I.J. Modern Education and Computer Science, **2025, 4, 70-81**
Published Online on August 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.04.05

Developing Audio-to-Text Converters with Natural Language Processing for Smart Assistants

Pratistha Tulsyan
Bachelor of Computer Science, School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, 632014, India
E-mail: pratistha.tulsyan2021@vitstudent.ac.in

Mareeswari V.*
Department of Software and Systems Engineering, School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, India
E-mail: vmareeswari@vit.ac.in
ORCID iD: https://orcid.org/0000-0002-2768-943X
*Corresponding Author

Vijayan Ramaraj
Department of Information Technology, School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, India
E-mail: rvijayan@vit.ac.in
ORCID iD: https://orcid.org/0000-0002-1164-7569

Suji R.
Bachelor of Computer Science, School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, 632014, India
E-mail: suji.r2021@vitstudent.ac.in

Received: 09 October, 2024; Revised: 27 April, 2025; Accepted: 16 May, 2025; Published: 08 August, 2025

Abstract: In recent years, smart assistants have transformed human interaction with technology, offering voice-controlled interactions like music playback and information retrieval. However, existing systems often struggle with accurately interpreting natural language input. To address it, this proposed work aims to develop an audio-to-text converter integrated with natural language processing (NLP) capabilities to enhance interactions of smart assistants. Additionally, the system will incorporate intent recognition to discern user intentions and generate relevant responses. The proposed work commenced with a literature survey to gather insights into existing smart assistant systems. Based on the findings, a comprehensive architecture was designed, integrating NLP techniques like tokenization and lemmatization. The implementation phase involved developing and training a Feedforward Neural Network (FNN) model tailored for NLP tasks, leveraging Python and libraries like TensorFlow and NLTK. Testing evaluated the system's performance using standard evaluation metrics, including Word Error Rate (WER) and Character Error Rate (CER), across various audio input conditions. The system exhibited higher WER and CER with accented speech (15.3% and 7.9% respectively) while the clean audio dataset produced WER and CER of 4.7% and 2.55% respectively. The proposed work also involved training the FNN model while monitoring training loss and accuracy to ensure model performance. Ultimately, the model achieved an accuracy of 97.62% with training loss reduced to 1.45%. Insights from the training phase inform further optimization efforts to improve system performance. It practices the Google WebSpeech API and compares it with other Speech-to-text models. In conclusion, our proposed work represents a significant step towards realizing seamless voice-controlled interactions with smart assistants and enhancing user experiences and productivity. Future work includes refining the system architecture, optimizing model performance, and expanding the capabilities of the smart assistant for broader application domains.

Index Terms: Natural Language Processing, Feedforward Neural Network, Smart Assistant, Intent Recognition, Audio-To-Text Converter, Virtual Voice Assistant, Word Error Rate, Character Error Rate

1. Introduction

The surge of smart assistants has transformed human interaction with technology, offering seamless voice-controlled access to tasks like alarm notifications, streaming music, and retrieving information. However, despite their widespread adoption, current smart assistant systems face limitations in accurately interpreting and responding to natural language input. Traditional audio-to-text conversion methods and NLP often struggle to grasp the nuances of human speech and contextual input [11], resulting in suboptimal user experiences. Particular focus should be made on the lack of accessibility catering to the needs of visually impaired individuals, ensuring inclusivity and usability for all users [12]. When we choose the audio-to-text converter, we consider such factors: how fast the model processes the word accurately, multilingual support, cost of service, and how the model performs for certain customer-service environments and sentiment analysis.

To address these challenges, this proposed work endeavors to develop a sophisticated audio-to-text converter integrated with natural language capabilities tailored to enhance smart assistant interactions. The proposed work is motivated by the existing gaps in research, which reveal a need for more advanced systems capable of understanding natural language input with higher accuracy and context sensitivity. This proposed work also acknowledges the definite hurdles encountered in healthcare settings, where the accurate interpretation and response to natural language input are paramount [13]. By leveraging advanced techniques in speech recognition and NLP, the proposed system aims to enable more intuitive and personalized interactions between users and smart assistants, ultimately enhancing user satisfaction and productivity.

The main goal here is to propose and execute a robust system architecture that precisely transcribes spoken language and interprets transcribed text using techniques like tokenization, part-of-speech (POS) tagging, and named entity recognition (NER). Additionally, the system will incorporate intent recognition capabilities to discern user intentions from transcribed text, facilitating contextually relevant interactions with the smart assistant. Through a comprehensive literature study, the proposed work seeks to identify gaps in existing research and contribute to the advancement of smart assistant technology by addressing these gaps. Overall, this proposed work aims to bridge the existing gaps in smart assistant technology by developing an advanced system capable of understanding and responding to natural language input with higher accuracy and context sensitivity. By addressing these scientific and technical challenges, the research holds significant potential to enhance user experiences and productivity in the realm of human-technology interaction.

The motivation behind embarking on this proposed work is rooted in the recognition of the existing limitations within contemporary smart assistant systems. While these systems have become ubiquitous in daily life, their efficacy in accurately comprehending and responding to natural language input remains somewhat constrained. Users often encounter challenges with voice commands and queries, resulting in suboptimal user experiences and a sense of frustration. As such, there is a pressing need to advance the capabilities of smart assistants by integrating sophisticated audio-to-text conversion technologies with NLP techniques. By addressing these limitations, this proposed work seeks to usher in a new era of smart assistant interactions characterized by seamless, intuitive, and personalized experiences. Through this endeavor, the aim is to enhance user satisfaction and also boost overall productivity in various domains.

2. Related Works

The research [1] developed and optimized the speech-to-text conversion system with the hidden markov model (HMM) and N-gram models and practiced vectors to reduce word error rates. The NLP enhances the facilities of virtual assistant systems [2] with the SentencePiece tokenizer and RoBERTa methods and analyzes XNLI Accuracy, exact match error rate, semantic error rate, and perplexity parameter. The 17M-parameter model illustrates an enhancement of 4.83% over XLM-R. AI voice assistants [3] based on GPT-3 technology using Large Language Models (LLM) can improve content aggregation and context provision in the context of response generation.

Employing the Whispers model to develop cohesive interfaces for audio-to-text conversion [4]. This focuses on NLP, loading time, stress test, and transcription accuracy and speed. It can support up to 180 simultaneous users with an average response time of 309 ms, managing 471.5 requests per second. The design and implementation of a smart voice assistant, Alice [5], uses neural networks to recognize commonly used words. Alice can detect hot words, recognize intent, and convert voice-to-text and text-to-voice. The researcher analyzed the identification rate and training accuracy. 13 MFCC coefficients with a two-layer feed-forward and 50 neurons in the hidden layer generated a recognition accuracy of 85%.

