# CLEANTONE: ENHANCING AUDIO CLARITY THROUGH ADVANCED DISFLUENCY DETECTION AND REMOVAL

# INTRODUCTION

Speech disfluencies refer to the interruptions or irregularities in the flow of spoken language that include silent pauses, prolongations, repetitions of words, hesitation markers. Efficient communication relies on smooth and uninterrupted speech. In addressing these interruptions, especially for fast-paced speakers, strategic inclusion of silent pauses becomes essential.

The main goal is to detect and eliminate the interruptions, to incorporate the tips for improving communication skills and ultimately to enhance the fluidity of spoken language for improved communication and processing tasks.

# PROBLEM STATEMENT

The frequent occurrence of disfluencies poses a significant challenge for effective communication processing. These interruptions disrupt the fluency, lead to the misinterpretation of the speaker’s intended message and degradation in overall audio quality.

In such instances, there is a need for the detection and removal of these disfluencies in the recorded audio to enhance the overall speech quality and also analyzing the speaker’s delivery of speech for improving their communication skills.

# OBJECTIVES

1. To improve the overall quality and fluency of the spoken content through precise detection and efficient removal of disfluencies.
2. To ensure that the removal of such disfluencies does not affect the intended meaning or conveyance of the actual message.
3. To perform detailed analysis by comparing the audio content, emphasizing speaker’s speech rate to assess improvements effectively.
4. To implement a question pool generation system based on the audio content.
5. To optimize the disfluency detection and removal algorithms to achieve high accuracy and efficiency in real-time.

