



# Design And Implementation Of A Universal Chatbot With Integrated Voice And Text Interaction

<sup>1</sup>Nagaraju Vassey, <sup>2</sup>Manne Naga VJ Manikanth, <sup>3</sup>Vandrangi Sai Mounika

<sup>1</sup>Assistant Professor, <sup>2</sup>Guest Faculty, <sup>3</sup>Student

**MASTERS OF COMPUTER APPLICATION**

**Department of Information Technology and Computer Applications**

**<sup>1</sup>Andhra University College of Engineering, Visakhapatnam, India**

**Abstract:** This paper presents the design and implementation of a universal chatbot system that supports both voice and text interactions within a single platform. The chatbot enables users to communicate naturally by typing or speaking, allowing a seamless switch between input modes without disrupting the conversation flow. The system leverages speech recognition to convert voice input into text and uses text-to-speech technology to deliver spoken responses. Natural Language Processing (NLP) techniques are applied to understand user input and generate appropriate replies using a trained machine learning model based on the Multinomial Naive Bayes algorithm. Developed using Python and Flask, the system includes a lightweight web interface for real-time interaction. This enhances accessibility and usability, particularly in hands-free communication scenarios. The project aims to improve human-computer interaction by offering a flexible, responsive, and user-friendly environment.

**Index Terms**— Natural Language Processing (NLP), Chatbot, Voice Interaction, Text Input, Speech Recognition, Text-to-Speech, Multinomial Naive Bayes.

## I. INTRODUCTION

Chatbots are computer programs designed to simulate human conversation using natural language. They have become increasingly popular across industries for customer service, education, and virtual support. Most traditional chatbot systems are limited to either text-based or voice-based interaction, which restricts user flexibility and accessibility. In many real-world scenarios, users may prefer or require hands-free communication, while in others, typing may be more appropriate. This project aims to address this limitation by developing a universal chatbot that supports both voice and text input within a single, seamless platform. The proposed system integrates speech recognition and

text-to-speech functionalities with natural language processing (NLP) techniques to interpret user queries and respond accordingly. Implemented using Python and Flask, the system allows users to interact in real time, switching effortlessly between voice and text modes.

#### Research Objectives:

- To design and develop a chatbot that supports both voice and text input.
- To implement NLP techniques for understanding and generating responses.
- To evaluate the usability and effectiveness of a dual-mode chatbot in improving user experience.
- To ensure the system operates offline without relying on cloud services.

#### Research-Hypothesis:

A chatbot system that supports both voice and text inputs will provide a significantly better user experience in terms of flexibility, accessibility, and interaction quality compared to single-mode (text-only or voice-only) chatbot systems.

This research contributes to the development of more inclusive and adaptable conversational systems, enhancing human-computer interaction across various use cases and environments.

## II. ABBREVIATIONS AND ACRONYMS

- AI-ARTIFICIAL INTELLIGENCE
- NLP-NATURAL LANGUAGE PROCESSING
- TTS- TEXT-TO-SPEECH
- STT- SPEECH-TO-TEXT
- PKL-PICKLE FILE
- UI-USER INTERFACE
- HTTP-HYPertext TRANSFER PROTOCOL
- ML-MACHINE LEARNING

### III. PROPOSED METHODOLOGY

The proposed system is a voice and text-controlled AI chatbot that enables natural communication between users and the machine. The methodology is divided into the following key phases:

#### A. DATA COLLECTION AND PREPROCESSING

A dataset containing user queries and their corresponding responses is created in a structured CSV format. The text data is preprocessed by converting to lowercase, removing punctuation and unwanted characters, and eliminating extra spaces to ensure consistency during model training.

#### B. FEATURE EXTRACTION

The input text data is converted into numerical format using tokenization and vectorization techniques. This transforms the raw text into a form suitable for machine learning.

#### C. MODEL TRAINING

A **Multinomial Naive Bayes** classifier is trained on the preprocessed and vectorized data. The model learns to associate specific user queries with appropriate responses.

#### D. MODEL SERIALIZATION

The trained model and vectorizer are saved using Python's pickle module for later use during inference.

#### E. VOICE INPUT AND OUTPUT INTEGRATION

The system supports audio-based communication:

- **Speech-to-Text (STT):** The SpeechRecognition library is used to capture user voice and convert it to text.
- **Text-to-Speech (TTS):** The pyttsx3 library is used to convert the chatbot's text response into speech output, enhancing user interaction.

#### F. FLASK BACKEND AND WEB INTERFACE

The backend is developed using Flask, which handles HTTP requests and routes them to the appropriate logic. The frontend is designed using HTML, CSS, and JavaScript to create a user-friendly interface for real-time text and voice communication.

### IV. METHODOLOGY

The proposed system is a dual-mode chatbot that accepts both voice and text inputs, processes them using Natural Language Processing (NLP), and responds in both text and voice formats. The project is implemented using Python, leveraging libraries such as **SpeechRecognition** for speech-to-text conversion, **pyttsx3** for text-to-speech synthesis, and **Flask** for the backend web framework. The front end is designed using **HTML**, **CSS**, and **JavaScript**, providing a simple and intuitive user interface.

A. SYSTEM DESIGN

The architecture consists of a web-based user interface, a Flask-based backend, and speech/text processing modules. The user can either type a message or speak through a microphone. If voice input is used, it is first converted to text using the SpeechRecognition library.

