Speech Correction System for Stutterers using Deep Neural Networks

Abstract: refer other papers for idea

The goal of this paper is to implement a system which monitors a stutterers speech in real time, develop statistics and further inferences about their disfluencies, introduce and train them to apply a technique known as “long lengthening style”, which is known the be very effective to alleviate the problems associated with stuttering. A privately prepared dataset, consisting of “LL and non –LL words”, is fitted into a deep neural network, which is used to monitor the user's speech in real time, and alert them if they are using the prescribed technique. The primary goal of this study is to habituate this new fluent way of speaking, long enough for it have a permanent impact on the user's speech, such that the benefits of fluent speech will remain, even after discontinuation from the system, resulting in permanent fluency.

Introduction to Problem Statement:

What is Stuttering and why does it occur:

Speech is the primary source of communication used by humans, today, and is considered one of the strongest habits developed in the human brain. There are many disorders associated with speech. They can be congenital, developmental, or neurogenic. Stuttering is one such disorders that can originate through all these routes, being congenital in the sense that children will more likely develop a stutter if there is a history of the condition in the family. It has not been proved that stuttering carries a strong deterministic representation in human genetics, that can be passed from parent to progeny.

Most cases of stuttering are developmental, in the sense that it is primarily a habitual disorder which is introduced in childhood, through environmental factors, which may or may not carry on till adolescence and adulthood.

Neurogenic stuttering can occur as a result of traumatic experiences, accidents or events, suddenly, in a person who has never reported its symptoms before.

The definitive cause of this disorder has yet to be found, although there are many factors that contribute to the likelihood of its appearance.

1% of the global population suffers from this condition, it is 4 times likely to appear in males than females.

Current Studies and Solutions:

Current studies are mainly focused on developing systems to accurately detect the presence of stuttering, and its degree of occurrence to some extent, by decomposing it to its main symptoms. 1. Prolongation 2. Interjection 3. Blockage 4. Phrase/word Repetition. It focuses on the techniques used for feature extraction of spoken speech, the different classifiers available for usage, and their varying accuracy for classifications of disfluent speech.

Introduction to Method Used:

What Is the solution chosen?

Long Lengthening Speech, henceforth referred to as "LL", is an amalgamation of different methods to alleviate symptoms, introduced and taught by speech pathologist Partha Bagchi, who is a long-time researcher and instructor, who has a patient for the same disorder in question. The exact details and guidelines introduced with this technique are delineated in futures sections, and can be found in several of his books, and documentation.

How does it work and is it effective?

The user is instructed to follow the following guidelines strictly in order to make use of this technique, and consequently speak fluently without any disruptions.

1. Person talks more slowly than average spoken speed. (~60 words per minute)
2. Person uses an alternative rhythm/prosody of speech.
1. Person avoids conjunction of words while speaking, and instead a word by word approach.

Problems with the technique:

Under patients of Partha Bagchi, it is generally observed that, the technique, when used properly under guidance from a mentor, largely alleviates the problems associated with stuttering, if not completely, among first time users under supervision, while speaking to a crowd of other patients, or while speaking in a conversation with another person, or reading from pre written material.

It is prescribed by him as an all-encompassing strategy for mild or even severe stutterers, if practiced diligently, and used properly while speaking.

However, there are unavoidable problems associated in this approach of “habit modeling”, not in the technique itself, but the attempt of successfully incorporating it in the speech of a seasoned stutterer, in all times and situations.

It is important to realize that if this technique is used properly, following the prescribed guidelines, there is no disfluency present in the speech of the user. All incomplete/ improper attempts at using this technique, can result in stuttering symptoms, which gives the false indication of failure in the technique, which instead is a failure in execution due to multiple reasons further discussed below.

Following are the most common problems associated with using this technique:

1. The user often neglects practice and use of this technique, due to lack of diligence or motivation.
2. The initially observed “failure” of this technique, while using it under socially stressful situations, as perceived by the user, resulting to the seemingly correct, by objectively improper use of this technique, due to the following reasons.

A) Anxiety due to not being able to conform to the standard way of speaking, resulting in partial use of technique.

B) Anxiety due to social pressure, due to consciousness in the environment of speaking, and lack of focus into proper usage of technique and formation of words.

Proposal to implement the device:

In order to mimic the environment of having a supervisor that constantly monitors speech and gives feedback, to ensure it is correctly used, a device is proposed that has the similar ability to distinguish between LL and non-LL speech, forcing the user to conform to the guidelines.

This device ensure that the user gives adequate amount of focus into his speech and serves as a feedback device to get the user back from disfluency, to fluency and LL usage.

