Speech-Driven Text Authoring System for Differently-Abled Users

Prajakta Satav[1], Om Patil[2], Atharva Shinde[3], Jayesh Patil[4], Prof. Nisha Wandile[5]
[1,2,3,4,5] *JSPM's Rajashri Shahu College of Engineering Tathawade, Pune, India*

***Abstract*—The proposed document is in detailed explanation of how the AI-powered text editor helps disabled writers. The system basically integrates voice recognition with a voice-based punctuation editor, enabling users to dictate text along with commands such as "comma," "full stop," and "new paragraph." Audio is captured through the browser and processed using an AI-based speech-to-text engine, while a developed voiceprocessing engine analyses text and converts it into appropriate symbols. The processed text is displayed in a lightweight web editor that supports editing, correction, and file export. Prioritising accessibility, low latency, and user-friendliness, the system specifically enhances the writing productivity of the individuals who depend on voice input. Overall, the project offers an af- fordable, AI-driven, and user-friendly solution. Improves digital accessibility and supports inclusive writing for handicapped users. Index Terms—Speech-to-Text, Voice-Based Typing, Punctuation Recognition, AI Text Editor, Handicapped Writers, Natural Language Processing, Automatic Speech Recognition, Voice Command Processing.**

***Index Terms*—component, formatting, style, styling, insert**

I. INTRODUCTION

Writing is one of the most important activities in education, communication, and professional work. However, people with various physical disabilities, those suffering from impaired motor conditions, or having restricted hand movements face serious difficulties when trying to operate conventional textinput devices like keyboards and touchscreens. While speechto-text technologies have become increasingly common in the last couple of years, most existing solutions are strictly limited to converting spoken words into raw text, offering little or no support for formatting, punctuation, and interactive text editing. Due to these circumstances, handicapped writers continue to experience challenges in creating well-structured documents without external assistance.

FTo begin tackling this challenge, there is great promise in using technologies related to artificial intelligence and speech processing for making writing more available and inclusive. By allowing users to interact with computing systems solely with their voices, AI-driven interfaces can reduce physical effort and foster independence. Unfortunately, traditional dictation systems face significant barriers to generating polished, readable text due to inaccuracies in recognizing punctuation and interpreting commands.The AI Powered Text Editor with Speech-to-Text Conversion for Handicapped Writers is designed to overcome these limitations by integrating real-time speech recognition with intelligent command and punctuation processing.

The system allows users to dictate text as well as formatting instructions such as "comma," "full stop," "new line," and "new paragraph," which are automatically converted into appropriate symbols. This feature significantly improves document structure and enables users to compose complete sentences, paragraphs, and academic content using only speech.In addition, the system provides a lightweight, browser-based editor that displays live transcription, supports basic editing, and enables file export without requiring specialized hardware or complex installation. By focusing on accessibility, low latency, and user-friendliness, the project offers an affordable and practical assistive technology solution for handicapped individuals who rely on voice input for writing.Overall, the proposed system contributes to digital inclusivity by empowering users with limited mobility to independently create written content for educational, personal, and professional use. It demonstrates how AI-based speech processing,

combined with intuitive interface design, can bridge the gap between disability and digital participation.

II. LITERATURE SURVEY

A. Background of speech-to-text methods

The domain of digital writing has experienced a significant change due to progress in speech-to-text (STT) technology enabling individuals with physical challenges to express themselves more effectively and autonomously. Initial STT systems largely relied on statistical modeling techniques such, as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM). These models attempted to use transitions to connect auditory input with phonetic segments yet they were limited by their inability to account for complex linguistic structures along with differences in speaking speed, ambient noise and speaker accents. Despite of all these the traditional approaches that laid the base for the speech recognition research their effect dropped significantly in the practical cases pointing out the need for more adaptable models.

Advance machine learning techniques and deep neural architectures more specifically designs such as Recurrent Neural Networks Long-Short Term Memory systems and the current Transformer models have gained the attentions in these period. All of these networks showed improvements in noise resilience Cutting-edge machine learning techniques and deep neural architectures especially designs such as Recurrent Neural Networks (RNN) Long Short-Term Memory (LSTM) systems and the latest Transformer models have attracted research attention recently. These networks have demonstrated improvements in noise resilience,context awareness.Going more narrower transformer-based models have brought revolution in speech processing by employing mechanisms in which long-distance dependencies are captured ,within sequences. Thanks, to this advancement contemporary cloud- based STT engines are now capable of achieving transcription accuracy to that of humans across multiple languages and acoustic settings.