This review [6] evaluates the current usability metrics applied to voice assistants and independent variables used in their context in the ISO 9241-11 framework. They analyzed voice assistant personalities, query expressions, experience, task characteristics, context, communication type, response type, and conversational type. It examines whether the ISO 9241-11 framework serves as a standardized metric for assessing voice assistants. The rule-based smart assistant application using the RBMT framework [7] develops the intelligent bot application based on NLP and generates a report on response generation, context, and face recognition. Intelligent conversational assistance [8] using an NLP-based HCI architecture synthesizes the speech to recognize the context and generate the response. The CNN and

Generative Adversarial Network (GAN) [9] were utilized for audio style conversion, and the system's Mel-frequency, model accuracy, and loss were analyzed. The classifier networks achieve an accuracy of 99% and 94%, respectively.

Some researchers [10] focused on specific native languages like Marathi and employed deep learning techniques for POS tagging in NLP, specifically employing Bi-LSTM models. They analyzed the mean average precision (mAp), working area size, detection time, and BFLOPS. The deep learning models and Bi-LSTM models exhibited notable performance in POS tagging for Marathi text, with accuracy rates of 85% for deep learning models and 97% for Bi-LSTM models. Here we list some research questions from this study.

2.1 Research Questions:

Q1. What are the key challenges in developing audio-to-text converters for smart assistants?

Developing audio-to-text converters for smart assistants poses several key challenges [14]. These include achieving high accuracy in speech recognition amidst variations in accents, speech patterns, and background noise. Additionally, natural language understanding techniques are essential for interpreting user intents and context from the transcribed text. Context sensitivity is crucial for delivering personalized responses, while handling ambiguity and variability in natural language further complicates the task. Ensuring the privacy and security of user data, real-time processing for low-latency interactions, and scalability to diverse user bases are also significant challenges.

Q2. How can NLP enhance the performance of audio-to-text converters?

NLP plays a vital role in enhancing the performance of audio-to-text converters in several ways. Firstly, NLP techniques enable the conversion of transcribed text into structured data by tokenization, lemmatization, and part-of-speech tagging, facilitating deeper analysis and understanding of the text. This structured representation enables the system to extract meaningful information, such as user intents and context, from the transcribed audio. The dependency parsing and contextual embedding are the more powerful tools in NLP. The dependency parsing builds the tree structure relationship between the words, such as subject-verb and object-verb relationships. Unlike traditional models, the BERT model translates the word meanings based on context, which improves the performance in sentiment analysis and named entity recognition.. Additionally, NLP algorithms can identify and handle linguistic nuances, including synonyms, idiomatic expressions, and colloquialisms, improving the accuracy and robustness of transcription. Furthermore, NLP facilitates context-aware responses, allowing smart assistants to generate more relevant and personalized interactions based on the transcribed text. Overall, integrating NLP capabilities empowers audio-to-text converters to deliver more accurate, contextually relevant, and intuitive user experiences.

Q3. What are the existing methods or algorithms used in audio-to-text conversion, and how effective are they?

Several methods and algorithms are utilized in audio-to-text conversion, each with its strengths and weaknesses. One common approach is Automatic Speech Recognition (ASR) [16], which employs techniques like HMMs and Deep Neural Networks (DNNs) to transcribe spoken language into text. ASR systems have shown considerable effectiveness, especially in controlled environments, but may struggle with accents, background noise, and complex speech patterns. Another method is the use of recurrent neural networks (RNNs) and their variants like Long Short-Term Memory (LSTM) networks, which excel in capturing temporal dependencies in audio data, thus improving accuracy. However, RNNs may face challenges in processing long sequences and can suffer from vanishing gradients. Additionally, transformer-based models like BERT and GPT have gained popularity for their ability to capture contextual information effectively, leading to impressive results in audio-to-text conversion tasks. While these methods show promise, there is still room for improvement, particularly in handling diverse accents, noisy environments, and conversational speech for more robust and accurate transcription.

Q4. How can the accuracy of audio-to-text converters be improved?

Improving the accuracy of audio-to-text converters can be done by refining model architectures with advanced neural networks like transformer-based models, leveraging extensive and diverse training data, implementing effective preprocessing techniques to handle environmental noise and speech variations, selecting informative features, fine-tuning models on domain-specific data, employing ensemble methods, and applying post-processing techniques. By addressing these factors comprehensively, the accuracy of audio-to-text converters can be significantly enhanced, facilitating more reliable and precise transcription of spoken language for various applications.

Q5. What is the role of Machine Learning in developing audio-to-text converters for smart assistants?

Machine learning plays a pivotal role in developing audio-to-text converters for smart assistants. By leveraging machine learning algorithms and techniques, such as deep learning, neural networks, and statistical modeling, developers can train models to accurately transcribe spoken language into text. These models learn from large volumes of annotated audio data to recognize patterns, phonetics, and linguistic structures inherent in human speech. Through iterative training and optimization, machine learning models can adapt to various accents, languages, and environmental conditions, improving the accuracy and robustness of audio-to-text conversion. Additionally, machine learning facilitates continuous improvement and adaptation of converters based on user feedback and evolving language trends, ensuring optimal performance over time. Most of the strategies used to generate graph structures for data sets employ

statistical methods. A typical way of classifying them is taking into account the randomness adopted in the graph structure generation, that is, whether it is random or non-random. Regarding the random methods, the results obtained by [6] in the study of the small-world phenomenon have been explored in the modeling of graph structures for a variety of real networks. This is the case of [7], who studied the search for data in large information networks, and [8, 9], who analyzed the small-world problem as an intrinsic characteristic of any type of network. The concept of topology was also explored by [10] in the construction of random graphs for collections of textual documents.

The second category of strategies to generate graph structures, i.e., non-random methods, is more useful for both theoretical and practical purposes. An example is the recursive matrix model proposed that can generate realistic graphs efficiently. Other network topology models were explored and applied to generate graph structures for backbone networks. The online probabilistic topological mapping proposed by [15] also represented a relevant advance in the problem of finding the graph structure of an environment from a sequence of measurements.

