BTP-2

SPEECH TRANSCRIPTION VERIFICATION

Under the Guidance of Prof. Anil Kumar Vuppala

Mentor - Sai Aakarsh

Srujana Vanka - 2020102005
Shreeya Singh - 2020102011

OBJECTIVE

To create an efficient speech transcription validation system that optimizes accuracy by segmenting lengthy audio files into manageable segments, generating transcriptions for each segment, and providing users with a user-friendly interface for reviewing and refining transcriptions, with options to "edit", "save" or “discard” them.

UNDERSTANDING OF THE PROJECT

The conversion of speech into text plays a pivotal role in various applications, including automatic captioning, audio search, and speech recognition. However, transcribing lengthy audio files is often time-consuming and prone to errors. The accuracy of these transcriptions holds paramount importance for application effectiveness. Hence, there arises a crucial need for an efficient and dependable speech transcription verification system that can guarantee precision while minimizing the effort demanded from transcribers.

Our system addresses these challenges by streamlining the transcription verification process. Our primary objectives include improving transcription accuracy, reducing the workload on transcribers, providing flexibility for multiple languages, ensuring user-friendliness, and implementing robust security measures through user authentication.

TECH STACK

Audio processing: Implemented in Python to split an audio file into smaller segments.

Generating transcripts: Each audio chunk is converted into text using the Speech-to-Text API provided. Making POST requests to the API service.

Front-end: ReactJS for building the user interface of the application.

Back-end: Node.js for server-side logic and API development

Caching and Data Storage: LocalStorage API employed for client-side caching to store transient data, ensuring data continuity.

Other Libraries: Axios and Fetch API for save/discard, Pydub and WebRTC Voice Activity Detection (VAD) for audio chunking, react-router-dom for client-side routing and navigation, react-simple-keyboard for Telugu keyboard.

SIGNIFICANT PROGRESS AND FUTURE PLANS FOR OUR PROJECT

PROGRESS MADE SO FAR

User Authentication:

- We have successfully implemented a user login page, which not only enhances security but also helps in monitoring transcriber activities.
- This data can be valuable for minimizing inaccuracies and detecting malpractices.

Language Selection:

- Users can now choose the language of the audio file they want to transcribe.
- This makes our system versatile and capable of handling multiple languages.

Audio File Selection:

- Users can select specific audio files for transcription, providing customization.

Telugu Keyboard:

- We've expanded our application's language support by adding a Telugu keyboard, facilitating the verification of Telugu transcripts.
- This sets the stage for incorporating more languages.

Continuity:

- Users can log in and out while maintaining access to their work, ensuring a seamless workflow.
- This is achieved using efficient caching mechanisms (LocalStorage API)

Login Data Export:

- We've added a feature to download login data, allowing for easy access to a record of users who have accessed the system.
- This is achieved using efficient caching mechanisms.

Navigation Improvements:

- We've implemented a navigation system that empowers users to seamlessly navigate between audio chunks.
- With the addition of a user-friendly search bar, users can swiftly locate and access their desired transcript, eliminating the need for repetitive "Next" button clicks.
- This offers greater flexibility and control over their transcription verification process.

UPCOMING ENHANCEMENTS

Transliterator:

- Transliterator is used to represent or spell in the characters of another alphabet.
- We plan to introduce a transliterator to further simplify the transcription editing process.
- This tool will be a valuable addition for transcribers.

Language Expansion:

- Our goal is to extend language support to include Hindi and possibly more Indian languages, broadening the application's reach.

Verification Checks:

- Transcribers will receive immediate alerts if they attempt to move to another transcript without completing the verification of the current one.

Packaging:

- We aim to create a user-friendly deployment package, making it easy to implement our application.