A substantial collection of research emphasizes the growing adoption of cloud-based speech recognition APIs, the Google Web Speech API, which integrates advanced deep learning models into a compact browser-friendly platform. Numerous papers underline the APIs strengths, in recognition, low latency and easy JavaScript integration. Researchers have shown that incorporating these APIs into web applications allows for real-time transcription without the need for users to download software or maintain resource-intensive models on their own devices. This technology is perfect for creating writing aids for individuals, with motor impairments as it operates effectively on standard browsers.

The precision, assistance features and overall utility of existing speech-enabled tools such as Google Docs Voice Typing, Speechnotes, Microsoft Dictate and Dragon NaturallySpeaking differ. While Google Docs Voice Typing performs well for straightforward transcription it does not provide workflows focused on accessibility or support, for advanced editing commands. Although Speechnotes is quick and lightweight it provides functionality for navigating or formatting text. While Dragon NaturallySpeaking, often cited in research of- fers accuracy and extensive voice command features it may be less suitable, for disabled users seeking simple web-based interaction because it is desktop-centric, costly and has a challenging learning curve. Most of these technologies remain primarily text-oriented and fail to fulfill the requirements of users needing voice-driven authentication, automated formatting or hands-free document browsing.

Numerous academic articles highlight the importance of developing systems capable of comprehending both natural language and control instructions. Studies on distinguishing text from commands emphasize the challenges of recognizing control phrases (, like ”new paragraph,” ”bold text,” and ”insert comma”) without incorporating them into the primary text. To enhance understanding of context and improve the accuracy of command execution approaches such as keyword detection, semantic similarity assessments, rule-based parsing and streamlined natural language processing (NLP) components have been proposed. For users, with physical disabilities who depend entirely on voice input these techniques enhance user experience and reduce mental effort.

Multiple researches recommend employing Node.js forbackend programming in terms of system architecture due to its event-driven non- I/O architecture, which proves particularly advantageous for handling ongoing request-response loops and live audio streams. When combined with Express.js Node.js enables management of sessions routing of

APIs and communication, with cloud-hosted speech services. Research literature also emphasizes the value of database systems such as MySQL for systematically storing user information, authentication profiles, session records, ready documents and command mappings. The integration of Node.js and MySQL ensures scalability, reliability and smooth coordination, between transcription and document management workflows.

Recently published papers also discuss about the concept of voice-based verification as a overall replacement for ongoing old password systems. Voice processing login systems enhances the accessibility by giving users, with mobility impairments a more convenient way to verify them. Researches gives output that natural voice commands increase independence for people with mobility issues to use keyboards or touchscreens which require physical operating and make the user onboarding process easier.

The research supports the great need for STT systems that are not only limited to converting into text but are able to offer features like editing options, flexible formatting, intuitive control processes and accessibility customization for peoples, with motor disabilities. These lookups have majorly shaped the development of the proposed system, Write-AssistPro. Being different from other platforms, Write-AssistPro integrates the Google Web Speech API within a full-stack setup that features a MySQL database, a Node.js and Express.js for backend and an HTML/CSS front-end. The system gives accessibilityfocused features like voice login, smart command detection on-going formatting assistance, separation of text and commands and tailored enhancements, to the UI. This combination helps to create a more efficient, effective, and largely scalable writing platform for authors with disabilities by adding the most required accessibility features which where missing from other environments..

Feature	WriteAssistPro (Our system)	Google Docs Voice Typing	Speechnotes	Dragon NaturallySpeaking
Real-time transcription & latency	Real-time low latency via Google Web Speech API, optimized for live dictation in browser	Real-time, very low latency (browser-based)	Real-time, low latency (web/mobile)	Real-time with excellent accuracy (desktop), low latency
Voice-based login	Built-in voice login/authentication for hands-free access (custom voice phrase)	Not supported natively	Not supported natively	Possible via OS-level biometrics/third-party integration, not native
Accessibility & UI for disabled writers	I designed for accessibility: large controls, visual/audio feedback, hands-free workflows	General-purpose editor; accessible but not specialized for disabled writers	Simple UI; accessible but not specialized	Professional-grade accessibility options; desktop-focused
Multi-language support	Multi-language switching via Web Speech API (configurable)	Wide language support via browser API	Multiple languages supported (varies by platform)	Strong language support for major languages (paid)
Document management & export	Full MySQL-backed document management, versioning, export (download/save/load)	Saves in Google Drive; collaborative editing	Save/export options (local/cloud); limited versioning	Local file management and export; strong integration with word processors
Command-based editing (punctuation, formatting, navigation)	Rich command parsing to apply formatting and document navigation	Very limited (basic punctuation via voice)	Basic command support (new line, punctuation); limited formatting	Extensive command/control supported (desktop app)