3. Materials and Methods

The Feedforward Neural Network (FNN) model is an artificial neural network with a straightforward structure. Its node connections never form loops. Also known as a multilayer perceptron (MLP), the FNN consists of an input layer, one or more hidden layers, and an output layer, tabulated in Table 1. Information moves in one direction, from the input nodes through the hidden nodes to the output nodes, without any feedback cycles. When it comes to NLP [16], FNN models are often used in tasks like text classification, sentiment analysis, and NER. NLP modules interact with an FNN to process and transcribe speech efficiently. The raw audio signals are converted into numerical representation using spectrograms. The background noises are filtered out to improve the transcription accuracy. Splits the continuous speech audio into chunks for processing. The NLP pre-processing module refines the transcriptions by tokenization, which splits text into words or subwords. Then it identifies naming identities such as name and place. It analyses the grammatical relationship between the words. The contextual embedding module enhances the meaning of words for clarity. The FNN input layer receives the processed text embeddings and learn to extract features from input text embeddings and make predictions based on those features. The hidden layers employ non-linear transformations to learn the patterns. The FNN model has been trained multiple times with input-output layers, and adjusting the weights between nodes leads to minimizing the prediction error. This process, known as backpropagation, allows the FNN to learn complex patterns and relationships within the data. Finally, the system predicts the most probable intent category based on learned patterns.

FNN models are relatively simple compared to other types of neural networks, making them easy to implement and train. However, they may struggle with capturing long-range dependencies in sequential data, such as in language modeling tasks. As a result, more advanced architectures like recurrent neural networks (RNNs) or transformers are often preferred for NLP tasks that involve sequential data. RNN model is suitable for time-series data, but it struggles to process long-term dependencies. Transformers outperform in complex sequences, but they require a high computational cost. Hence, the FNN is employed for the feature extraction process in the proposed work, it utilizes such features of FNN as a simple, fast effective method for structured data. Mostly smart assistants do the sequential process with limited memory; in that case, FNN is suitable for sequential data processing with a lack of memory. When the smart assistants are utilized in routine work, the system can learn the basic structure of input patterns; hence, the system can train the model with an inexpensive FNN model for handling static data.

Table 1. Model Summary

Model: "sequential"		
Layer (type)	**Output Shape**	**Param #**
dense (Dense)	(None, 128)	10,752
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 11)	715
Total params: 39,448 (154.10 KB) Trainable params: 19,723 (77.04 KB) Non-trainable params: 0 (0.00 B) Optimizer params: 19,725 (77.05 KB)		

Fig. 1 represents a detailed process of a voice recognition and response generation system. Here's a breakdown of its components: 1. User is the actor who interacts with the system. In this case, it's the person using the voice recognition system. 2. Sound waves are the input from the user. The user speaks, and the sound waves are captured by the microphone. 3. The microphone device captures the sound waves from the user and converts them into a digital signal. 4. The digital signal is the output of the microphone. The sound waves are now converted into a digital signal and sent to the Speech Recognition system. 5. The speech recognition system processes the digital signal to extract the query in the form of text. 6. Query (text) is the output of the Speech Recognition system. The digital signal is now converted into text and sent to the Intent Identifier. 7. Intent identifier system receives the text query, identifies the intent and query, and forwards it to the Response Generator. 8. Intent and query are the output of the Intent Identifier. It

represents the identified intent and query of the user's input. 9. Response generator system receives the intent and query, and calls upon External APIs and Python Libraries to generate a response in the form of text. 10. Response (text) is the output of the Response Generator. It's the response generated based on the identified intent and query. 11. Text-to-speech system receives the response in text form, converts it into audible feedback, and sends it to the speaker or display. 12. Speaker/Display is the final output of the system. It's the response delivered to the user either audibly through a speaker or visually as text on a display.

