An Adaptive Speech Recognition Tool with Multi-Platform Interface Support for Enhanced Accessibility

1. Dr. K. Little Flower
dept. of Computer Science & Engineering
St. Peter's Engineering College
Jawaharlal Nehru Technological University
Hyderabad, India
klittleflower@stpetershyd.com

2. Punna Sai Ganesh
dept. of Information Technology
St. Peter's Engineering College
Jawaharlal Nehru Technological University
Hyderabad, India
punnasaiganesh111@gmail.com

3. Bellamkonda Raghu
dept. of Information Technology
St. Peter's Engineering College
Jawaharlal Nehru Technological University
Hyderabad, India
bellamkondaraghu6@gmail.com

4. Vanjarapu Bhanu Charan
dept. of Information Technology
St. Peter's Engineering College
Jawaharlal Nehru Technological University
Hyderabad, India
vanjarapu.bhanucharan0123@gmail.com

***Abstract*— This speech recognition system proposed within this project has developed with ease and accessibility into the transcription of spoken words into text so that it would now be possible to both personal and professional uses in speech. The Google Cloud Speech API used in building this system provides both interfaces of user preference: Notepad and Web. The Notepad Interface interacts well with the Notepad application, so the transcriptions will appear and be saved as. txt files. The interface itself uses keyboard input simulation with pynput and pywinauto to interact with Notepad to provide a rather obvious choice for text-based document storage. The Web Interface would provide a facility to browser-based facility; users would be authorized to download a transcription file as a Portable Document format to make their transcriptions easier to download and exchange. The System Homepage would allow the user, according to his or her requirement, to switch easily between options between Notepad or Web Interface. Included within are audio preprocessing, feature extraction, and language modeling that power the main processes under Google Cloud Speech API for reliable high accuracy transcriptions. This double interface solution allows for a greater flexibility and precision in voice-to-text experiences to answer the range of accessibility and industry-specific requirements.**

Keywords— *Speech Recognition, Google Cloud Speech API, Accessibility, Dual Interface, Web Interface, Notepad Interface, Hidden Markov Model (HMM), Deep Learning, Voice-to-Text Conversion*

I. INTRODUCTION

It completely revolutionizes the accessibility of speech recognition technology with an accurate, easy to use solution for transcribing recorded or live speech to text by way of the Google Cloud Speech API. Two very disparate interfaces are available: with the Notepad Interface, transcriptions save directly as .txt files, while the Web Interface allows users to download their transcriptions as shareable PDF files. Proper and thorough design on the aspect of accessibility and convenience ensures good transitions from one interface to another, hence also applicable for work purposes or personal life. Whether you're transcribing meetings, lectures, or even just personal notes, powerful accuracy and flexibility may help streamline the process through the application. This kind of system comes with goodies such as audio pre-processing and language modeling, which can be good assurances of a quality outcome in the form of extremely powerful functions able to promise better efficiency in terms of productivity and access to and potentiality of communicating in various industries.

II. LITERATURE REVIEW

In [1] Stenman (2015), Stenman tests the performance of Google Speech under various conditions, background noise and accents. This work helps evaluate the real-world functionality of the Google Cloud Speech API and provides strategies for improving accuracy, especially in challenging environments and provides insight into potential upgrades for your system.

In [2] Matarneh, R., Maksymova, S., Lyashenko, V., & Belova, N. (2017), an aspect as comparative review of several speech recognition systems is given with respect to their accuracy, speed, and adaptability. Such comparison can be very useful for general assessment purposes against other comparable systems with the Google Cloud Speech API, which can support your decisions in your project.

In [3] Anggraini, N., Kurniawan, A., Wardhani, L. K., & Hakiem, N. (2018). The authors develop an application for speech recognition on the Android platform using Google Cloud Speech API to help persons with impaired speech. The demonstration demonstrated the utility of translating speech in terms of text, thereby taking forward communication for persons with speech disabilities. This research work forms a critical part of the cloud-based APIs in accessible technology and closely resonates with your project objectives of accessibility and conversion of speech to text.

In [4] Shakhovska, N., Basystiuk, O., & Shakhovska, K. (2019), authors develop a speech-to-text chatbot interface with Google API; they give a system in which the user engages with a conversational interface. This paper addresses the challenges and design issues applying speech recognition in such interfaces with very important insight in developing your web interface to be of accessibility and friendly use.

In [5], Jindam, S., Devaraju, G., Sindhu, T., Mudiraj, K. A., & Chittiprolu, A. (2024), the study explored the possibility of using Google Cloud Speech API with Notepad, similar to what you have for your Notepad interface. The research demonstrated that speech recognition indeed can be applied in text editors, which would make your Notepad integration possible and technically advise its development process.

