHA** | |
|---|---|---|---|---|---|---|
| **Disfluency Type** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** |
| Filled Pause | 92.73 | 0.927 | 91.60 | 0.910 | 94.96 | 0.937 |
| Prolongations | 94.90 | 0.949 | 94.01 | 0.941 | 97.68 | 0.969 |
| Part-word Repetitions | 83.43 | 0.826 | 80.45 | 0.799 | 89.57 | 0.892 |
| Word Repetitions | 81.16 | 0.807 | 79.87 | 0.795 | 89.52 | 0.889 |

bring a more comprehensive view of the proposed transfer learning method, the performance of the pre-trained model detecting the spontaneous speech disfluency without domain adaptation is shown, from Table 4.5 and Table 4.6 we can see there is a considerable difference in detection performance. i.e., in the case of domain adaptation, detection performance increases significantly compared to directly using a pre-trained model for disfluency detection in the IIITH-IED dataset.

## 4.7 Summary

In this chapter, we proposed a transfer learning approach to detect disfluencies in spontaneous speech using a model trained on stutter data. Disfluency detection is done for four types of disfluencies as a binary classification task. IIITH-IED dataset was used for disfluencies in spontaneous speech, while the UCLASS dataset was used for stuttering data. Three types of deep learning models, i.e. DNN, BiLSTM and attention-based, were trained for the classification of a particular type of stutter using MFCC features as input. These trained models were then tested for detecting the disfluency, which is most similar to that stutter-type. Using this approach, we obtained an average relative improvement of 2.69%, 2.00% and 2.04% in detection accuracy across all four disfluencies over the baseline experiment (when only the samples from the IIITH-IED dataset were taken) using DNN, BiLSTM and attention-based models, respectively.

*Chapter 5*

# Conclusions and Future Scope

## 5.1 Conclusions

In this thesis, using various machine learning techniques, we explore and develop models for detecting disfluencies in spontaneous English speech. Aligning with this problem, the issue of lack of resources for the study of speech disfluencies in Indian English is also addressed, and IIITH-IED dataset is proposed for the same. In addition to it, baseline disfluency detection systems are developed on the dataset, using state of the art techniques such as DNN, BiLSTM and attention based models. Finally, we propose a transfer learning approach to detect disfluencies in spontaneous speech using a model trained on stutter data.

The following are the conclusions drawn from this thesis:

- For preparing IIITH-IED dataset, 10-hours of lecture mode speech in Indian English was collected and annotated for five main types of disfluencies occurring in spontaneous speech, namely filled pause, prolongations, and 3 types of repetitions (part-word, word, and phrase).
- The collected data was then used to develop frame-level automatic disfluency detection systems.
- Two categories of experiments are performed. In the first one, the detection of every type of disfluency was set up as a binary classification task - the speech frame either belong to that disfluency type or does not. Another set of experiments were conducted to determine whether or not a specific speech frame is disfluent.
- For baseline experiments, two sets of features - Filterbank and MFCC features were used here for developing the disfluency detection systems with 5 different types of classifiers. MFCC features outperform the filterbank features, with Random forest coming as best classifier, closely followed by Multi Headed Attention based model.
- The accuracy and F1-score obtained for repetition type disfluencies are lower than those obtained for filled pause and prolongation. This is due to the fact that these types of disfluencies have a longer average duration, allowing for more variance in the samples of these classes.

- The accuracy and F1-score for phrase repetitions are lowest in all disfluencies. This can be attributed to the fact that with this class having the longest average duration, the number of samples of phrase repetition in the dataset are not enough to efficiently model this class.

- For further improving the baseline results, transfer learning based approach is proposed to detect disfluencies in spontaneous speech using a model trained on stutter data (UCLASS dataset). Four types of disfluencies are considered here, leaving out phrase repetitions because of above mentioned reason.

- In order to test hypothesis, MFCC features were used in transfer learning based experiments, with 3 state of the art deep learning architectures, namely, Deep Neural Network, Bidirectional-LSTM based model, and Multi Headed Attention based model.

- By using the proposed transfer learning approach, an increase in detection accuracy and F1-score was obtained for all four types of disfluencies, with all 3 classification methods.

- In the case of part-word and word repetitions, the improvements obtained using the transfer learning setup are substantially less than those obtained in filled pause and prolongation. This is because the occurrences of these two types of disfluencies can take up many forms, leading to high intra-class variance. This makes using the transfer learning setup to model samples from part-word repetition and word repetition difficult.

- Overall, transfer learning based approach used with multi headed attention gives results at par with Random forest in case of filled pause and prolongations. While for repetitions, random forest still outperforms all deep learning techniques, which can be attributed to the fact that more data is required by deep learning architectures to learn intra-class variance present in repetitions.