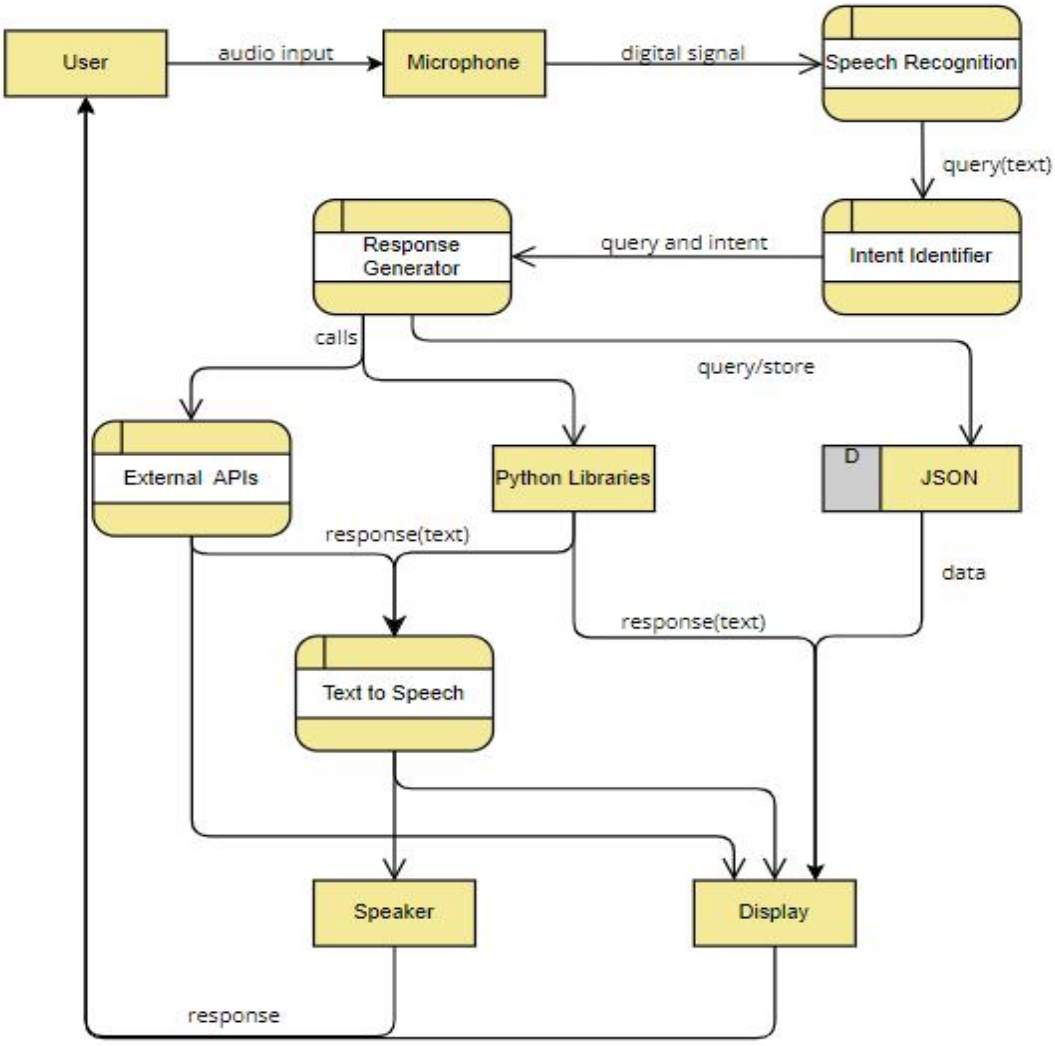

Fig. 1. System Overview

The proposed architecture (see Fig.2) for the audio-to-text converter with NLP capabilities for smart assistants is designed to enable seamless and intuitive interactions between users and the assistant. It consists of several interconnected modules, each serving a specific function in the overall system. At its core, the architecture facilitates the conversion of spoken language into text, understanding of the transcribed text, recognition of user intents, generation of relevant responses, and integration with external services and APIs.

The architecture begins with the User Input Module, responsible for capturing and preprocessing audio signals to ensure clarity and consistency in subsequent processing stages. The training module analyses the transcribed text to understand its meaning, structure, and context based on the intent dataset stored in a JSON file. Various Python Libraries and External APIs are used to carry out simple tasks. The User Interface serves as the interface between users and the system, facilitating seamless interaction through voice-based commands and displaying the output according to the user's input. Overall, the architecture provides a comprehensive framework for developing an intelligent smart assistant system capable of understanding and responding to user queries effectively, enhancing user experiences and productivity.

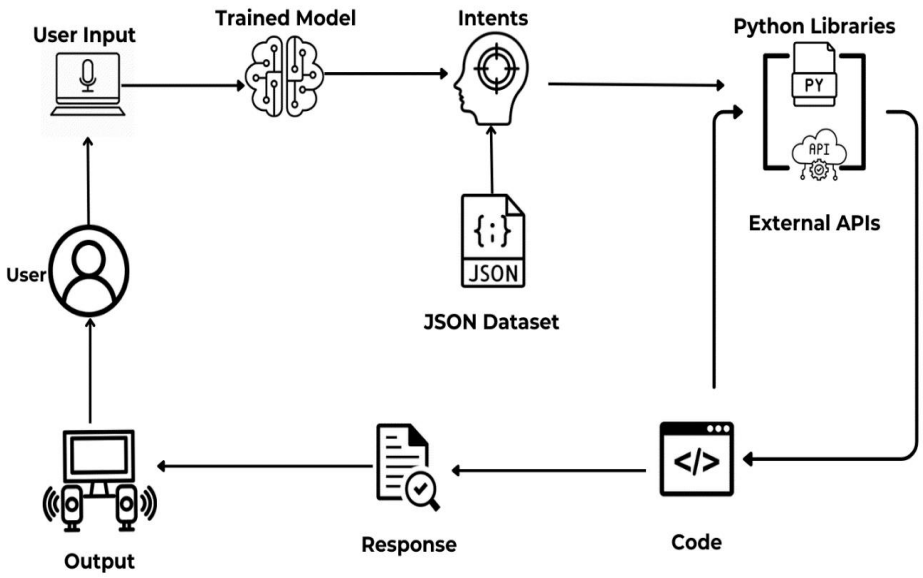

Fig. 2. System Architecture

The proposed architecture (refer to Fig.2) for the smart assistant system encompasses several interconnected modules designed to seamlessly handle audio input, speech recognition, NLP, intent recognition, response generation, user interaction, and integration with external APIs and services. Each module plays a critical role in facilitating effective communication between the user and the smart assistant, enabling intuitive and contextually relevant interactions. The architecture diagram shows how these components interact with each other. The user interacts with the chatbot through the GUI. The user's input is processed by the trained model to identify the intent. Depending on the intent, the chatbot may fetch data from an external API (like the weather API), use Python libraries to generate a response, or directly generate a response based on the identified intent. The response is then displayed to the user through the GUI. This architecture allows the chatbot to handle a variety of user inputs and provide relevant responses.

The user input module is the starting point of the process, where a user provides input via a computer interface. It's responsible for capturing and processing the raw data from the user to be fed into the trained model. The trained model module is responsible for training the chatbot. It uses NLTK for NLP, including tokenization and lemmatization, and TensorFlow for building and training a Feedforward Neural Network (FNN) model. The Training class reads intent data from a JSON file, processes it to create training data, and then trains the model using this data. The trained model is saved to be used later for making predictions. The Testing class also includes methods for cleaning up sentences and converting them into word vectors suitable for classification by the trained model. After the trained model processes the user input, the identified intents are extracted in the intents module. These intents guide how the system will respond to specific user inputs, ensuring that the responses are relevant and appropriate. The intents then interact with a JSON dataset in the JSON dataset module. This dataset contains pre-defined responses or actions associated with various intents. It ensures that each recognized intent is matched with an appropriate response or action. Based on the matched intents and the corresponding data in the JSON dataset, a response is generated in the response module. This module ensures that the outputs are well-formulated and relevant to the identified intents.

The code module includes the coding part that can be involved in processing or responding to certain types of inputs or intents. This allows for flexibility and customization in the responses. Python Libraries and External APIs modules indicate that the system integrates with Python libraries and external APIs. This enhances the system's functionality, broadens the scope of responses, or performs specific tasks requested by users that aren't covered by pre-defined responses in the JSON dataset. The output module is the final stage where all the processed information converges to produce an output. The output is then delivered through a computer interface back to the user. This module ensures that users receive timely, relevant, and accurate responses based on their initial inputs.

The provided use case diagram (see Fig.3) represents the interactions between a user and a Virtual Voice Assistant. This diagram helps in understanding the different functionalities that a Virtual Voice Assistant can perform and how a user can interact with it using voice queries. They extend from the primary use case (Voice Query) and provide weather updates, deliver the latest news headlines, tell jokes, inform about the current time, play songs, search content in Wikipedia, and respond to the user's queries.