Such a device can also be used to build up a database of statistics, with relevant information regarding speech, with its timestamp, to track progress and give useful insights into the efficacy of the long lengthening technique.

Such I described in the future scope section of this paper.

Personal History of Stuttering:

Author is a 21-year-old Male, with a history of persistent developmental stuttering in the immediate family, reported to experience the problem from the age of 5.

Similar Technologies in Use:

DAF (Delayed Auditory Feedback) trains patients to speak fluently by playing back the patient's voice after some delays. DAF plays the person's own voice back to them, and they hear it with a slight delay, usually about one tenth of a second later. The aim of the paper is to design DAF in order to cure stuttering and relief patients from stresses that caused by stuttering.

Unlike DAF, which completely ignores the input by the user, this speech monitor gives useful information and guidance back to the speaker.

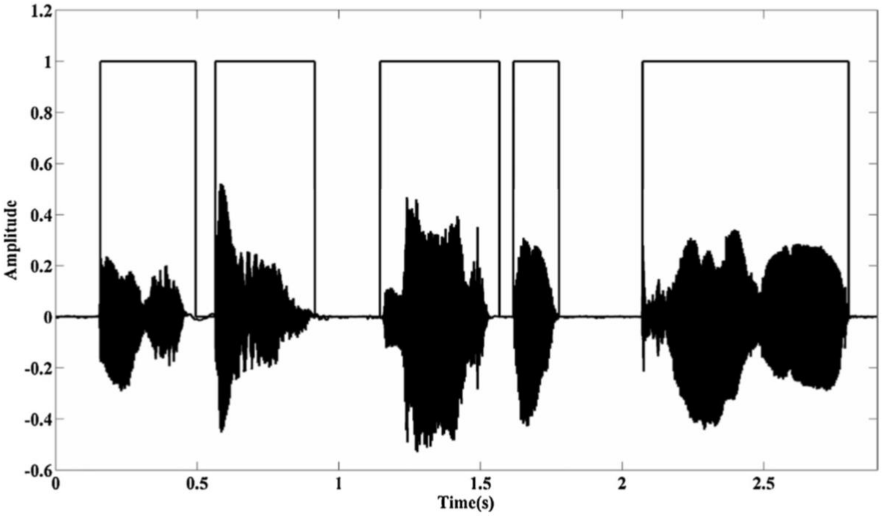
PROJECT WORKFLOW:

1. Input sentence into microphone and store.
2. Split the sentence into words, with the help of volume thresholding.
3. Store extracted words into directory and normalize the data.
4. Feed words into model and make predictions. "LL or Non-LL"

5. Build statistics based on predictions and give alert when needed to the user in real time.

WORD EXTRACTION:

Each sentence that is recorded is stored and split into its corresponding words, as .wav files ready for processing.



FEATURE EXTRACTION:

MFCC or “Mel scale Filter Cepstrum Co-efficient" is an indispensable tool used in speech recognition. Along with ANN, it is known to successfully classify speech data with an accuracy of 90%. Its pipeline is inspired by the physiological inner working of a human ear, according to the “Mel Scale”,

MFCC:

The time domain waveform of a speech signal carries all of the auditory information. From the phonological point of view, very little can be said on the basis of the waveform itself.

However, past research in mathematics, acoustics, and speech technology have provided many methods for converting data, which can be considered as information if interpreted correctly. In order to find some statistically relevant information from incoming data, it is important to have mechanisms for reducing the information of each segment in the audio signal into a relatively small number of parameters, or features. These features should describe each segment in such a characteristic way that other similar segments can be grouped together by comparing their features.

Methods that utilize information in the periodicity of speech signals could be used to overcome this problem, although speech also contains a periodic content. The non-linear frequency scale used an approximation to the Mel-frequency scale which is approximately linear for frequencies below 1 kHz and logarithmic for frequencies above 1 kHz. This is motivated by the fact that the human auditory system becomes less frequency-selective as frequency increases above 1 kHz. The MFCC features correspond to the cepstrum of the log filter bank energies.