## 5.2 Future Scope

Following are the directions for future work:

- Other than frame level disfluency detection systems which are developed in this work, utterance level disfluency detection systems will also be aimed in future works.

- To study the effect of combination of text and speech features on this disfluency detection architecture, text-based features can also be incorporated into this pipeline.

- Further study needs to be done on segregation of fluent and disfluent repetitions, by incorporating prosodic information in the detection pipeline.

- Annotations of other disfluencies such as repairs should also be the focus of future works.

- Finally, these experiments would be extended to also include disfluencies in other Indian languages, analyse their forms and frequencies, and feature analysis in order to develop robust automatic disfluency detection systems in the Indian context.

# Publications

## Related Publications

- Sparsh Garg, Utkarsh Mehrotra, Gurugubelli Krishna, and Anil Kumar Vuppala. "Transfer Learning based Disfluency Detection using Stuttered Speech." In Workshop on Machine Learning in Speech and Language Processing (MLSLP) 2021.
- Sparsh Garg, Utkarsh Mehrotra, Gurugubelli Krishna, and Anil Kumar Vuppala. "Towards a Database For Detection of Multiple Speech Disfluencies in Indian English." In 2021 National Conference on Communications (NCC), pp. 1-6. IEEE, 2021.

## Other Publications

- Utkarsh Mehrotra, Sparsh Garg, Gurugubelli Krishna, and Anil Kumar Vuppala. "Detecting Multiple Disfluencies from Speech using Pre-linguistic Automatic Syllabification with Acoustic and Prosody Features." In Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2021.

# Bibliography


[1] S. Alharbi, M. Hasan, A. J. Simons, S. Brumfitt, and P. Green. A lightly supervised approach to detect stuttering in children's speech. In *Proceedings of Interspeech 2018*, pages 3433–3437. ISCA, 2018.

[2] K. Audhkhasi, K. Kandhway, O. D. Deshmukh, and A. Verma. Formant-based technique for automatic filled-pause detection in spontaneous spoken english. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4857–4860. IEEE, 2009.

[3] N. Bach and F. Huang. Noisy bilstm-based models for disfluency detection. In *INTERSPEECH*, pages 4230–4234, 2019.

[4] S. Betz, R. Eklund, and P. Wagner. Prolongation in german. In *DiSS 2017 The 8th Workshop on Disfluency in Spontaneous Speech, KTH, Royal Institute of Technology, Stockholm, Sweden, 18–19 August 2017*, pages 13–16. KTH Royal Institute of Technology, 2017.

[5] C. Büchel and M. Sommer. What causes stuttering? *PLoS biology*, 2(2):e46, 2004.

[6] D. Cai, W. Cai, and M. Li. Within-sample variability-invariant loss for robust speaker recognition under noisy environments. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6469–6473. IEEE, 2020.

[7] S. Candeias, D. Celorico, J. Proença, A. Veiga, and F. Perdigão. HESITA(tions) in Portuguese: a database. In *Proc. Disfluency in Spontaneous Speech (DiSS 2013)*, pages 13–16, 2013.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[9] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob. Automatic detection of prolongations and repetitions using lpcc. In *2009 international conference for technical postgraduates (TECHPOS)*, pages 1–4. IEEE, 2009.

[10] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob. 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)*, pages 146–149. IEEE, 2009.

[11] M. Corley and O. W. Stewart. Hesitation disfluencies in spontaneous speech: The meaning of um. *Language and Linguistics Compass*, 2(4):589–602, 2008.

[12] A. Das, J. Mock, H. Chacon, F. Irani, E. Golob, and P. Najafirad. Stuttering speech disfluency prediction using explainable attribution vectors of facial muscle movements. *arXiv preprint arXiv:2010.01231*, 2020.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] C. Donahue, B. Li, and R. Prabhavalkar. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5024–5028. IEEE, 2018.

[15] Q. Dong, F. Wang, Z. Yang, W. Chen, S. Xu, and B. Xu. Adapting translation models for transcript disfluency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6351–6358, 2019.

[16] R. Dufour, Y. Estève, and P. Deléglise. Characterizing and detecting spontaneous speech: Application to speaker role recognition. *Speech communication*, 56:1–18, 2014.

[17] J. Ferguson, G. Durrett, and D. Klein. Disfluency detection with a semi-markov model and prosodic features. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 257–262, 2015.

[18] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

[19] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning (adaptive computation and machine learning series). *Cambridge Massachusetts*, pages 321–359, 2017.

[20] M. Goto, K. Itou, and S. Hayamizu. A real-time filled pause detection system for spontaneous speech recognition. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[21] B. Guitar. *Stuttering: An integrated approach to its nature and treatment*. Lippincott Williams & Wilkins, 2013.