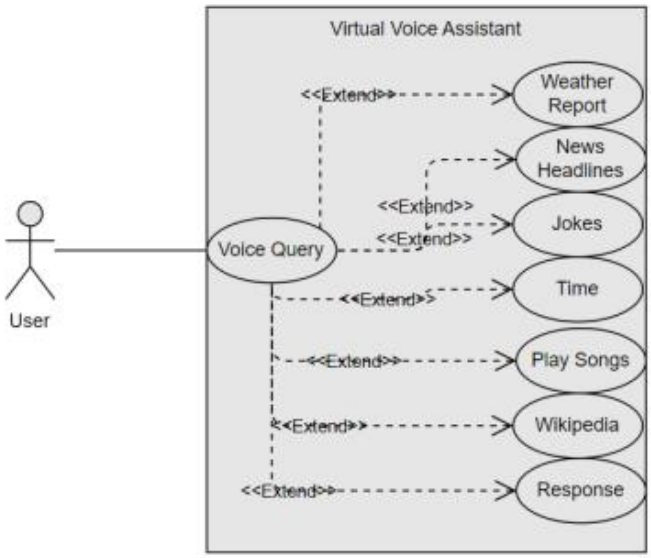

Fig. 3. Voice Assistant Use Cases

3.1 Dataset:

To develop a highly effective and versatile smart assistant system, a comprehensive selection of datasets meticulously curated to encompass a broad spectrum of linguistic nuances, contexts, and scenarios is employed. The data collection strategy prioritizes diversity, quality, and relevance, ensuring that the models are trained on rich and representative datasets that closely mirror real-world interactions. The LibriSpeech dataset is integrated for testing the model. LibriSpeech, a renowned repository of read English speech sourced from audiobooks available in the public domain, features a vast array of genres, reading styles, and linguistic complexities. By leveraging LibriSpeech, the speech recognition models are equipped with the ability to handle a broad spectrum of linguistic content with accuracy and precision.

An intent dataset sourced from Kaggle [17], a renowned platform for machine learning datasets, serves as the backbone of the proposed work. This dataset provides a diverse range of intents, patterns, and responses crucial for training the intent recognition system. Utilizing this dataset after fine-tuning ensures that the model is exposed to a wide array of user queries, enabling it to learn and recognize various intents accurately. Additionally, the richness of the dataset allows for thorough testing and evaluation, ensuring the robustness and generalization capabilities of the model across different intents and scenarios. The data preprocessing stage is crucial in preparing the collected audio and text data for further analysis, training, and evaluation. This process involves several steps aimed at cleaning, formatting, and standardizing the data to ensure its quality and consistency. The dataset had been preprocessed by removing background noise using noise reduction algorithms. Generally, this dataset is derived from various books, hence, a variety of accents are preprocessed by a normalization approach and ensure the file formats.

Text data preprocessing involves various techniques as well. Tokenization, for instance, is the process of splitting text into individual words or subwords for further analysis, often using whitespace or punctuation as delimiters. Tokenization breaks down the text into smaller units, facilitating analysis at the word or subword level. Moreover, lemmatization, which involves the reduction of words to their base or canonical form (lemma), simplifies analysis by treating different inflections of a word as a single entity. Lemmatization ensures consistency in word representation and reduces the vocabulary size. Additionally, stemming, a similar process to lemmatization, reduces words to their root form by removing affixes, but may result in non-dictionary words. Stemming is a more heuristic approach compared to lemmatization and is useful for reducing word variations. Lastly, stop word removal eliminates common words (stop words) that do not carry significant semantic meaning and are unlikely to contribute to analysis or understanding. Stop word removal reduces noise in the text data and improves the efficiency of downstream analysis.

These preprocessing techniques are essential for enhancing the quality of the data and making it suitable for downstream analysis and processing tasks. By applying these techniques, the project aims to create clean, standardized datasets that facilitate accurate and effective training of speech recognition and NLP models, ultimately leading to improved system performance and usability. Through careful preprocessing and analysis of this dataset, the foundation is laid for developing a versatile intent recognition system tailored to the needs of the smart assistant application. The judicious selection and integration of these diverse datasets provide the smart assistant system with access to a wealth of linguistic diversity and richness. Exposing the models to a broad spectrum of linguistic contexts, accents, and domains equips the system with the adaptability, accuracy, and robustness required delivering intuitive, contextually relevant interactions tailored to users' diverse needs and preferences. The users must be aware that their voice data is being collected and how and which third-party applications are used. Transparency rules are defined for the storage duration and data-sharing practices, and let the user decide on opt-out options.

3.2 Performance Metrics:

The data evaluation for audio-to-text conversion is done on two metrics for voice recognition – Word Error Rate (WER) and Character Error Rate (CER). The WER metric computes substitutions, insertions, and deletions on the word level. This implies that each word of the error is annotated separately. Next, the WER can be calculated as follows:

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{1}$$

Here, S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, and N is the number of words in the reference (N=S+D+C).

Systems are evaluated at the character level by the CER metric. This implies that we break apart our words into their characters and annotate errors character by character. Next, the WER can be calculated as follows:

$$CER = \frac{S+I+D}{N} = \frac{S+I+D}{S+D+C} \tag{2}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct characters, and N is the number of characters in the reference ($N = S + D + C$).

On the other hand, in the training module, two key metrics are produced: training loss and accuracy. Training loss denotes how closely the model's predictions match the actual outcomes. It's calculated using a loss function; in this case, categorical cross-entropy is appropriate for multi-class classification tasks. The goal during training is to minimize this loss. The formula for categorical cross-entropy loss of a single sample is:

$$L = -\sum_{i=1}^{c} y_{i.\log(p_i)} \tag{3}$$

Here, c is the number of classes, y_i is the actual value (0 or 1) of class i for the current sample, and pi is the predicted probability of class i for the current sample.

Accuracy is the proportion of correct predictions made by the model out of all predictions. It's a direct measure of the model's performance. The model predictions are better when they produce higher accuracy.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4}$$

During each training epoch, the model's weights are updated to minimize the loss, and the accuracy is calculated. This process is repeated for the number of epochs specified (200). The hist variable will contain the history of the loss and accuracy metrics for each epoch.

The model is then saved for future use. Also, the training data, words, and classes are saved using the pickle module. This allows for the model to be loaded later and used for predictions. For data verification, the trained model is used to predict outcomes for a separate dataset, known as the test set, that the model hasn't seen during training. This provides an unbiased evaluation of how well the model generalizes to new, unseen data. The same metrics (loss and accuracy) can be calculated for the test set to evaluate performance.