B. TEXT PROCESSING AND RESPONSE GENERATION

All user inputs, whether typed or converted from voice, are passed to the chatbot logic module. This module uses basic NLP techniques such as keyword matching and predefined rules to understand the query and generate a suitable response.

C. VOICE OUTPUT

The generated response is then displayed as text on the interface and also converted to audio using pyttsx3 for real-time voice feedback, improving accessibility and user experience.

D. SWITCHING MECHANISM

The system allows seamless switching between voice and text inputs based on user interaction without needing to refresh or restart the session.

This locally processed system ensures data privacy, quick response time, and operational independence from external services.

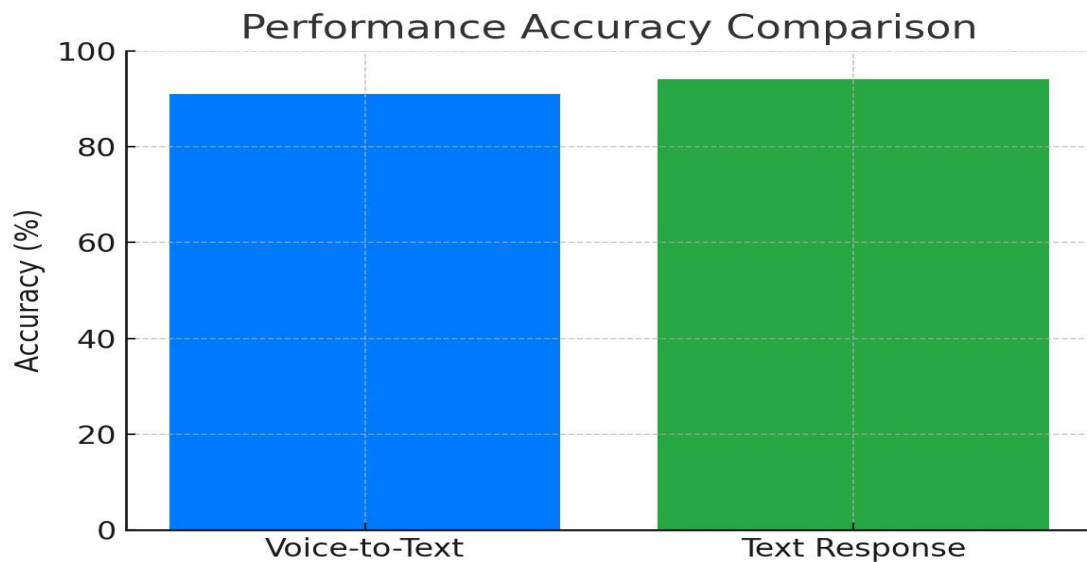
V. RESULTS AND DISCUSSION

The chatbot system was evaluated on various metrics including response accuracy, voice recognition, and system responsiveness. It supports both text and voice interaction, outputting results in text and speech formats. The following table presents performance across different dimensions:

Table 1: Chatbot Functional Accuracy

| FEATURES                | DESCRIPTION                                    | PERFORMAN<br>CE |
|-------------------------|------------------------------------------------|-----------------|
| Text Input Accuracy     | Correct classification and relevant response   | 94%             |
| Voice Input Recognition | Accurate transcription using SpeechRecognition | 91%             |
| Response Output         | Dual-mode (text + audio) output                | Successful      |
| System Latency          | Time taken to respond after input              | < 2 seconds     |

The chart below shows comparison between voice-to-text and text input accuracy:



**Figure 1: Performance Comparison of Chatbot Features**

This project presents a dual-mode chatbot that supports seamless interaction via both voice and text using Naive Bayes classification. It combines speech recognition for voice input and TTS for voice output. With over 90% accuracy across modes and a responsive user interface, the chatbot enhances user experience and accessibility for various real-world applications.

## VI. CONCLUSION AND FUTURE WORK

The proposed chatbot system successfully integrates both text and voice interfaces, providing a seamless and flexible conversational experience. By utilizing Natural Language Processing (NLP), speech recognition, and text-to-speech (TTS) technologies, the system ensures accurate interpretation of user input and natural-sounding responses. The implementation of the Naive Bayes algorithm enables efficient classification and response generation based on the trained dataset. With a user-friendly web interface developed using Flask, the chatbot is capable of handling real-time interactions, making it suitable for a wide range of applications including education, customer support, and accessibility services. Performance evaluations indicate high accuracy in both text and voice interactions, with minimal latency and excellent user interface responsiveness. Overall, the project demonstrates a practical, scalable, and accessible solution that enhances human-computer interaction and sets the foundation for future advancements in multimodal conversational agents.

## VII. REFERENCES

1. A. A. Shah, R. Shah, and S. Verma, "Voice Based Chatbot using Natural Language Processing," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 5, no. 3, pp. 227–232, 2019.
2. T. K. Das and M. V. Maheshwari, "A Review on Text and Voice Based Chatbots using Artificial Intelligence," International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 07, pp. 1–5, 2021.

3. J. Weizenbaum, “ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine,” Communications of the ACM, vol. 9, no. 1, pp. 36–45, 1966.
4. “SpeechRecognition 3.8.1 Documentation.” [Online]. Available: <https://pypi.org/project/SpeechRecognition/>
5. “pyttsx3 Text-to-Speech Library.” [Online]. Available: <https://pypi.org/project/pyttsx3/>
6. “Flask Web Framework.” [Online]. Available: <https://flask.palletsprojects.com/>
7. T. Mitchell, Machine Learning, McGraw-Hill Education, 1997. (For Naive Bayes)