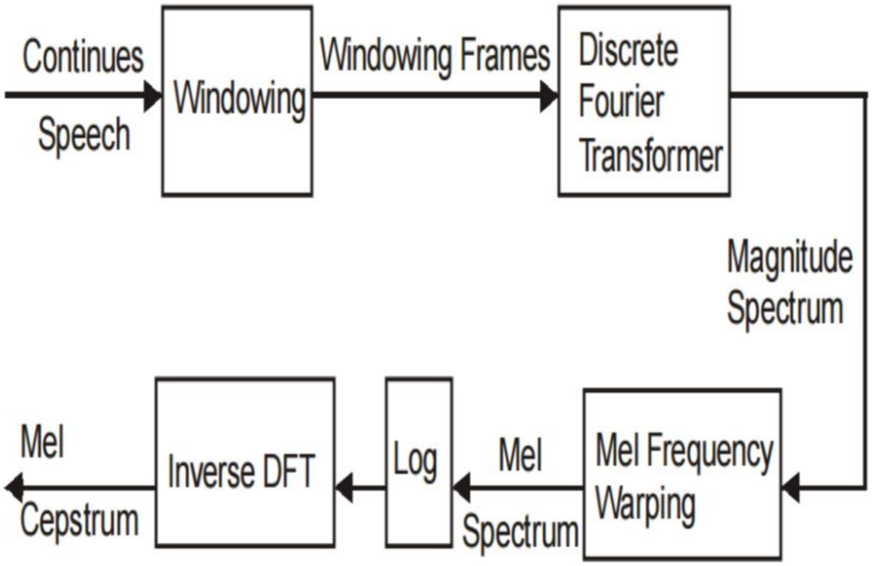
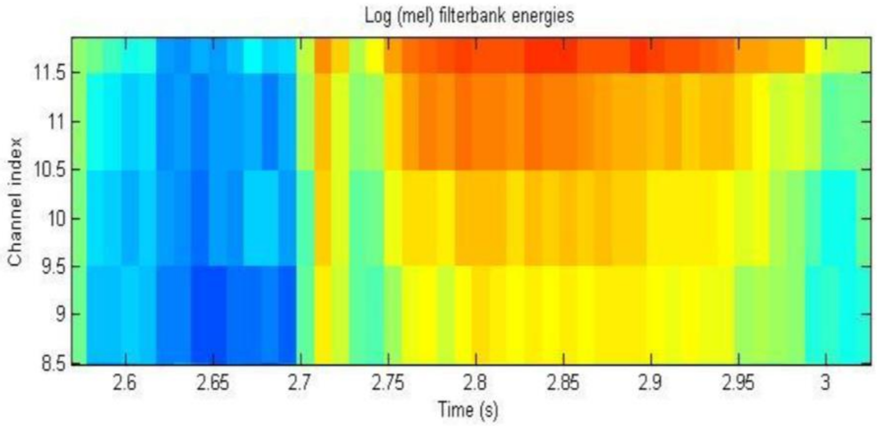
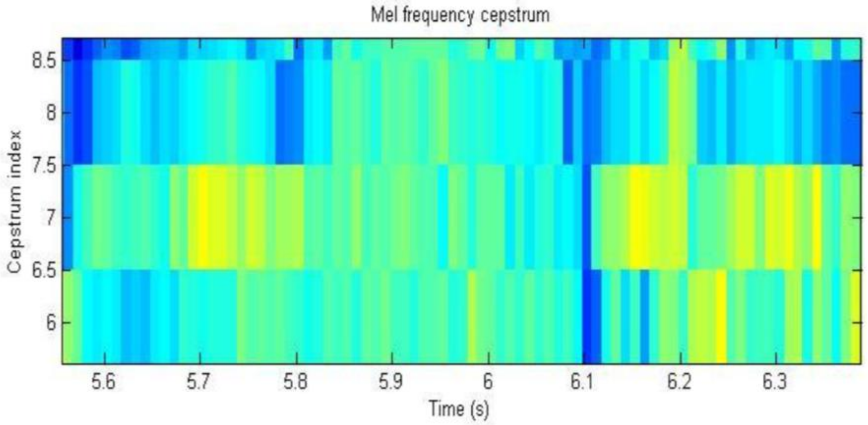