4. Results and Discussions

The results obtained from evaluating the speech recognition system offer a comprehensive understanding of its performance across various audio conditions and during the training phase. One notable insight gleaned from the analysis is the significant impact of different audio environments on recognition accuracy. This observation underscores the need for robust algorithms capable of effectively handling diverse audio conditions, which is crucial for real-world applications where environmental noise and speaker accents are prevalent. Hence, the instance of the Recognizer class from speech recognition library listens to the ambient noise from the source for 0.2 seconds. It adjusts the energy threshold to filter out background noise. Moreover, the analysis reveals intriguing patterns regarding the pace of speech and its influence on recognition accuracy. Fast speech tends to result in higher error rates compared to slower speech. This finding highlights the importance of developing algorithms that can adapt to variations in speaking speed, as users may naturally speak at different rates in different contexts. Speaker adaptation and personalization are available, allowing users to customize speech models based on their accent, learning from individual pronunciation habits to refine transcription accuracy over time. Additionally, the evaluation of the Librispeech dataset emphasizes the critical role of dataset quality in determining recognition performance. Clean data leads to significantly lower error rates than other datasets, underscoring the importance of curating high-quality training data to improve model accuracy.

Furthermore, the results from the training phase provide valuable insights into the progression of the model over epochs. The observed trend of decreasing loss and increasing accuracy indicates that the model is learning effectively from the training data and improving its performance over time. This gradual improvement in accuracy suggests that the model is effectively taking the underlying patterns in the data and adjusting its parameters accordingly. Additionally,

the achievement of a 100% accuracy rate in the 198th epoch highlights the potential of the model to achieve high levels of performance with further training.

The evaluation of the audio input in various conditions reveals the following insights, as shown in Fig.4. In a noisy environment, the average WER is 12.7% and the CER is 6.3%. However, in a quiet environment, both WER and CER significantly improve to 5.2% and 2.8%, respectively. Accented speech seems to pose a challenge with a WER of 15.3% and a CER of 7.9%. The pace of speech also impacts the results: fast speech results in a WER of 10.1% and a CER of 5.4%, while slow speech improves the rates to 6.8% and 3.5%, respectively. When tested on the Librispeech dataset, the clean data yielded a WER of 4.7% and a CER of 2.5%, while the other data resulted in a WER of 8.9% and a CER of 4.7%.

The model's training process over 200 epochs shows a general trend of decreasing loss and increasing accuracy, indicating that the model is learning from the data, as shown in Fig.5. The loss started at 0.9994 in the first epoch and decreased to 0.0145 by the 200th epoch. The accuracy, however, began at 0.0714 and increased to 0.9762 by the end of the training. The highest accuracy achieved was 1.0000 in the 198th epoch. These metrics are crucial for understanding the performance of the model during the training phase and can help in tuning the model for better performance.

WER and CER are the primitive metrics of speech recognition systems [18]. These addressed the transcription errors at the word and character levels, directly impacting the quality of speech recognition accuracy. These provide a quantifiable measure of transcription errors, making them essential for evaluating objective and subjective dependencies in audio-to-text models. BLEU (Bilingual Evaluation Understudy) score [19], precision, and recall metrics are more relevant to the classification in intent recognition models and text generation in machine translation models, but these metrics do not directly evaluate the transcription accuracy. BLEU is a comparative metric that evaluates the fluency and relevance, so it is less effective for raw speech transcription. These metrics are suitable for understanding semantic and text-based NLP tasks. Hence, the proposed model prioritizes the WER and CER to focus on error correction in transcription. The experiment results show that lower WER and CER are associated with higher user satisfaction in the audio-to-text converter.

Thus, the insights gained from analyzing the speech recognition system's performance offer valuable guidance for future optimizations and improvements. Addressing the challenges posed by different audio conditions, such as noise and accents, will be crucial for enhancing the system's robustness and real-world applicability. Moreover, continued efforts to refine the algorithm and optimize training strategies based on the observed patterns will be essential for achieving even higher levels of accuracy and performance. Overall, these insights contribute to advancing the field of speech recognition and pave the way for more effective and reliable speech-based interaction systems.