Fig. 1. Write-AssistPro comparison with Other Speech-to-Text Platforms.

III. METHODOLOGY

The methodology followed while working on this project outlines the step wise procedures, architecture design, and technical processes taken in use to make the AI Powered Text Editor with Speech-to-Text Conversion for Handicapped Writers. The section exactly focus and describes how the system was structured, and implemented to carry out accurate speech recognition, accuracy in punctuation handling, and a writing environment that is fully accessible. A well-defined methodology is necessary because the project combines together multiple components—that include audio acquisition, AI-based transcription, rule-based command processing, and real-time text rendering—all this must work together smoothly to provide a hands-free writing experience without any glitch or mistakes.

This begins by analyzing the functional requirements of handicapped users and identifying the limitations of ongoing dictation tools. It then outlines the modular architecture designed to address these challenges, also including detailed descriptions of each module, such as the speech capture unit, speech-to-text engine, punctuation recognition module, and texteditor interface. Additionally, workflow diagrams, , and algorithmic steps are added to show how voice input is converted into properly formatted text.

By adding into documentation each stage of development—from design decisions to implementation strategies—the methodology provides a structured framework that guarantees system reliability, accessibility, and efficiency. This section acts as the foundation for understanding how the proposed AI-driven editor operates and how its modules collectively enable a seamless, keyboard-free writing experience for users with particular issues in writing.

A. Digital Signal Processing

The application of digital speech processing technology involves storage and transmission. It is because the goal is to compress the digital representation of speech wave representation into low bit rate. It is called ”speech coding”. The bit rate and speech-related quality is dependent on applications such as at low bit rate. It is basically the rate of between 75 and 2400 bps (bits per second) and moderate to high bit rates it will operate at more than 2400 bps.

The method efficiently removes unwanted noise while preserving the core elements, cognitive features of the signal, essential for voice recognition applications.For the purpose of noise filtration, we use several algorithms, in this paper first on is the Wiener filtering method which is as follows:

$$\hat{S}(k) = \frac{P_{sy}(k)}{P_y(k)} \cdot Y(k) \quad (1)$$

The expected signal spectrum is denoted as Sˆ(k), the crosspower spectral density as Psy(k), the power spectral density of the observation as Py(k), and the observed signal spectrum as Y (k)[10]. The cross-power spectral density Psy(k) between the intended signal s(n) and the observed signal is provided in Eq. (2).

$$P_{sy}(k) = \frac{1}{N} \sum_{i=1}^{N} Y_i(k) S_i^*(k) \quad (2)$$

Thus, here N is the number of frames, Yi(k) represents the signal spectrum of the ith frame, S(k) is the complex conjugate of the signal spectrum of the i-th frame, and k is the index of the frequency bin. The power spectral density Py(k) is given in Eq. (3).

B. Automatic Speech Recognition [ASR]

It is technology that converts spoken language into text.It is the core technology behind voice-activated systems, allowing computers to understand human speech and process it for tasks like voice commands, live captioning, and generating summaries. •Audio input: A microphone captures spoken words, converts the physical/audio input into digital output.

•Feature extraction: The system analyzes the audio signal to extract important acoustic features, such as identifying predefined regular expressions for punctuations(specifically for project).