**ARCHITECTURE DIAGRAM**

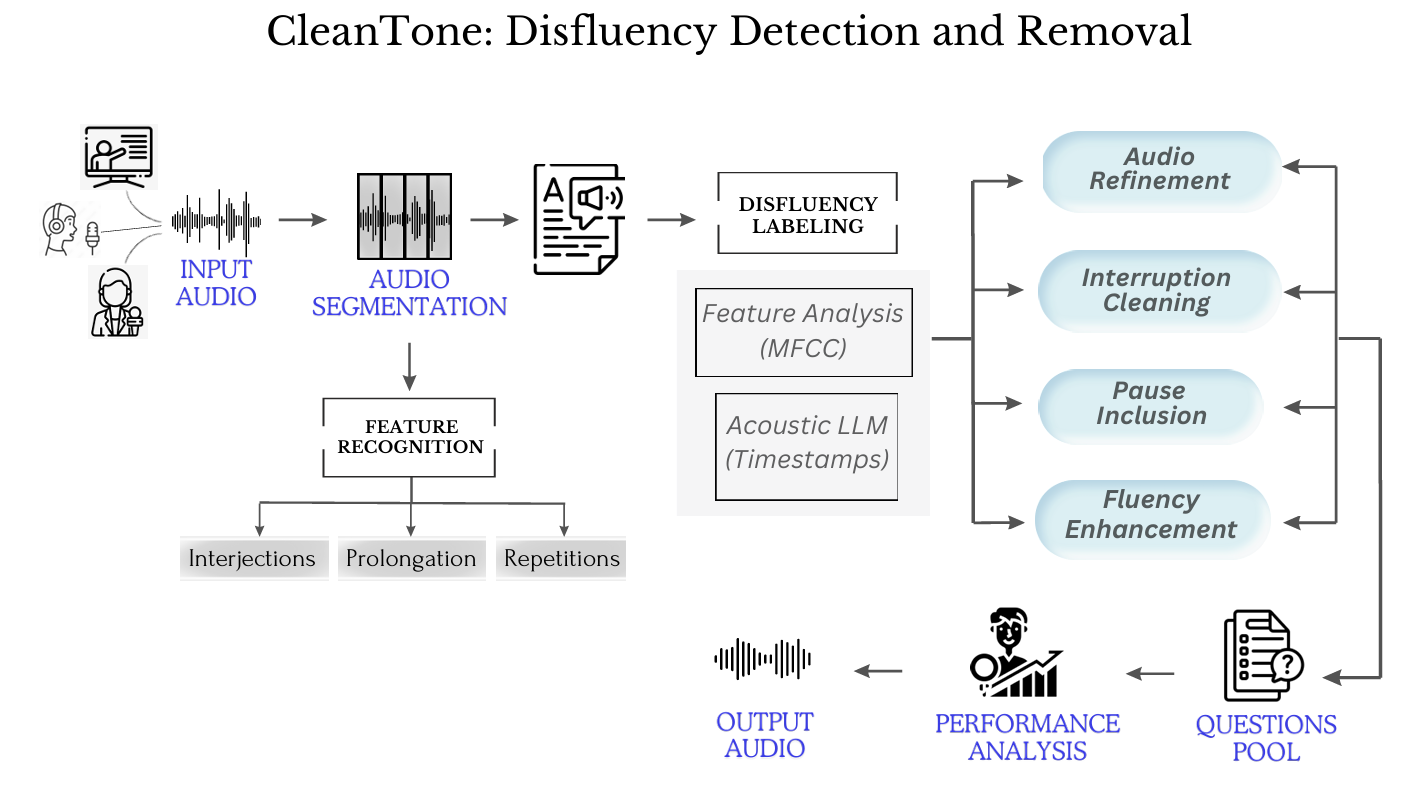
****

Fig 1 CleanTone: Disfluency Detection and Removal

# PROPOSED WORK

* + To analyse the audio signals and addressing interruptions, particularly for speakers with fast-paced speech using advanced algorithms.
  + Mel-frequency cepstral coefficients (MFCCs) or other features will be extracted from the preprocessed audio to capture speaker characteristics and speech patterns.
  + Train the model and optimize the accuracy in detecting various disfluencies.
  + Remove the detected disfluencies and ensure that the removal process maintains the naturalness and authenticity of the speaker's voice.
  + An interactive question pool will be generated based on the audio content and also performance of the speaker will be evaluated by analyzing the speech rate.
  + Implement techniques to perform fluency analysis of the source audio content and processed audio content to better understand the speaker's intentions, enabling more accurate disfluency detection and removal.

