



Submitted By:

Veda Samhitha Dyawanapally

Vigneshwar Muriki

Deepthi Gade

Mounika Kasaragadda

INTRODUCTION

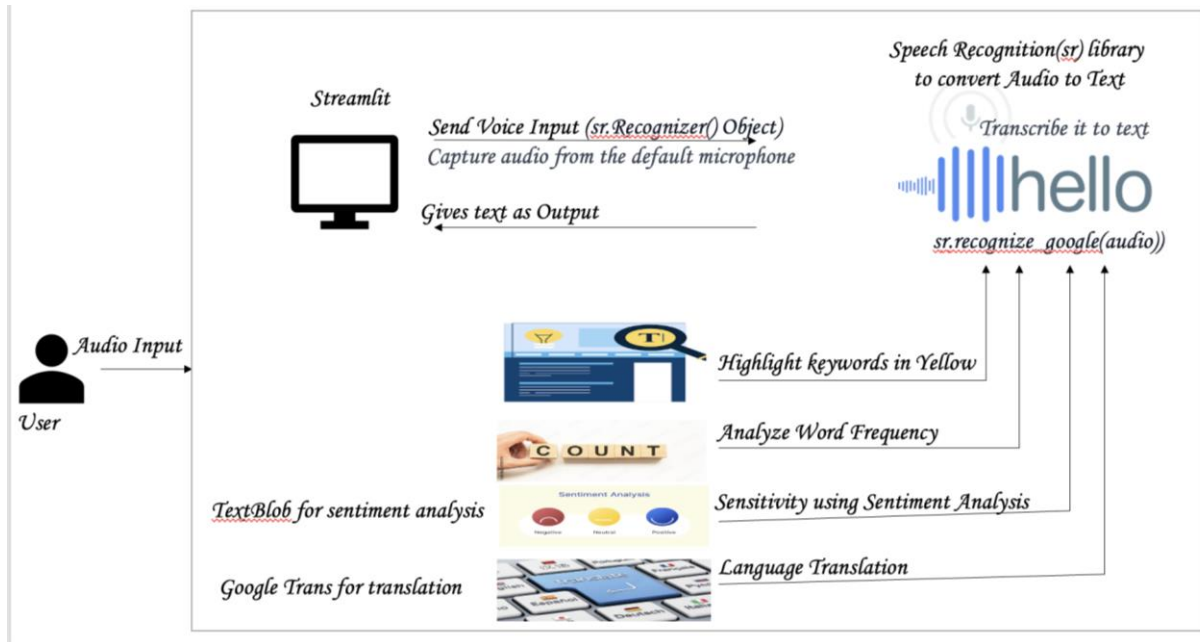
Speech to Text Recognition:

Automatic speech recognition (ASR), often known as speech to text technology, is the process of converting spoken words into written text utilizing algorithms and machine learning techniques. Many different sectors, including healthcare, banking, education, and telecommunications, use applications like virtual assistants, voice-controlled gadgets, transcription services, and dictation software. STT technology has significantly advanced as a result of advances in machine learning algorithms and natural language processing (NLP) techniques. The reliable transcription of many speakers, coping with regional accents and dialects, and recognizing non-standard speech patterns are still challenges. Despite these challenges, STT technology remains a valuable tool for a range of applications, enabling people to communicate and connect with technology in a more relaxed and natural way.

Application areas of this Technology:

- Siri, Alexa, and Google Assistant, among other virtual assistants, all use speech-to-text technologies.
- For meetings, interviews, and court procedures, audio and video recordings are converted into text using speech-to-text technology.
- For movies, television shows, and online videos, closed captions are produced using speech to text technology.
- Dictation software uses speech-to-text technology to let users dictate text directly into a computer.
- Speech-to-text technology is employed in the healthcare industry for medical transcription and dictation, speech therapy, and support with communication for those who are speech-impaired.
- Applications for speech to text technology are utilized in customer service, enabling businesses to provide voice-controlled self-service alternatives and to track client interactions.
- For the benefit of students with hearing problems, speech-to-text technology is utilized in the classroom to provide transcription and captioning for video and audio content.

Architecture:



Libraries Used:

- **Streamlit:** A python library used for creating Web-apps.
- **Speech-Recognition Library:** A python library used for speech recognition.
- **Textblob:** A python library used for natural language processing tasks including sentiment analysis.
- **googleTrans:** A python library used for google translation.

Implementation and Features

Streamlit is a popular Python framework that simplifies the process of building interactive data science applications. To implement Speech-to-Text using Streamlit, one can leverage existing libraries such as Speech Recognition, PyAudio, and pocketsphinx. These libraries provide speech recognition capabilities that can be easily integrated into a Streamlit app. One can start by creating a simple user interface using Streamlit's interactive widgets such as sliders, buttons, and text inputs. The user can then speak into the microphone, and the speech recognition algorithm converts the speech into text, which is displayed on the Streamlit app.

There are several additional features that can be added to a speech-to-text Streamlit application to enhance its functionality and user experience. Some of these features include:

- **Real-time recording:** This feature enables the software to continuously type out speech as it is being spoken, giving the user a direct feedback loop.
- Support for several languages can increase the app's adaptability and usefulness to a wider group of users by being able to identify and transcribe voice in several different languages.
- **Customized vocabulary:** When transcribing for fields or businesses, such as law or medicine, using a customized vocabulary can increase the accuracy of the transcription.
- **Presenter recognition:** Including speaker identification features can make it easier to distinguish between different talkers in a discussion and match each speaker's words to the appropriate transcript.
- **Additional formatting options:** Giving users the opportunity to highlight certain words or phrases in the transcription by bolding, underlining, or other formatting techniques might assist them concentrate on the most crucial portions of the transcript.
- **Controls for playback:** By including playback controls like pause, rewind, and play, users will be able to conveniently browse the transcription and go over particular passages.