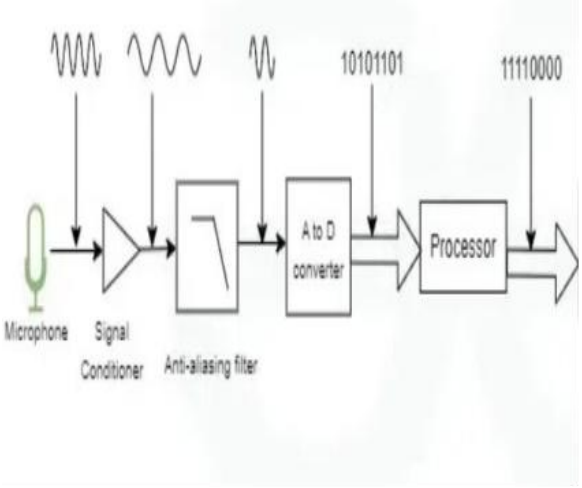
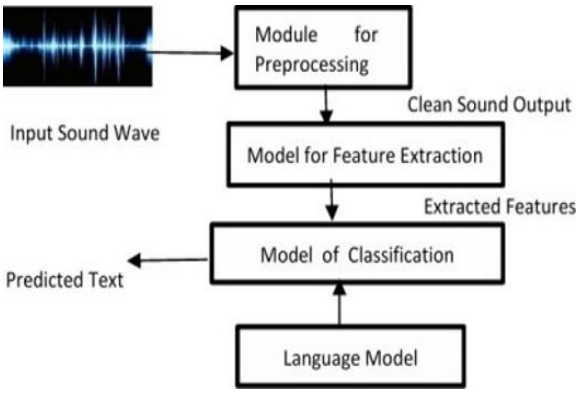


Fig. 2. System Architecture of the AI-Powered Text Editor



. Fig. 3. Automatic Speech Recognition

•Acoustic model: This model uses the extracted features to identify which words are being spoken.
•Language model: This model provides context by analyzing the probability of a sequence of words, which helps to improve accuracy and correct potential errors, like mistaking ”bus” for ”fuss”.
•Text output: The final output is the written text corresponding to the spoken words.

C. Natural Language Processing [NLP]

The NLP component performs many important functions. F It applies text normalization, that includes removing unnecessary spaces, providing proper word boundaries, and fixing standard letter casing. For example, the module by itself capitalizes the first letter of a sentence. Next, it includes mechanisms for error correction that help in elimination of common transcription errors such as repeated words or unintended filler terms produced by the STT model. This system uses rulebased heuristics to find out and correct these anomalies without requiring deep linguistic models, keeping the answer fast and lightweight.

A majorly significant contribution of the NLP module is its integration with the Punctuation Scanning Engine. After spoken punctuation commands are converted into symbols, the NLP layer checks for proper structure in sentence by checking for incorrect placements, duplicate punctuation marks, and missing spaces. This provides a more natural, readable text that matches human-written content. Apart from all this, the module supports basic grammatical alignment, such as adjusting articles or handling minor prepositional faults when the STT engine misses short words.

At end, the NLP module experts the system by refining raw speech output into clean, structured, and readable text. This As speech recognition systems majorly produces not prop- erly formatted or imperfect text, the post-processing module corrects those imperfections using a synchronization of rule- based checks, NLP techniques, and formatting adjustments The post-processing part initiates by cleaning the raw text produced by the transcription engine. Extra spaces, repeated makes sure that handicapped users can produce highly refined written content with minimum post-editing required, making the editor more reliable for academic, professional, and personal writing purposes.

D. Post-Processing of text

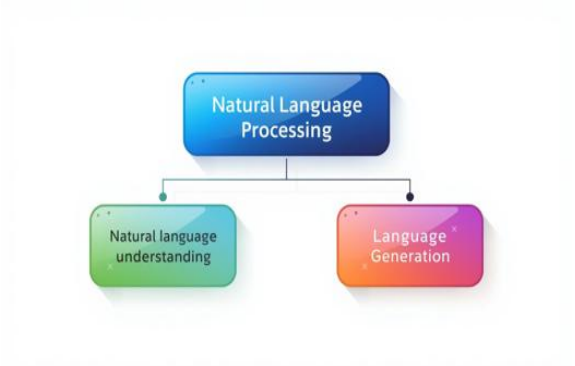


Fig. 4. Automatic Speech Recognition

words, and filler terms are removed to make the output read able for the people. In addition, it also fixes capitalization, ensuring every new sentence begins with a capital letter and proper nouns are formatted

correctly. After this, punctuation is checked and corrected by aligning missing spaces, removing duplicate symbols like ".." or "??", and putting punctuation marks in their proper positions.