[22] R. Hamzah and N. Jamil. Investigation of speech disfluencies classification on different threshold selection techniques using energy feature extraction. *Malaysian Journal of Computing*, 4(1):178–192, 2019.

[23] P. Howell, S. Davis, and J. Bartrip. The university college london archive of stuttered speech (uclass). 2009.

[24] F. S. Juste and C. R. F. De Andrade. Speech disfluency types of fluent and stuttering individuals: age effects. *Folia Phoniatrica et Logopaedica*, 63(2):57–64, 2011.

[25] M. Kaushik, M. Trinkle, and A. Hashemi-Sakhtsari. Automatic detection and removal of disfluencies from spontaneous speech. In *Proceedings of the Australasian International Conference on Speech Science and Technology (SST)*, volume 70, 2010.

[26] T. Kourkounakis, A. Hajavi, and A. Etemad. 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)*, pages 6089–6093. IEEE, 2020.

[27] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

[28] R. J. Lickley. *Detecting disfluency in spontaneous speech*. PhD thesis, University of Edinburgh, 1994.

[29] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on audio, speech, and language processing*, 14(5):1526–1540, 2006.

[30] P. J. Lou and M. Johnson. End-to-end speech recognition and disfluency removal. *arXiv preprint arXiv:2009.10298*, 2020.

[31] Y. Lu, M. J. Gales, K. Knill, P. Manakul, and Y. Wang. Disfluency detection for spoken learner english. In *SLaTE*, pages 74–78, 2019.

[32] P. Mahesha and D. Vinod. Lp-hillbert transform based mfcc for effective discrimination of stuttering dysfluencies. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2561–2565. IEEE, 2017.

[33] S. R. Maskey, Y. Gao, and B. Zhou. Disfluency detection for a speech-to-speech translation system using phrase-level machine translation with weighted finite state transducers, Dec. 28 2010. US Patent 7,860,719.

[34] S. M. Mathews. Language skills and secondary education in india. *Economic and Political Weekly*, 53(15):20–22, 2018.

[35] S. Oue, R. Marxer, and F. Rudzicz. Automatic dysfluency detection in dysarthric speech using deep belief networks. In *Proceedings of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies*, pages 60–64, 2015.

[36] J. Proença, D. Celorico, S. Candeias, C. Lopes, and F. Perdigão. Children's reading aloud performance: a database and automatic detection of disfluencies. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[37] X. Qian and Y. Liu. Disfluency detection using multi-step stacked learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–825, 2013.

[38] R. Riad, A.-C. Bachoud-Lévi, F. Rudzicz, and E. Dupoux. Identification of primary and collateral tracks in stuttered speech. *arXiv preprint arXiv:2003.01018*, 2020.

[39] E. Salesky, M. Sperber, and A. Waibel. Fluent translations from disfluent speech in end-to-end speech translation. *arXiv preprint arXiv:1906.00556*, 2019.

[40] J. Santoso, T. Yamada, and S. Makino. Classification of causes of speech recognition errors using attention-based bidirectional long short-term memory and modulation spectrum. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 302–306. IEEE, 2019.

[41] E. Shriberg. To 'errrr'is human: ecology and acoustics of speech disfluencies. *Journal of the international phonetic association*, 31(1):153–169, 2001.

[42] V. Silber-Varod, M. Gósy, and R. Eklund. Segment prolongation in hebrew. In *The 9th Workshop on Disfluency in Spontaneous Speech*, page 47, 2019.

[43] G. Tottie. On the use of uh and um in american english. *Functions of Language*, 21(1):6–29, 2014.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[45] B. Villegas, K. M. Flores, K. J. Acuña, K. Pacheco-Barrios, and D. Elias. A novel stuttering disfluency classification system based on respiratory biosignals. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4660–4663. IEEE, 2019.

[46] M. Wiśniewski, W. Kuniszyk-Jóźkowiak, E. Smołka, and W. Suszyński. Automatic detection of prolonged fricative phonemes with the hidden markov models approach. *Journal of Medical Informatics & Technologies*, 11, 2007.

[47] C.-H. Wu and G.-L. Yan. Acoustic feature analysis and discriminative modeling of filled pauses for spontaneous speech recognition. In *Real World Speech Processing*, pages 17–30. Springer, 2004.

[48] V. Zayats and M. Ostendorf. Robust cross-domain disfluency detection with pattern match networks. *arXiv preprint arXiv:1811.07236*, 2018.

[49] V. Zayats, M. Ostendorf, and H. Hajishirzi. Disfluency detection using a bidirectional lstm. *arXiv preprint arXiv:1604.03209*, 2016.