III. .PROPOSED SYSTEM

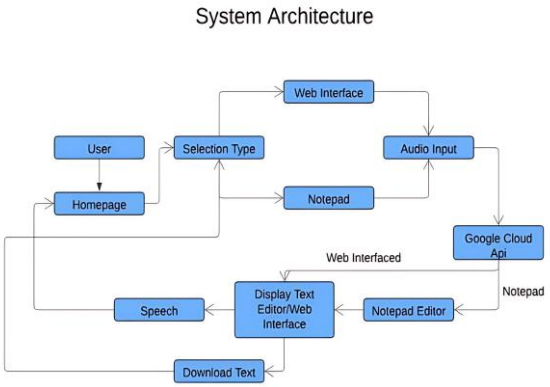
The proposed system, named Speech Recognition System Using Google API, is aimed at offering an efficient, accessible, and highly accurate solution for speaking-to-text conversion, so that it meets the requirements of diverse users with a dual-interface design. Advanced transcriptions using strong language models and deep learning techniques are used for handling complicated speech patterns, a variety of accents, and procedures with utmost accuracy through the integration of Google Cloud Speech API in the system. Recognizing flexibility and access, the system includes two different user interfaces: Notepad Interface and Web Interface.

Notepad Interface The Notepad Interface directly transcribes the spoken words to the .txt file using the Notepad application and the Python libraries of Pynput and Pywinauto. In doing so, the interface also provides a viewable, editable, and savable environment on the desktop for text, and this way users may quite easily translate any on-the-fly interchanges offline or locally. Alternatively, with Flask, was developed the Web Interface which permits users to access the system through a web browser from which they can see a transcription that is effortlessly downloadable as a PDF. Access from multiple devices also saves time on convenient sharing and portability of transcribed content, thereby being accessible to professionals and students who may require digital access to copies of their documents.

This ensures reliable performance even in difficult acoustic conditions, combining deep learning frameworks and Hidden Markov Models (HMM). HMM-based features are extracted from the audio inputs but LSTMs for deep learning models enhance the accuracy, capturing complex patterns of spoken language with reducing transcription errors even from multiple backgrounds. To access this device, interaction is automated and manual input is at a minimum; hence, it's accessible even to users physically or speech-impaired, while convenience and inclusivity are emphasized.

Overall, this dual-interface setup of the proposed system enables selection of the format and device based on the needs of the users in a flexible yet accurate manner. This design will portray Speech Recognition System Using Google API as an everyday and educational tool both at personal and professional levels that addresses given drawbacks of speech-to-text technology and adapts towards a healthy and efficient speech transcription in lots of different usage cases.

IV. SYSTEM DESIGN



The Google API-based Speech Recognition System is modular and scalable in design, providing robust and flexible speech-to-text capabilities; the following represent the core design features

1. Dual Interface Options: The software offers two user interfaces, the Notepad Interface and the Web Interface, while the former provides the opportunity for users to select the preferred output format. The Notepad Interface, integrated with Pynput and Pywinauto, simply writes speech on a paper or text file created directly within the Notepad application; hence, the produced documents are stored locally. The Web Interface, designed using Flask, posts the produced transcription online, and through a simple click of the Download button, users can convert them into PDF files.

2. Google Cloud Speech API Integration: At the core of the system is the Google Cloud Speech API, which processes audio and provides very accurate transcriptions of it. It is a combination of Hidden Markov Models and advanced deep learning algorithms it uses, like LSTMs, for managing complex patterns and noise with diverse accents.

3. Audio processing and Pre-processing: The system records the audio with a microphone and performs preliminary work to achieve high clarity without noise. The result from this is that the audio stream would be good and accurate for transcription by the API.

4. Output Flexibility and Accessibility: This design provides flexibility so that the user can toggle between the different formats of output for better suitability. It stresses accessibility through features like automation and easy interface options suitable for users who are also speech or physically challenged.

5. User-Centric Design: Having flexible options for transcription, the system is valuable for personal, educational, as well as professional settings and guarantees efficient, accurate conversion of speech-to-text for wide-ranging applications.

The system architecture also supports future scalability along with easy enhancements such as multi-language support, mobile integration, and other accessibility features.