Figure 3: MFCC block diagram

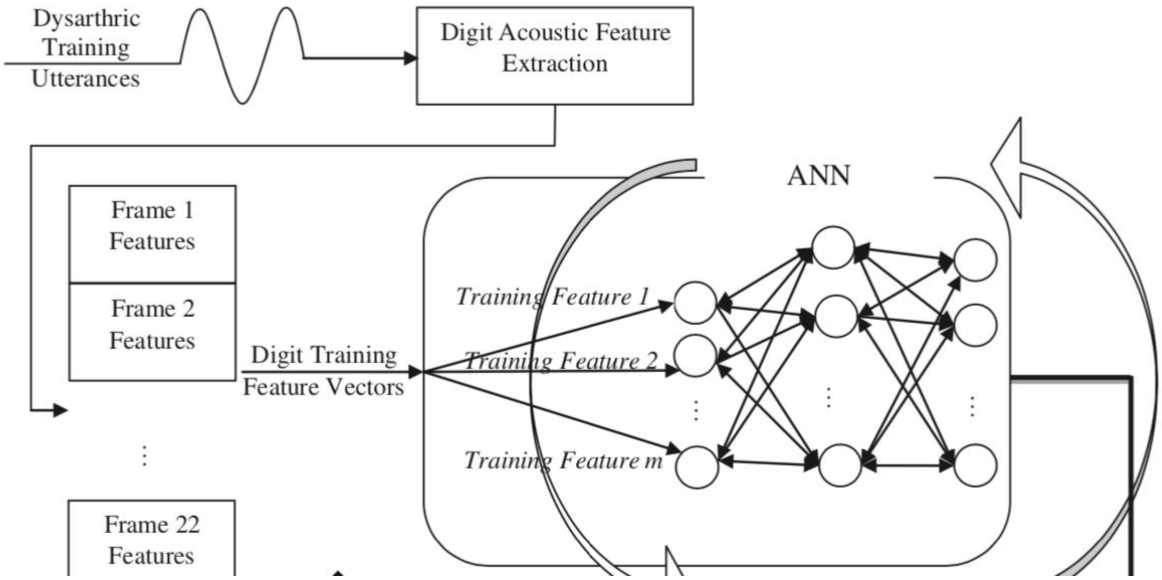




Deep Neural Network:

A neural network, in general, is a technology built to simulate the activity of the human brain – specifically, pattern recognition and the passage of input through various layers of simulated neural connections.

Many experts define deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as “feature hierarchy.” One of the key uses of these sophisticated neural networks is dealing with unlabeled or unstructured data. The phrase “deep learning” is also used to describe these deep neural networks, as deep learning represents a specific form of machine learning where technologies using aspects of artificial intelligence seek to classify and order information in ways that go beyond simple input/output protocols.



Our model:

JSON description of the ANN below:

{"class_name": "Sequential", "keras_version": "2.2.4", "config": {"layers": [{"class_name": "Dense", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "dense_1", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "dtype": "float32", "activation": "relu", "trainable": true, "kernel_regularizer": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 20, "batch_input_shape": [null, 20], "use_bias": true, "activity_regularizer": null}}, {"class_name": "Dropout", "config": {"rate": 0.3, "noise_shape": null, "trainable": true, "seed": null, "name": "dropout_1"}}, {"class_name": "Dense", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "dense_2", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "activation": "relu", "trainable": true, "kernel_regularizer": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 80, "use_bias": true, "activity_regularizer": null}}, {"class_name": "Dropout", "config": {"rate": 0.3, "noise_shape": null, "trainable": true, "seed": null, "name": "dropout_2"}}, {"class_name": "Dense", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "dense_3", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "activation": "relu", "trainable": true, "kernel_regularizer": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 40, "use_bias": true, "activity_regularizer": null}}, {"class_name": "Dropout", "config": {"rate": 0.3, "noise_shape": null, "trainable": true, "seed": null, "name": "dropout_3"}}, {"class_name": "Dense", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "dense_4", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "activation": "relu", "trainable": true, "kernel_regularizer": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 20, "use_bias": true, "activity_regularizer": null}}, {"class_name": "Dropout", "config": {"rate": 0.3, "noise_shape": null, "trainable": true, "seed": null, "name": "dropout_4"}}, {"class_name": "Dense", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "dense_5", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "activation":

"softmax", "trainable": true, "kernel_regularizer": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 2, "use_bias": true, "activity_regularizer": null}}], "name": "sequential_1"}, "backend": "tensorflow"}

Code:

```
def makemodel():

  print ("Making model")
  model = Sequential()

  BatchNormalization(
    axis=-1, momentum=0.99,
    epsilon=0.001,
    center=True,
    scale=True,
    beta_initializer='zeros',
    gamma_initializer='ones',
    moving_mean_initializer='zeros',
    moving_variance_initializer='ones',
    beta_regularizer=None,
    gamma_regularizer=None,
    beta_constraint=None,
    gamma_constraint=None)

  model.add(Dense(units=20, activation='relu', input_dim=20))
  model.add(Dropout(0.3))
  model.add(Dense(units=80, activation='relu'))
  model.add(Dropout(0.3))
  model.add(Dense(units=40, activation='relu'))
  model.add(Dropout(0.3))
  model.add(Dense(units=20, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(activation='softmax', output_dim=2))

  adam = optimizers.Nadam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, schedule_decay=0.004)

  model.compile(loss='categorical_crossentropy',
          optimizer=adam,
          metrics=['accuracy'])

  return model
```