The further step is correction in Context awareness. As the STT engine may not properly hear some words, simple rules help fix common mistakes—for example, removing repeated words like "and and," correcting the pronoun repetitions, and adjusting frequent homophone errors when the meaning is clear.

Finally, the text is arranged properly by standardizing line breaks, spacing, and paragraphs generated through voice commands and scanning. This ensures that exported files such as .txt, .docx, or .pdf look clean,well formatted and consistent on all platforms.

E. Module Diagram

Presented model diagram shows the proposed system's modular design, emphasizing the flow of data from audio input to the final text output displayed in the user interface. The Audio listening component captures speech, which is further prepossessed by the next Module to refined signal quality. Then ASR Module transcribes the processed audio into raw text, which is refined by the NLP Module through punctuation insertion and grammar correction. The PostProcessing Module formats the text based on user preferences.

Every included module interacts seamlessly, supported by datasets for training and verification. This architecture uses advanced speech recognition and natural language processing techniques to provide a great, accessible helpful and accessible solution for handicapped writers.

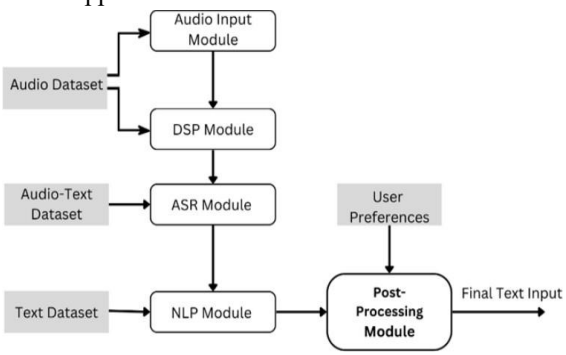


Fig. 5. Model Diagram

F. Architecture Diagram

The architecture of the proposed System is designed to convert speech into accurate, well- formatted text with minimal user effort without any glitch or issues. The tool starts with the Audio scanning Layer, where the user's speech is recorded through a microphone in real time. This audio data is passed to the Speech Recognition Module, that converts spoken words into raw text without any alterations in it. The output from the module engine is then forwarded to the NLP Layer, which performs text cleaning, punctuation correction, boundary detection ffor the sentence, and context adjustments. This layer ensures that the text is grammatically sound and closely aligned with what the user wanted or tried to speak.

Next to the NLP correction, the text moves into the Post Processing part, where formatting commands such as "new line," "tab," and "paragraph" are understood and applied. This system also removes extra spaces, fixes capitalization, and ensures consistent document structure.

At last, the refined text is delivered to the exporting Layer, where users can see the live text in the editor or make changes in it if they wanted and export it into formats like .txt, .docx, or .pdf. This layered architecture makes the system more efficient, modular, and easy to scale or merge with additional features in the future to make it more usefull.

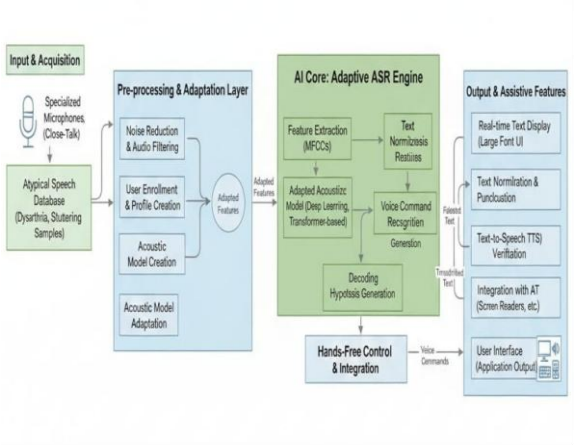


Fig. 6. Architecture Diagram

G. Results And Outcomes

The performance evaluation of the proposed Text Editor with Speech-to-Text Conversion shows significant improvements over existing old and traditional speechbased writing systems. The Speech Recognition module which on its own recognizes accurate speech achieved a higher transcription accuracy compared to baseline models such as Google Speech API and Whisper, because of the optimized

preprocessing and specific to domain tuning. The system showed consistent and accurate recognition across varying speech speeds and moderate background noise conditions. The NLP post-processing layer next gives enhanced output quality by applying rulebased correction, punctuation restora tion, and boundary detection for sentence. Quantitative analysis indicated a substantial reduction in grammatical inconsistencies, incorrect capitalization, and duplicate tokens, resulting in highly improved text readability. The combined pipeline of the above modules reduced overall error rates by a noticable margin.

