Steven Rud and Daniel Mints COSI 136a Corpus creation and ASR system Assessment 1

To run the code please download the file called mp3s and original_text .That's the only two file needed to run the code

In this assignment we analyzed the Bible verses from Matthew 01 to Matthew 12 (approximately 60 min of audio) in the New Testament in Ukrainian. We meticulously prepared the audio file for the purpose of creating an Automatic Speech Recognition (ASR) corpus. This report will outline the methods we used in the code to collect, process, and analyze the corpus, including: the extraction of audio features, transcription, alignment, combination of transcriptions, and metadata extraction.

Firstly, for the Corpus creation we converted the given mp3 file from the audio recording of the bible verse into a wav audio file for further analysis. Next, we began the audio processing.

Audio Processing

We first transformed the original big chunks that were around 5 minutes to be **16000hz** and **single channel**. The python code does resampling of WAV audio files to a uniform sample rate. It uses the librosa library to load each file from a specified source directory, converting multi-channel audio to mono and then resamples the audio to a target sample rate of 16000 Hz. The sound file library is employed to save the resampled audio into a designated output directory. This process is iterated for all WAV files in the source directory. After that, we split the 5 minute long wav files into small 5 second long wav files. The segmentation process involves breaking down the original audio file into smaller, more manageable chunks. The python code segments WAV audio files into smaller chunks based on periods of silence using the pydub library. It processes each file in a specified directory, dividing the audio into 5-second chunks where silence longer than 700 milliseconds and quieter than -40 dB is detected. These chunks are then saved in a separate directory each named according to its parent file.

Final Analysis and Evaluation

Now we move onto comparing the created transcription with the original text. We have used the google API to transcribe the audio files for us. Even though it's not a golden standard, it worked very well and gave solid results. The python code automates the transcription of WAV audio files using the speech_recognition library. It processes each WAV file in a specified directory, converting the audio to text using Google's speech recognition service, with the language set to Ukrainian ('uk-UA'). The transcriptions are saved as text files in a separate directory, each named corresponding to its source audio file. The code is particularly useful for converting large batches of audio files into textual data, facilitating tasks like data analysis or content accessibility in different languages. After that we combine all into a CSV file where the first column states the wav file and the transcription states what the wav files contain. Then we compared the original text file with our transcription of it.

Our results were:

- Similarity for 1 file: 97.37%
- Similarity for 2 file: 97.44%
- Similarity for 3 file: 97.07%
- Similarity for 4 file: 97.34%
- Similarity for 5 file: 99.25%
- Similarity for 6 file: 98.51%
- Similarity for 7 file: 97.03%
- Similarity for 8 file: 98.23%
- Similarity for 9 file: 98.98%
- Similarity for 10 file: 99.14%
- Similarity for 11 file: 97.86%
- Similarity for 12 file: 98.32%

We then also found and determined the metadata of each audio file to include in the corpus. The metadata in the corpus included details about each audio recording, such as duration, sample rate, channels, language, age, gender, and accent of the speaker. It is vital to obtain these details as metadata extraction is crucial for understanding the characteristics of the dataset and may aid in developing more robust ASR systems. The metadata extraction process was implemented using Pydub and the Google Web Speech API.

In conclusion, The Ukrainian Bible Verse Corpus has been systematically prepared for ASR applications through a series of well-defined steps, including audio segmentation, transcription, text combination, and metadata extraction. The corpus is intended to serve as a valuable resource for the development and evaluation of Ukrainian language ASR systems. The provided code offers transparency into the preparation and analysis processes, facilitating reproducibility and further refinement of the corpus for future endeavors, including but not limited to the next assessment.

GitHub Repository: https://github.com/DanMint/ASR_Project1/tree/main