# ILLUSTRATION

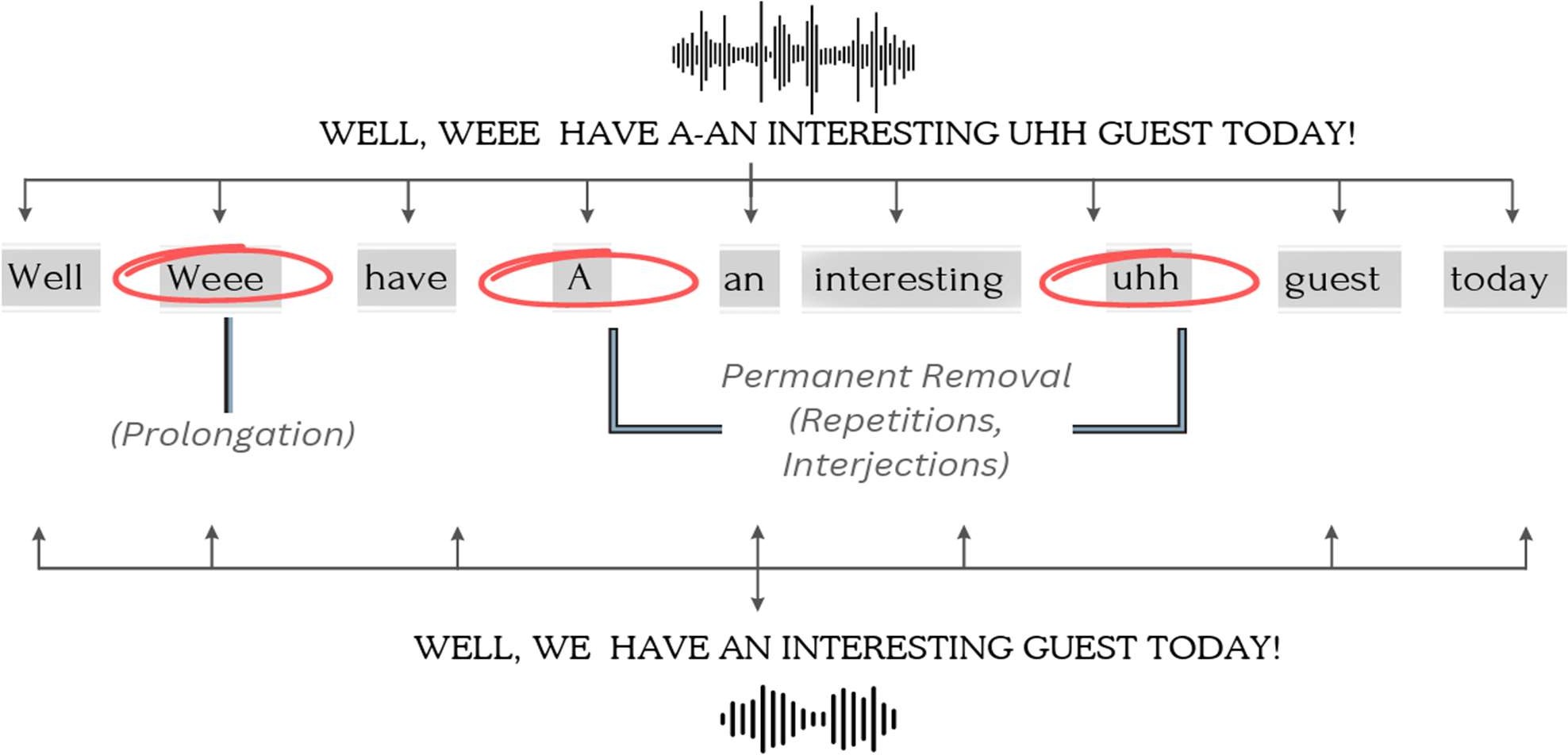


Fig 2 Workflow Diagram

**MODULES:**

1 – Data Preparation and Feature Analysis

2 – Disfluency Detection

3 – Disfluency Elimination

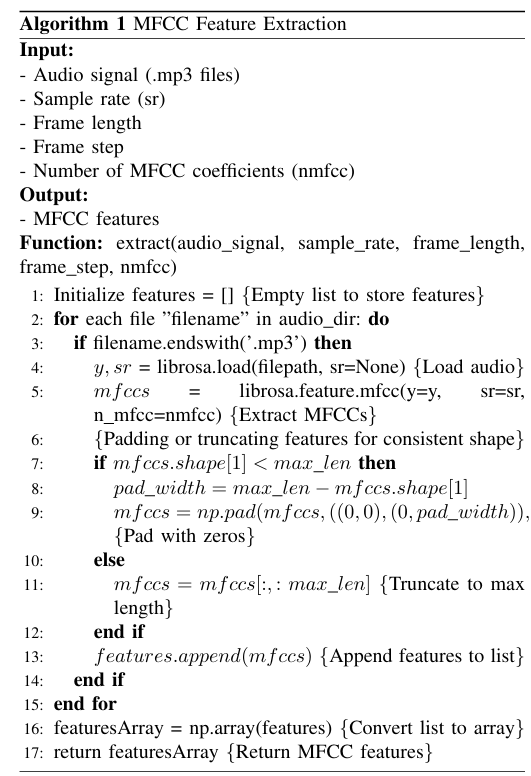
4 – Question Pool Generation

5 – Speaker’s Performance Analysis

## Module 1

## Mel-Frequency Ceptral Coefficients (MFCC):

* + MFCC is a widely used feature extraction technique and it involves Pre- emphasis, Framing, Windowing, FFT, Mel Filterbank, DCT.



## 

## Module 2

## Disfluency Detection

## Involves developing a model to automatically identify disfluencies (e.g., repetitions, fillers) in speech data.

## Whisper-timestamped, which is a layer on top of the Whisper set of models enabling us to get accurate word timestamps.

## This enables the identification of interruptions with high accuracy, shedding light on the nuances of speech patterns.

## As per in the Fig 2, it predicts disfluency types, counts occurrences, and extracts segments.

## 

**Module 3**

## Disfluency Elimination

## We identify and remove disfluencies, such as pauses, hesitations, and interruptions, from spoken language recordings.

## ‘Moviepy’ to extract audio from, split, and then re-concatenate our original video files.

## The audio segmentation process involves dividing the audio data into smaller frames or segments using timestamps which is displayed in the Fig 3.