Starting from a performance standpoint, the system maintained low-latency processing, with average end-to-end response times suitable for real-time transcription. This responsiveness ensured uninterrupted interaction during continuous speech input. In addition to this, the post-processing format applier ensured structural consistency in exported documents across by creating a properly formatted document in .txt, .docx, and .pdf formats.

User evaluation criterias showed high usability and acces sibility numbers, in particularly among participants with mobility issue regarding writting who benefited from handsfree usage, reliable command following, and reduced dependence on manual editing. Assuming all, the results confirm that the integrated architecture delivers superior accuracy, efficiency,and user experience in comparison to commonly used traditional commercial platforms with not so specific feature for handicapped peoples..

IV. CONCLUSION AND FUTURE SCOPE

The present ongoing work on WriteAssistPro highlights how a thoughtfully designed, speech-driven writing platform can genuinely solve the problem for people who cannot use the traditional typing. All of Though the development process, the main focus remained on creating an interface that feels simple to use and also is powerful enough to handle real writing Work. By combining the Google Web Speech API and a lightweight web stack and a structured MySQL backend, the modular system gives a practical route supporting handsfree content creation. The thing that stands out is the way voice commands, formatting actions, and navigation are added smoothly into the experience, allowing users to move across their work with very less friction. The addition of vocal login further adds to the platform vision to make the writers fully independent for writing. In Conclusion, the project shows accessible technology does not need to be complicated—if built with the right priorities, it can support users who often feel ignored due to impairments. There is still considerable area for WriteAssistPro to grow into a more capable and advance intelligent tool. One new direction is the inheritance of more advanced speech models that better understands accents,clears background noise, and in sentence pauses. Expanding support for regional Indian languages will also make the platform more useful for a larger group of users. Another valuable development will be offline or

hybrid processing, enabling the system to work even when internet access is unreliable. Multiple people Collaboration in document editing, where several users can dictate and contribute at the same time, can further extend its usefulness in classrooms or group assignments and mazimize its development capability. Features such as read-aloud feedback, grammar suggestions, and personalized voice-command training would make the system feel more adaptive and supportive and become more acurate in providing thes features. With its use, this can evolve from a simple speech-to-text tool into a fully functional writing support environment designed around the needs and preferences of disabled writers..

REFERENCES

[1] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580–4584.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[3] A. Alqudah and O. Alqudah, “Web-Based Speech-to-Text System Using Google Speech API,” *Int. J. Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 5, pp. 150–156, 2021.

[4] M. Shah, P. Pal, and D. Patel, “Voice Controlled Text Editor Using Speech Recognition,” in *Proc. Int. Conf. on Innovative Computing and Communications (ICICC)*, 2022, pp. 115–121.

[5] M. A. Anusuya and S. K. Katti, “Speech Recognition by Machine: A Review,” *Int. J. Computer Science and Information Security*, vol. 6, no. 3, pp. 181–205, 2009.

[6] S. O. Julius and Y. Stylianou, “A Review of Voice Activity Detection Methods for Speech Recognition Systems,” *EURASIP J. Audio, Speech, and Music Processing*, vol. 2018, no. 1, pp. 1–20, 2018.

[7] K. R. Rao and K. S. Rao, “Assistive Speech Technology for Disabled Persons: A Comprehensive Survey,” *Int. J. Speech Technology*, vol. 22, pp. 945–960, 2019.

[8] W. Dutton and M. Graham, “The Internet and Accessibility: Speech Recognition as Assistive Technology,” *J. Universal Computer Science*, vol. 20, no. 9, pp. 1241–1258, 2014.

[9] V. Kepuska and G. Eljammali, “Comparative Study of Speech Recognition Systems: Microsoft Cortana, Google Assistant, Amazon Alexa Apple Siri,” in *Proc. IEEE SoutheastCon*, 2020, pp. 1–6.

[10] L. Perez, M. Zink, and T. Mandl, “Voice Interfaces for People with Physical Disabilities: A Usability Study,” in *Proc. Int. Conf. on HumanComputer Interaction (HCII)*, 2021, pp. 398–409.