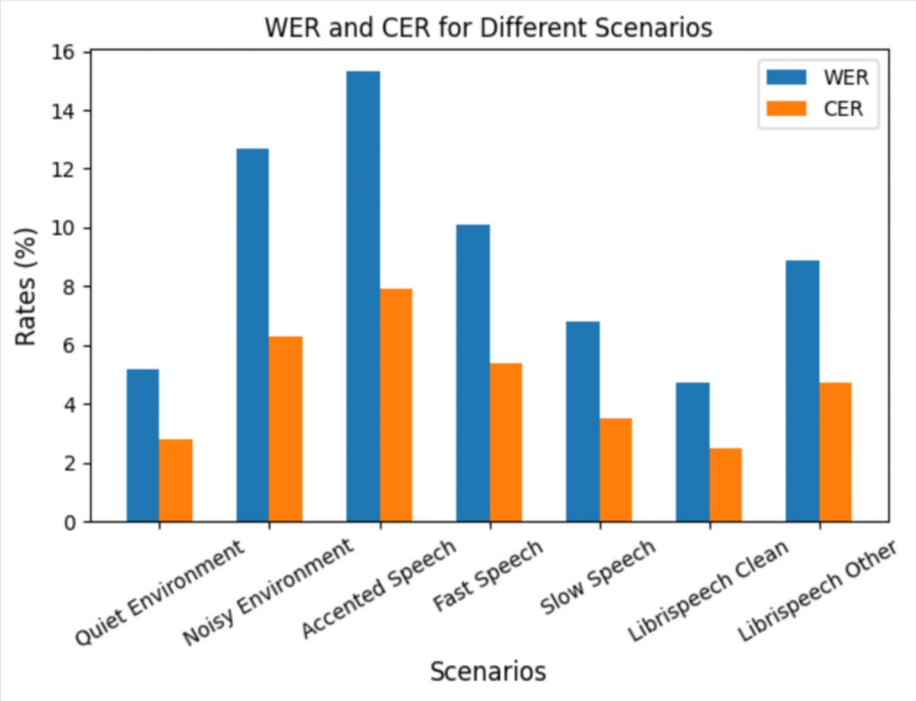

Fig. 4. WER and CER evaluation graph

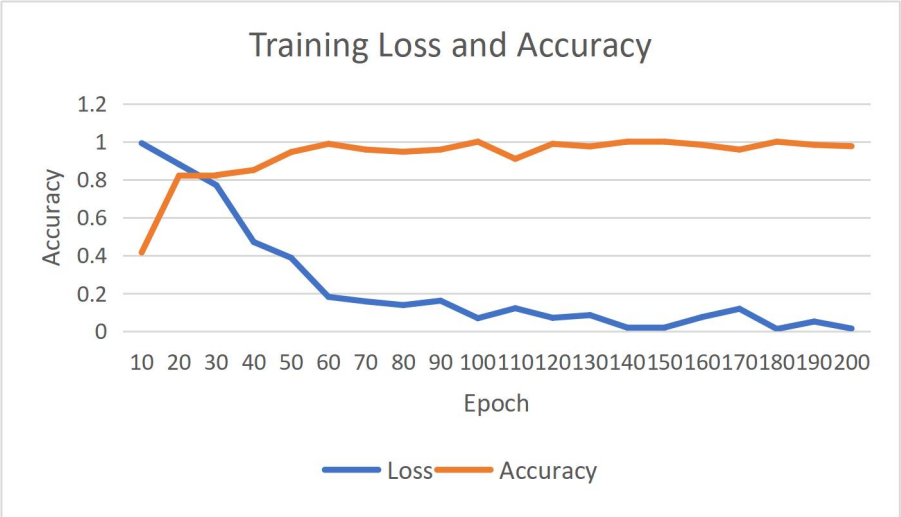

Fig. 5. Training Loss and Accuracy

After conducting a thorough evaluation of existing Speech-To-Text (STT) models, Google WebSpeech [20, 21], DeepSpeech[22], and transformer-based Wav2Vec2[23], it is evident that Google WebSpeech consistently outperforms its predecessors across various scenarios of audio inputs. Google WebSpeech is a cloud-based service, so it requires an internet connection. It performs well with low latency and struggles with noisy and non-standard accents, but has lower WER/CER than other models. DeepSpeech is open source and fine-tunable, and can run offline. It has a higher inference time and a higher WER than Wav2Vec2. Wav2Vec2 is a self-supervised learning model, optimized for batch processing, but real-time performance depends on hardware. Table 2 shows the WER and CER for all the models in different scenarios. Therefore, based on these findings, Google WebSpeech emerged as the preferred choice for speech recognition and speech-to-text conversion. The cloud-based Google WebSpeech offers scalability, handles increasing user demands without performance degradation. It breaks the speech into smaller time frames for faster processing. It has GPU acceleration and TPU that speed up deep learning models in the cloud and transcribe audio efficiently. This streaming STT model transcribes speech continuously, making it ideal for real-time live applications [24].

WER and CER quantify the transcription errors by comparing the predicted text and ground truth text, while CER measures performance in real-world audio environments. K-fold validation [25] assesses generalization but doesn't provide a fine-grained error breakdown. WER/CER is more effective for testing against unpredictable conditions, while K-fold ensures stable training. K-fold validation is less resource-intensive and can be quickly calculated for multiple scenarios. The accuracy of the audio-to-text converter is decisive, and WER/CER is more actionable. K-fold validation is useful in small datasets or domains where overfitting is an issue, but WER/CER is more relevant for large-scale STT models. Hence, the proposed model practices WER and CER metrics to measure the transcription accuracy and assess the robustness of the model by ensuring good performance under various conditions and input variations.

Table 2. WER and CER comparative analysis

Environment	Google WebSpeech		DeepSpeech		Wav2Vec2	
	WER	CER	WER	CER	WER	CER
Quiet	5.2	2.8	9.6	4.7	3.8	2.1
Noisy	12.7	6.3	12.8	7.1	14.9	7.6
Accented	15.3	7.9	22.4	10.1	18.7	8.4
Fast	10.1	5.4	14.6	8.3	11.3	6.4
Slow	6.8	3.5	9.8	4.8	4.3	3.2

5. Conclusion

In conclusion, this proposed work has made significant strides in developing a sophisticated smart assistant system equipped with advanced audio-to-text conversion and NLP capabilities. Through meticulous research, design, and implementation, we have tackled several key challenges successfully and achieved notable successes in enhancing user interactions with the smart assistant. One of the major accomplishments of this proposed work is the successful integration of audio signal processing, speech recognition, and NLP. By leveraging an FNN model, we can achieve high accuracy and robustness in transcribing audio input and understanding natural language queries. Additionally, the system's modular architecture has facilitated seamless interaction between different components, enabling contextually relevant responses and improving overall user satisfaction. Additionally, while the system performed well under controlled conditions, its performance may vary in real-world scenarios with diverse accents, background noise, and environmental factors, suggesting more extensive testing and optimization.

Looking ahead, there are several avenues for future work and improvement. Firstly, additional research and development efforts could focus on the system's robustness and adaptability to diverse user inputs and environments. This could involve collecting diverse training data, fine-tuning models on specific user demographics, and implementing techniques for handling out-of-vocabulary words and domain-specific language. Furthermore, exploring alternative approaches to intent recognition and response generation, such as reinforcement learning and generative adversarial networks, could lead to more personalized and contextually relevant interactions. Moreover, there is potential for extending the functionality of the smart assistant system by integrating with external services and APIs, such as e-commerce platforms, social media networks, and IoT devices. This would enable users to perform a wider range of tasks and access information from multiple sources seamlessly. Additionally, improving the user interface and interaction design to accommodate accessibility features and support for different languages could enhance the inclusivity and usability of the system for diverse user groups. In summary, while this proposed work has achieved significant milestones in developing a sophisticated smart assistant system, there is still ample room for improvement and innovation. By addressing the identified challenges and pursuing future research directions, we can continue to advance the state-of-the-art smart assistant technology and deliver more seamless, intuitive, and personalized user experiences.

References

[1] Korchynskyi, V., & Vynogradov, I. (2024). Methods of improving the quality of speech-to-text conversion. Scientific Collection, InterConf, (194), 426-437.

[2] Antonius, F., Alapati, P. R., Ritonga, M., Patra, I., El-Ebiary, Y. A. B., Orosoo, M., & Rengarajan, M. (2023). Incorporating Natural Language Processing into Virtual Assistants: An Intelligent Assessment Strategy for Enhancing Language Comprehension. International Journal of Advanced Computer Science and Applications, 14(10).

[3] Chennuri, Varun & Rodda, Vamshi Prashanth. (2023). Development of AI-based voice assistants using Large Language Models. 10.13140/RG.2.2.20195.12321.