V. RESULTS

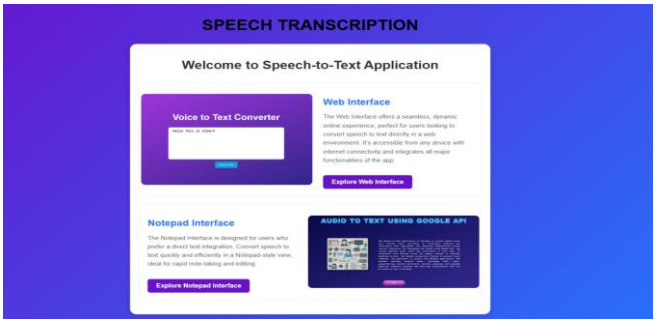


FIG-1 HOME PAGE

FIG-2 WEB INTERFACE TEXT DISPLAY

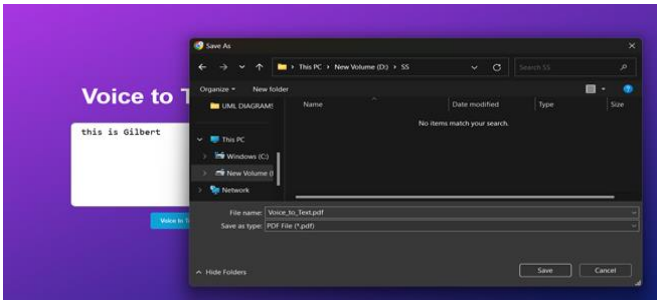


FIG-3 PDF DOWNLOAD PAGE

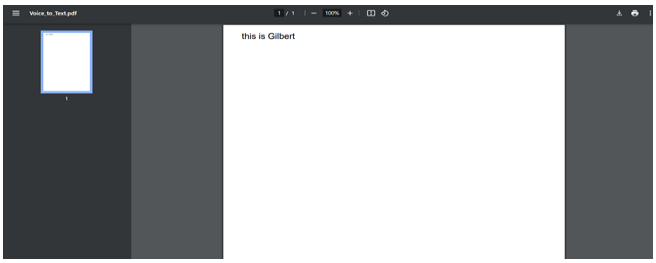


FIG-4 PDF PRESENTATION PAGE

FIG-5 NOTEPAD INTERFACE TEXT DISPLAY

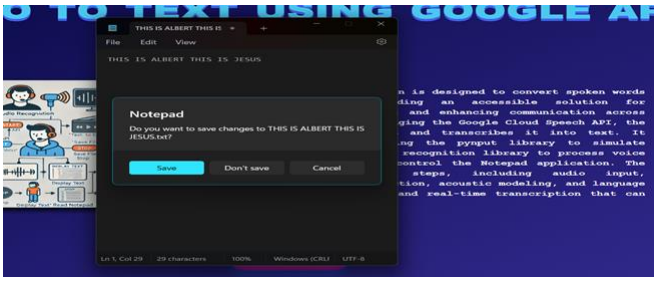


FIG-6 NOTEPAD SAVE PAGE

Speech Recognition System Using Google API, has been designed as a two-interface application that converts speech to text. It employs the use of the Google Cloud Speech API for its speech recognition. Pynput and Pywinaauto handle the audio input and automated interface. The web interface is powered by Flask. The Notepad interface saves the text into the converted system as a .txt file, whereas the Web version enables users to download the text in PDF format. Accessibility has been considered specially while designing, especially for speech-impairing users who would make full use of it, and built with integration of both HMM and Deep Learning models for perfection. The expandability scope includes adding features like multi-language support, mobile integration, and noise cancelling

.VI. CONCLUSION

In conclusion, the "Speech Recognition System Using Google API" successfully translates speech into text from two interfaces, accessible to the users, even with speech defects. The adoption of latest technologies such as the Google Cloud Speech API, Pynput, and models used for machine learning like HMM and Deep Learning contributes towards the precise speech recognition system. With the combination of Notepad and Web interfaces, it is possible for the users to save the output text or even download it. So, with some possible improvements, like multi-language support and noise cancellation, which can be included in the future, it has vast scope for the growth and more practical applications in real life.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to **Dr. K. Little Flower** for her invaluable guidance and unwavering support throughout this project. We are also thankful to the Head of the Department of Computer Science and Engineering for timely assistance, as well as, We are extremely thankful to our Principal **Dr. K. Sree Latha**, who stood as an inspiration behind this project and heartfelt for her endorsement and valuable suggestions, for her inspirational leadership. We respect and thank our secretary **Sri. T. V. Reddy** for providing us the opportunity to undertake this project at **St. Peter's Engineering College.** Lastly, we express our deep thanks to our parents and friends for their moral and financial support, which was crucial in our successful completion of this work..

REFERENCES

[1] Stenman, M. (2015). Automatic speech recognition An evaluation of Google Speech.

[2] Matarneh, R., Maksymova, S., Lyashenko, V., & Belova, N. (2017). Speech recognition systems: A comparative review.

[3] Anggraini, N., Kurniawan, A., Wardhani, L. K., & Hakiem, N. (2018). Speech recognition application for the speech impaired using the android-based google cloud speech API. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *16*(6), 2733-2739.

[4] Shakhovska, N., Basystiuk, O., & Shakhovska, K. (2019, May). Development of the Speech-to-Text Chatbot Interface Based on Google API. In *MoMLeT* (pp. 212-221).

[5] Jindam, S., Devaraju, G., Sindhu, T., Mudiraj, K. A., & Chittiprolu, A. (2024). 357 Speech Recognition System Using Google Cloud API with Notepad Integration. *Multifaceted approaches for Data Acquisition, Processing & Communication*, 43.

[6] Putthapipat, P., Woralert, C., & Sirinimnuankul, P. (2018, January). Speech recognition gateway for home automation on open platform. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)* (pp. 1-4). IEEE.

[7] Siegert, I., Sinha, Y., Jokisch, O., & Wendemuth, A. (2020). Recognition performance of selected speech recognition apis–a longitudinal study. In *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings 22* (pp. 520-529). Springer International Publishing.