## The steps an input file goes through:

## 

Fig 3 Various stages that the source file traverses

# IMPLEMENTATION:

# RECORD AUDIO FROM MICROPHONE

# By incorporating JavaScript for audio recording, the code enhances user interaction and accessibility, allowing for smooth real-time capture of audio input. In Fig 4, the conversion of the recorded audio into MP3 format further enhances usability.

# 

Fig 4 Live Audio Recording

# TRASCRIPTION OF AUDIO FILES WITH DETECTED DISFLUENCIES

# The provided Python script utilizes a speech recognition model to transcribe audio files, detecting disfluencies. As per in Fig 5, it predicts types, disfluency counts and extracts segments.

# 

Fig 5 Transcription of audio to text with detected disfluencies

The code extracts file names and disfluency counts from audio files info which contains file names, transcription of the source audio file, disfluency segments and disfluency counts. Then finally visualizes them in Fig 6.

# 

Fig 6 Visualizing Audio files with Disfluency Counts

**REMOVAL OF DETECTED DISFLUENCIES**

We identify and remove disfluencies, such as pauses, hesitations, and interruptions, from spoken language recordings. This process enhances communication clarity and fluency, ensuring a seamless listening experience and facilitating language processing tasks which is displayed in Fig 7.

# 

Fig 7 Merged Output Audio Files

**QUESTION POOL GENERATION**

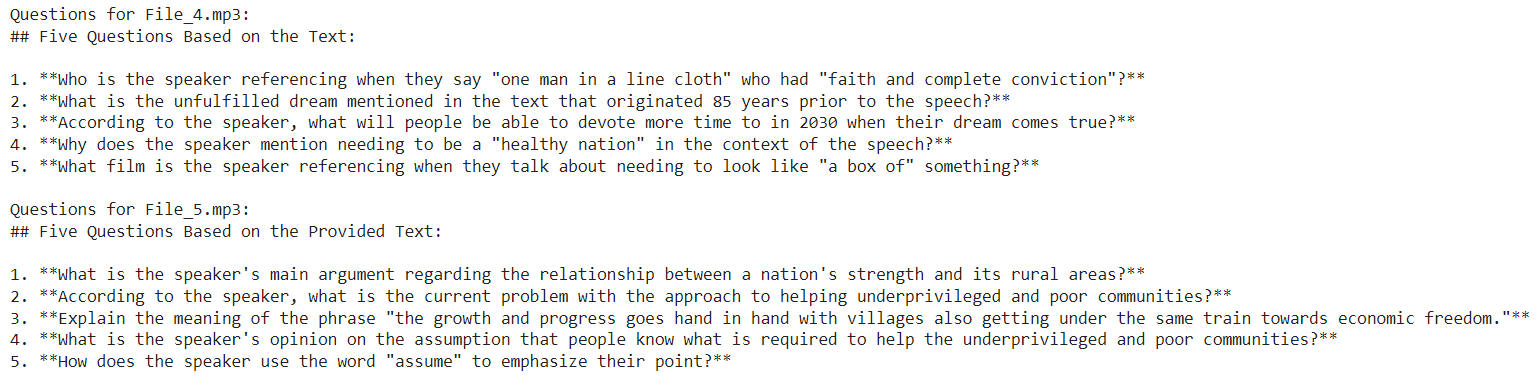
 The script iterates through audio files, generating questions based on their transcriptions using a language model. It prompts for summarization, then generates and prints five questions per file. As per in Fig 8, the output displays each file's name followed by the questions generated from its transcription, facilitating efficient question pool generation from audio content.

Fig 8 Generation of Questions

# PERFORMANCE ANALYSIS

The script analyzes audio files, comparing input and output sizes and durations. As per in Fig 9, it retrieves transcription and disfluency count information, then generates communication improvement tips based on this data. The output displays file names, sizes, durations, and personalized tips for enhancing speakers' communication skills, considering disfluencies in the audio content.

# 

Fig 9 Suggestions for Improving Speaking Skills

# 

Fig 10 Comparison of Source and Processed Output Files