[4] Haz, A. L., Fajrianti, E. D., Funabiki, N., & Sukaridhoto, S. (2023). A Study of Audio-to-Text Conversion Software Using Whispers Model. In 2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE) (pp. 268-273). IEEE.

[5] Abougarair, Ahmed. (2022). Design and implementation of smart voice assistant and recognizing academic words. International Journal of Robotics and Automation. 8. 27-32.

[6] Dutsinma, F.L., Pal, D., Funilkul, S., & Chan, J.H. (2022). A Systematic Review of Voice Assistant Usability: An ISO 9241–11 Approach. Sn Computer Science, 3.

[7] K, Vindhya & K, Tejasvi & K, Pravallika & Krishna, Ch. (2022). A Natural Language Processing based Intelligent Bot Application. 391-396.

[8] Sharma, D., Paliwal, M., & Rai, J. (2021). NLP for Intelligent Conversational Assistance. International Journal of Innovative Research in Computer Science & Technology, 9(3), 179-184.

[9] Ezhilan, A., Dheekksha, R., & Shridevi, S. (2021). Audio style conversion using deep learning. International Journal of Applied Science and Engineering, 18(5), 1-8.

[10] Deshmukh, R. D., & Kiwelekar, A. (2020). Deep learning techniques for part-of-speech tagging by natural language processing. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 76-81). IEEE.

[11] Vijayan, Ramaraj &. V, Mareeswari & Prince, Ephzibah & Kumar, Chandhan. (2024). Improving the BERT model for long text sequences in the question answering domain. International Journal of Advances in Applied Sciences. 13. 106. 10.11591/ijaas.v13.i1.pp106-115.

[12] V, M., Ramaraj, V., G, U., R, S. & E., P. (2019). Accessibility using human face, object, and text recognition for visually impaired people. International Journal of Innovative Technology and Exploring Engineering, 8(6):853–858.

[13] V, Mareeswari & Periyasamy, Jayalakshmi & Thangavel, Muthamilselvan & Bhatia, Surbhi & Almusharraf, Ahlam & Santhanam, Prasanna & Vijayan, Ramaraj & Elsisi, Mahmoud. (2023). Design and Development of IoT and Deep Ensemble Learning Based Model for Disease Monitoring and Prediction. Diagnostics. 13. 1942. 10.3390/diagnostics13111942. 1998.

[14] Sethiya, N., & Maurya, C. K. (2024). End-to-end speech-to-text translation: A survey. *Computer Speech & Language*, 101751.

[15] A. Ranganathan and F. Dellaert, "Online probabilistic topological mapping," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 755–771, 2011.

[16] Kim, K. M., Chen, X., & Liu, X. (2024). Accuracy scoring of elicited imitation: A tutorial of automating speech data with commercial NLP support. *Research Methods in Applied Linguistics*, *3*(3), 100127.

[17] https://www.kaggle.com/datasets/yasiashpot/librispeech/data

[18] Majhi, M. K., & Saha, S. K. (2024). An automatic speech recognition system in Odia language using attention mechanism and data augmentation. *International Journal of Speech Technology*, *27*(3), 717-728.

[19] Labied, M., Belangour, A., & Banane, M. (2024, December). Assessing Speech-to-Text Translation Quality: An Overview of Key Metrics. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 1-6). IEEE.

[20] Ashwell, T., & Elam, J. R. (2017). How Accurately Can the Google Web Speech API Recognize and Transcribe Japanese L2 English Learners' Oral Production?. *Jalt Call Journal*, *13*(1), 59-76.

[21] Prasad, P. K. V., Krishna, N. V., & Jacob, T. P. (2022, April). AI chatbot using web speech API and node. Js. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 360-362). IEEE..

[22] Al-Anzi, F. S., & Shalini, S. B. (2024, May). Continuous Arabic Speech Recognition Model with N-gram Generation Using Deep Speech. In *2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-9). IEEE.

[23] Jain, R., Barcovschi, A., Yiwere, M. Y., Bigioi, D., Corcoran, P., & Cucu, H. (2023). A wav2vec2-based experimental study on self-supervised learning methods to improve child speech recognition. *IEEE Access*, *11*, 46938-46948.
[24] Pople, V., Kanaujia, A., Vijayan, R., & Mareeswari, V. (2022). Face Mask Detection for Real-Time Video Streams. *ECS Transactions*, *107*(1), 8275.
[25] Chen, S. H. K., Saeli, C., & Hu, G. (2024). A proof-of-concept study for automatic speech recognition to transcribe AAC speakers' speech from high-technology AAC systems. *Assistive Technology*, *36*(4), 319-326.

Authors' Profile

Pratistha Tulsyan was a Bachelor of Computer Science student in SCORE, VIT, Vellore, India. She was good at academics and developed many software projects. She is currently pursuing her Master of Computer Applications (MCA), her research interests focus on Natural Language Processing and Cyber Security.

Mareeswari V. is working as an Associate Professor at the School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, Tamilnadu, India - 632014. She has over 21 years of academic and research experience. She received her Ph.D. in Information Technology and Engineering in the Web Service domain. She has produced several national and international articles in reputed journals, conferences, and book chapters. Her area of interest includes Medical Image processing using Machine and Deep Learning, Data analytics on Blockchain, Prediction systems on Web Service Ranking, Internet of Things (IoT), Routing on Mobile Ad-hoc networks, and E-Commerce systems. She is a life member of the Computer Society of India (CSI) and the Soft Computing and Research Society (SCRS).

Vijayan Ramaraj is working as a Professor at the School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology (VIT), Vellore, Tamil Nadu, India. He is a life member of the Computer Society of India (CSI) and the Soft Computing and Research Society (SCRS). He has produced several national and international research articles in reputed journals and conferences. His research interest involves web technologies, wireless networks, ad hoc networks, computer networks, cloud computing, and data analytics.

Suji R. was a Bachelor of Computer Science student in SCORE, VIT, Vellore, India. She was good at academics and developed many software projects. She is currently pursuing her Master of Computer Applications (MCA), her research interests focus on Natural Language Processing and Image Processing.

How to cite this paper: Pratistha Tulsyan, Mareeswari V., Vijayan Ramaraj, Suji R., "Developing Audio-to-Text Converters with Natural Language Processing for Smart Assistants", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.4, pp. 70-81, 2025. DOI:10.5815/ijmecs.2025.04.05