Code:

The Python code demonstrates a basic Speech to Text (STT) implementation using the Speech Recognition and Streamlit modules. The code creates a simple internet application that allows users to record their voice and convert it to text using the Google speech recognition API. By pressing the "Start Recording" button, the app launches the speech recognition library and starts listening for audio input from the user's built-in microphone. The audio is then translated into text by the code, which then displays the text on the screen, using the Google speech recognition API. If there are any problems, an appropriate error message will be given. This code can serve as the foundation for the creation of more complex STT applications.

Home Page:

Real-time Speech to Text with Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

Select a language for translation:

Start Recording

Test 1: Start recording and get 'Speak now' message.

Real-time Speech to Text with Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

Select a language for translation:

afrikaans ▼

Start Recording

Speak now...

Test 2: Error check when no one talks.

Real-time Speech to Text with Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

Select a language for translation:

afrikaans ▼

Start Recording

Speak now...

Sorry, I cannot hear you

Stop Recording

Test 3: Network Failure

Real-time Speech to Text with Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

Select a language for translation:

afrikaans ▼

Start Recording

Speak now...

Sorry, there was an error processing your request: recognition connection failed: [Errno 11001]
getaddrinfo failed

Stop Recording

Feature 1: Keywords Highlighting.

The revealed A web application for text-to-speech conversion is Python script. A user interface is made using the Speech Recognition and Streamlit libraries. The software listens for audio input from the built-in microphone when the user presses the "Start Recording" button, and then it uses the Google voice recognition API to transcribe the audio. The script also inserts HTML code to alter the background color of specific terms in the transcription text to yellow, such as "important," "highlight," and "focus." The app displays the highlighted text in the Streamlit interface if the transcription is successful. The app displays the relevant message if the speech cannot be transcribed or if there is a problem processing the request. The statement `if __name__ == '__main__':` makes guarantee.

When certain words or phrases need to be highlighted to catch the user's attention or are particularly important, this feature may be helpful. For instance, a teacher might want to highlight important vocabulary words in the transcription text of a speech-to-text application for a language learning platform to aid their students in identifying and learning new words.

Feature 2: Detection of sensitivity of the output text using Sentiment Analysis.

The program is a Streamlit app that converts spoken words into text and analyzes sentiments in the text as it is being transcribed. The program uses the Text Blob library to assess the sentiment of the transcribed text and the speech Recognition library to translate the audio. The application asks the user to talk while it checks the built-in microphone for audio input. The app displays the transcript of the text along with the findings of the sentiment analysis. An appropriate error message is displayed if the voice recognition procedure fails.

The user experience can be enhanced by determining whether a text's sentiment is positive, negative, or neutral by conducting a sentiment analysis.

Feature 3: Language Translation

- Speech to text translation involves turning spoken words in one language into written text in another. The spoken words are converted into text using speech recognition technology, and the text is then translated into the target language using machine translation algorithms.
- Language learning, accessibility for people with hearing impairments, and real-time translation during multilingual meetings are just a few of the uses for services like Google Translate, Microsoft Translator, and Amazon Transcribe.

Feature 4: Analyze Word Frequency.

- In speech to text, word frequency refers to how frequently a specific word or phrase appears in each piece of spoken text that has been converted into written form. Word frequency analysis can offer insightful information about the text's content, style, and organization as well as the usage patterns and vocabulary of the speaker.
- Marketing research, sentiment analysis, and language learning are just a few of the many uses for word frequency analysis. To prioritize their learning efforts and increase their vocabulary more effectively, learners can use word frequency lists to identify the most frequently used words in a language.

Screenshot1:

Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

good

Select a language for translation:

telugu

Start Recording

Speak now...

Transcription: today's a **good** day

Sentiment: Positive

Translation: ఈ రోజు మంచి రోజు

Word Frequency:

today's: 1

a: 1

good: 1

day: 1

Stop Recording

Screenshot2:

Sentiment Analysis, Word Frequency Analysis, and Translation

Enter keywords to highlight (comma-separated):

bad

Select a language for translation:

hindi

Start Recording

Speak now...

Transcription: today's a **bad** day

Sentiment: Negative

Translation: आज का एक बुरा दिन

Word Frequency:

today's: 1

a: 1

bad: 1

day: 1

Stop Recording

Challenges:

- Changing the Project from '3D Human Modelling' to 'Speech-to-Text' after Iteration 3 and implementing it.
- Build dependency issues in Streamlit Cloud and Heroku.

Conclusion:

We have successfully implemented Speech-to-text Project using Streamlit Application. This application captures audio in real time and converts speech into text and gives output. Added additional features to the project such as Highlighting Keywords, Detection of Sentiment of Text using Sentiment Analysis, Frequency of words, Language Translation.

Note:

WebRTC is not required for Streamlit speech to text conversion. Streamlit provides a built-in component called STT (short for "speech to text") that uses the Google Cloud Speech-to-Text API to convert spoken audio into text.