  
   
  
 Muhammad Shawaiz Ishtiaq (01-134212-128)  
 Muhammad Faizan Khan (01-134212-110)  
  
 **SPEECH RECOGNITION SYSTEM FOR STAMMERS**

DEPARTMENT OF COMPUTER SCIENCE  
  
**Problem and Significance**: Speech disorders, particularly stuttering, significantly impact an individual's ability to communicate effectively, leading to reduced self-confidence and social isolation. Despite advancements in speech therapy and related fields, the need for accessible, cost-effective, and accurate tools to detect and analyze stuttering remains a critical issue. Current methods are often manual, time-consuming, and require trained professionals, which limits their accessibility. Developing an automated system that leverages speech recognition technology to identify stuttering patterns can greatly enhance early detection and intervention, providing substantial benefits to individuals and healthcare providers.

**Method/Tool/Technology and Solution**: This project proposes the development of an automated stuttering detection system using advanced speech recognition and audio processing technologies. The system will utilize Python libraries such as SpeechRecognition for converting speech to text, pydub for audio file manipulation, and noisereduce for enhancing audio quality. By employing these tools within an accessible platform like Google Colab, users can upload audio recordings, which are then processed to identify repeated words and potential stuttering patterns. This solution aims to provide an accurate, scalable, and user-friendly tool that can be used by individuals, speech therapists, and researchers to better understand and manage stuttering.

**Objectives**

1. **Develop an Automated Stuttering Detection System**: Create a software tool that can accurately identify repeated words in audio recordings as potential indicators of stuttering.
2. **Enhance Accessibility of Speech Disorder Analysis**: Utilize cloud-based platforms to ensure the tool is widely accessible to users without the need for specialized hardware or software.
3. **Integrate Advanced Audio Processing Techniques**: Implement noise reduction and audio format conversion to improve the accuracy and reliability of speech recognition.
4. **Provide User-Friendly Interface**: Design an intuitive interface for easy audio file upload and analysis, suitable for both professionals and laypersons.

**Methodology**

1. **Literature Review**: Conduct a thorough review of existing speech recognition technologies and stuttering analysis methods.
2. **Tool Selection and Integration**: Choose appropriate Python libraries (SpeechRecognition, pydub, noisereduce) and integrate them into a cohesive system.
3. **System Development**: Develop the core functionality for audio file upload, conversion, noise reduction, and speech-to-text processing.
4. **Algorithm Design**: Create algorithms to analyze text for repeated words and stuttering patterns.
5. **Testing and Validation**: Test the system with a variety of audio samples to ensure accuracy and reliability.
6. **User Interface Design**: Develop a user-friendly interface in Google Colab for easy interaction with the system.
7. **Documentation and Training**: Provide comprehensive documentation and tutorials to assist users in utilizing the tool effectively.

**Feasibility Plan**

1. **Technical Feasibility**: Assess the availability of required libraries and their compatibility with Google Colab. Ensure that the proposed system can handle different audio formats and qualities.
2. **Operational Feasibility**: Determine the ease of use for potential users, including speech therapists and individuals with no technical background. Ensure the system can be easily accessed and operated.
3. **Economic Feasibility**: Evaluate the cost-effectiveness of using cloud-based platforms and open-source libraries. Estimate the costs related to development, testing, and deployment.
4. **Schedule Feasibility**: Develop a realistic timeline for each phase of the project, from initial research to final deployment and user training.

**Tools**

1. **Google Colab**: A cloud-based platform for developing and running Python code.
2. **SpeechRecognition**: A Python library for converting speech to text.
3. **pydub**: A Python library for audio file manipulation, including format conversion.
4. **noisereduce**: A Python library for reducing background noise in audio files.
5. **Google Drive Integration**: For easy storage and retrieval of audio files within the Colab environment.

**Code:**

!pip install SpeechRecognition pydub noisereduce

import speech\_recognition as sr

from pydub import AudioSegment

import noisereduce as nr

import os

def identify\_stuttered\_word(text):

    words = text.split()

    repeated\_words = {}

    for word in words:

        count = words.count(word)

        if count > 1:

            repeated\_words[word] = count

    return repeated\_words

recognizer = sr.Recognizer()

from google.colab import files

uploaded = files.upload()

file\_name = next(iter(uploaded))

mp3\_audio = AudioSegment.from\_mp3(file\_name)

wav\_file = "uploaded\_audio.wav"

mp3\_audio.export(wav\_file, format="wav")

try:

    with sr.AudioFile(wav\_file) as source:

        audio\_data = recognizer.record(source)

        print("Recognizing...")

        text = recognizer.recognize\_google(audio\_data)

        print("You said:", text)

        stuttered\_words = identify\_stuttered\_word(text)

        if stuttered\_words:

            most\_stuttered\_word = max(stuttered\_words, key=stuttered\_words.get)

            print("Most stuttered word:", most\_stuttered\_word)

        else:

            print("No stuttered words found.")

except sr.UnknownValueError:

    print("Sorry, could not understand audio.")

except sr.RequestError as e:

    print("Error:", e)

os.remove(wav\_file)