Hyper-Parameters of Best Model:

1. Distribution: train/test ratio = 0.8/0.2
2. Learning rate = 0.001, with ADAM Optimization,

```
adam = optimizers.Nadam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, schedule_decay=0.004)
```

3. Accuracy and Score:

   Test Set: Accuracy = ~95%, Score = ~0.20

   Test Set: Accuracy = ~88%, Score = ~0.40

4. Batch Normalization is used to speed up training: Momentum = 0.99, Epsilon = 0.001
5. Dropout: 0.3

Issues with Model:

The difference in train and test accuracy indicates that the model is overfit, even after using Dropout.

It can be alleviated with regularization techniques like L1, L2 regularization, or by increasing the size of the entire dataset.

Disadvantages of Proposed Solution:

1. The MFCC method of feature extraction is highly vulnerable to noise and causes misclassification of spoken words.
2. As a result, the accuracy in real time usage of the application is negatively affected.
3. The scope of this study only applies for a single speaker dataset and has not been tested for multiple individuals.
4. The size of the dataset is relatively small for the application.

Future Scope:

The study must be extended for multiple users to truly gauge the effectiveness of the Long Lengthening Technique. In further studies, a correlation between Fluency and LL can be assessed, and it can be determined to what degree does LL influence the occurrence of fluency in a stutterer's speech.

Conclusion:

A device like the one that has been developed, carries promising results as a system that can monitor and train an individual to build up fluency over time.

It has good immediate results while being used with a speaker who has been properly trained to use it. It immediately alleviates disruptions in speech and can help individuals in high stress situations.

BACKEND OF SPEECH APP: (developed from scratch)

https://github.com/sharan21/Speech-Assisting-App

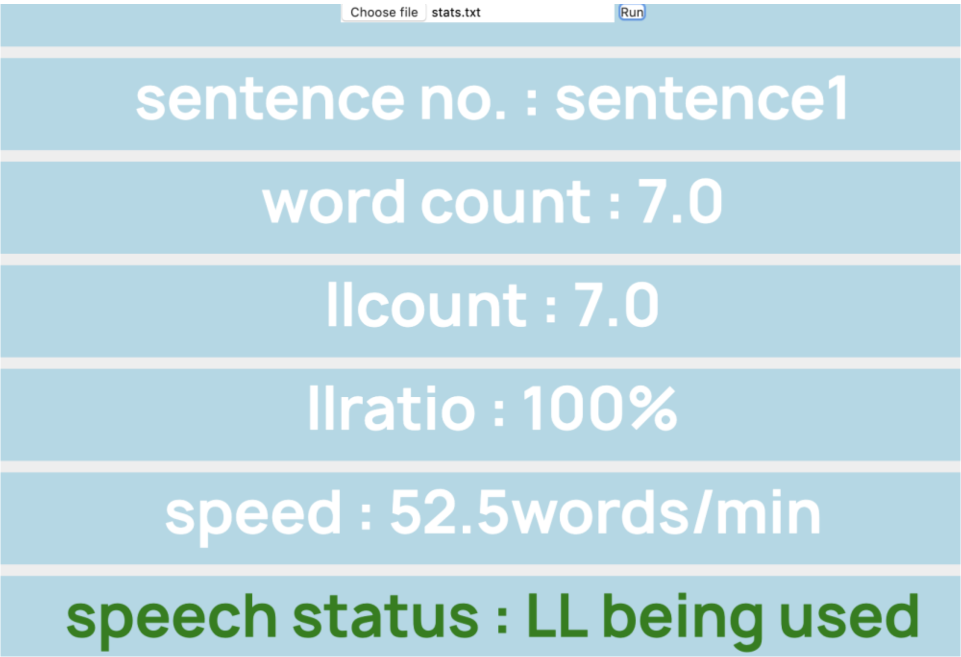
1. Consists of a Class, *speechanalyser()*, which monitors the speech and builds statistics.
2. This module also consists of the Database, Neural Network Model, and Functionality of Input Feature Extraction and Data Cleaning and Distribution.

SpeechApp/speechanalyser.py

STATISTICAL ANALYSIS:

After prediction using the above library, the predictions as well as relevant speech data is stored locally in a logs file.

It is then displayed on the App, to show to the user and create alerts to make user follow the proper technique.



sentence no. : sentence1

word count : 7.0

llcount : 7.0

llratio : 100%

speed : 52.5words/min

speech status : LL being used

The app continuously gets data every 8 seconds of speech.

REFERENCES: