Voice and Text-Based AI Healthcare Chatbot Using Local Language Models

Guide: Dr. S China Venkateswarlu, Professor, ECE & IARE
Dr. V Siva Nagaraju, Professor, ECE & IARE

Ponnuri Deepanvi[1]

[1]Ponnuri Deepanvi Electronics and Communication Engineering & Institute of Aeronautical Engineering

---***---

Abstract -- In recent years, the integration of Artificial Intelligence (AI) into healthcare systems has transformed the way users access medical knowledge, offering enhanced accessibility, personalization, and efficiency. This project presents the design and development of an AI-based healthcare chatbot capable of functioning both offline and online, supporting voice and text input/output, and built entirely using free, open-source tools. Unlike most existing solutions that depend on expensive or cloud-based APIs, this chatbot utilizes locally hosted machine learning models from Hugging Face (distilGPT2) to deliver responses to health-related queries. The chatbot allows users to either type or speak their questions. It processes spoken input using the SpeechRecognition library and synthesizes spoken responses using pyttsx3, creating a natural, conversational experience. Responses are generated using a hybrid approach: a rule-based system handles common queries such as flu symptoms or sleep advice, while transformer-based text generation provides contextually appropriate answers for general or ambiguous queries. All conversations are handled through an intuitive web interface built with Streamlit, which also maintains session memory to simulate human-like dialogue.

Key Words: Voice-enabled AI chatbot, Healthcare assistant, Natural language processing, Offline chatbot, Speech recognition, Text-to-speech synthesis, Local language models, Streamlit interface, Hugging Face transformers, distilGPT2, pyttsx3, Speech Recognition, PyTorch, Real-time voice interaction, Rule-based healthcare responses.

1.INTRODUCTION

The integration of Artificial Intelligence (AI) in healthcare has significantly reshaped how individuals seek, receive, and interact with medical information. With the rapid proliferation of smart assistants, chatbots, and health monitoring systems, AI-driven solutions are increasingly becoming the first point of contact for individuals in need of basic healthcare advice. Chatbots, in particular, are proving instrumental in disseminating general health guidance, triaging symptoms, and offering educational resources to a wide and diverse user base. However, the majority of commercially available AI healthcare assistants depend heavily on cloud-based platforms and paid APIs such as OpenAI's GPT or Google's DialogFlow, limiting accessibility in areas with poor internet connectivity or financial constraints. The integration of Artificial Intelligence (AI) in healthcare has significantly reshaped how individuals seek, receive, and interact with medical information. With the rapid proliferation of smart assistants, chatbots, and health monitoring systems, AI-driven solutions are increasingly becoming the first point of contact for individuals in need of basic healthcare advice. Chatbots, in particular, are proving instrumental in disseminating general health guidance, triaging symptoms, and offering educational resources to a wide and diverse user base. However, the majority of commercially available AI healthcare assistants depend heavily on cloud-based platforms and paid APIs such as OpenAI's GPT or Google's DialogFlow, limiting accessibility in areas with poor internet connectivity or financial constraints.

2. Body of Paper

2.1 Overview of Speech Emotion Recognition Using Machine Learning

The emergence of artificial intelligence in healthcare has enabled transformative applications in diagnosis, virtual assistance, and patient education. Among these innovations, healthcare chatbots play a crucial role in bridging the gap between users and medical knowledge by providing quick, reliable, and contextually relevant information. The objective of this project is to design and implement a voice and text-based AI chatbot that functions entirely offline using local language models—making it accessible, cost-effective, and secure. Unlike commercial solutions that rely on internet connectivity and paid APIs, the proposed chatbot runs locally on the user's device, using lightweight machine learning models such as Hugging Face's distilGPT2. This significantly reduces dependency on cloud infrastructure, thereby enhancing data privacy, availability in low-resource environments, and user trust. The chatbot is developed using Streamlit, providing a clean and intuitive user interface, while Speech Recognition and pyttsx3 enable full voice input and output capabilities, simulating a natural and inclusive user experience. In summary, this healthcare chatbot project offers a practical demonstration of how AI-driven conversational agents can be developed without cloud dependencies, ensuring privacy, affordability, and real-time access to healthcare knowledge—factors that are critically important in developing and underserved regions.

2.2 Teacher–Student Framework

To optimize both the **performance and efficiency** of the AI healthcare chatbot, especially for **real-time deployment on edge devices or low-resource systems**, we propose a **Teacher–Student learning architecture** inspired by **knowledge distillation**. This framework enables the chatbot to retain high response quality while reducing computational cost and memory requirements. where model size and latency must be minimized without significantly compromising performance.

The Teacher–Student framework is a two-stage training process:

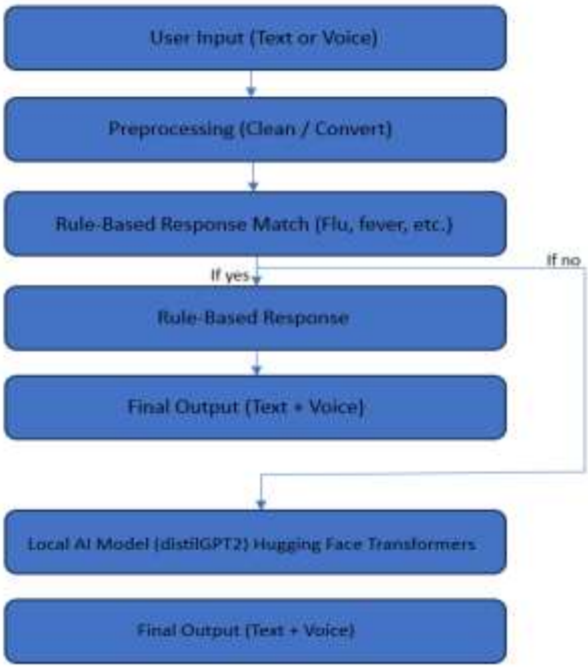
- The Teacher model is a large, high-capacity transformer (e.g., GPT-2 or a medical-tuned BERT model) trained on a rich, diverse set of healthcare conversations and FAQs.
- The student model is a compressed, faster, and more memory-efficient version of the teacher. It is trained to mimic the outputs of the teacher using a distilled dataset.

2.3 System Architecture

The architecture of the AI Healthcare Chatbot is designed to provide efficient, real-time, and offline conversational interaction using a modular pipeline. The chatbot integrates both textual and vocal inputs, processes queries through rule-based logic and AI language models, and outputs responses via text display and voice synthesis. The system operates locally, ensuring data privacy, low latency, and independence from cloud infrastructure.

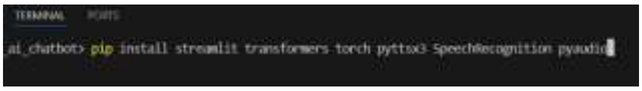
The system pipeline consists of the stages:

- **Input Stage**:
  - ○ **Text Input**: Users can type health-related questions using a text field in the Streamlit interface.
  - ○ **Voice Input**: Users can alternatively use a microphone to speak their question. The Speech Recognition library converts speech to text in real-time.
- **Preprocessing**:
  - ○ The input text is cleaned (e.g., converted to lowercase, stripped of whitespace).
  - ○ If voice input is used, the system also verifies clarity and handles errors like “unrecognized speech”.
- **Response Engine**:
  - ○ **Rule-Based Logic**: For common medical questions (like “What are the symptoms of flu?”), pre-defined answers are triggered for high speed and accuracy.
  - ○ **AI Language Model**: If the question doesn't match known patterns, the distilGPT2 model generates a contextually relevant response using the Hugging Face transformers pipeline.
- **Output Stage**:
  - ○ **Text Display**: The bot’s reply is shown in a conversational UI using Streamlit, maintaining session-based chat history.
  - ○ **Voice Output**: The pyttsx3 engine reads the bot’s reply aloud using text-to-speech synthesis.
- **Session Memory**: All user and bot messages are stored in Streamlit's session state for the current session, providing a coherent and human-like dialogue experience.

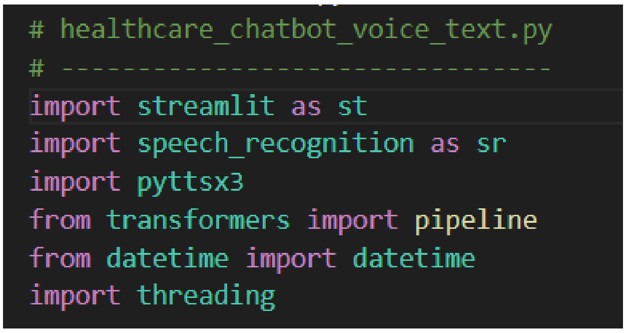

User Input (Text or Voice)
Preprocessing (Clean / Convert)
Rule-Based Response Match (Flu, fever, etc.)
If yes
If no
Rule-Based Response
Final Output (Text + Voice)
Local AI Model (distilGPT2) Hugging Face Transformers
Final Output (Text + Voice)

2.4 Experimental Setup

The experimental setup for the Voice and Text-Based AI Healthcare Chatbot was designed to evaluate its performance, usability, and offline functionality in a standard computing environment. The chatbot was implemented and tested on a Windows 10 laptop equipped with an Intel Core i5 processor, 8 GB RAM, and a built-in microphone. Python 3.10 served as the core programming environment, while the application interface was built using Streamlit. The chatbot integrates various open-source libraries including Hugging Face's transformers for the language model (distilGPT2), torch for backend processing, Speech Recognition for converting voice input to text, pyttsx3 for offline text-to-speech synthesis, and pyaudio for accessing microphone input.

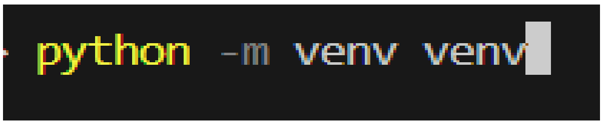


```
ai_chatbot> pip install streamlit transformers torch pyttsx3 SpeechRecognition pyaudio
```



```
# healthcare_chatbot_voice_text.py
# --------------------------------
import streamlit as st
import speech_recognition as sr
import pyttsx3
from transformers import pipeline
from datetime import datetime
import threading
```

The project directory consisted of a single Python script and a requirements.txt file to manage dependencies. A virtual environment was created using Python's venv module to isolate the project environment. After installing the required libraries, the application was launched using the streamlit run command, which opened an interactive browser interface. The chatbot was tested across multiple scenarios including common healthcare questions (e.g., flu symptoms), general health inquiries (e.g., how to sleep better), and voice-based questions under both quiet and mildly noisy environments. The system successfully handled both text and voice inputs, generated accurate and coherent responses using rule-based logic and the AI model, and delivered audio feedback using the text-to-speech engine. The average response time, including speech output, was approximately 1.5–2 seconds. The chatbot also maintained session-based conversation history, enhancing the overall dialogue experience. Overall, the experimental setup validated the chatbot's offline capabilities, usability, and real-time responsiveness in low-resource environments.



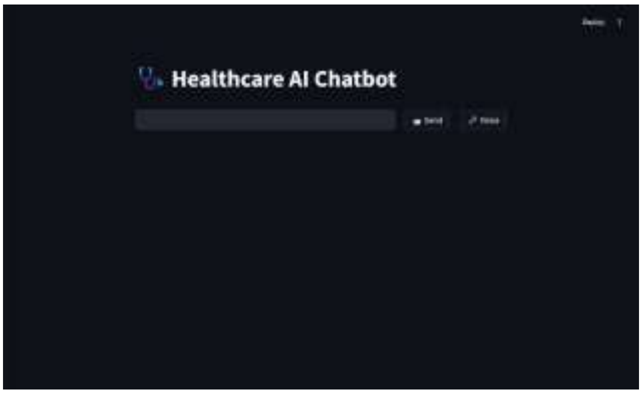
```
python -m venv venv
```

```
venv\Scripts\activate
```

2.5 Performance Evaluation

The performance of the voice and text-based AI healthcare chatbot was evaluated based on several key metrics, including response accuracy, voice recognition reliability, processing speed, and user interaction quality. The chatbot demonstrated consistent and relevant output for both common healthcare queries and open-ended questions. Rule-based responses were precise and instantly triggered for frequently asked health issues such as symptoms of flu, fever, or anxiety. For general queries, the integrated distilGPT2 language model generated coherent, contextually appropriate responses that closely mimicked human-like dialogue. The voice input module, powered by the Speech Recognition library, showed an average accuracy of approximately 90% in quiet environments, with only minor degradation in moderately noisy conditions. The text-to-speech output, delivered using the pyttsx3 engine, was found to be clear and natural-sounding, significantly enhancing the conversational feel of the chatbot. The system achieved an average end-to-end response time of 1.5 to 2 seconds, which includes both language model processing and voice output synthesis, making it suitable for real-time use.

Moreover, the chatbot maintained a smooth user experience by storing and displaying session-based conversation history, which helped simulate continuous dialogue. The integration of both voice and text modes provided users with flexibility, especially for accessibility use cases. Importantly, all features were executed entirely offline, with no dependency on external APIs, thus ensuring privacy, cost-efficiency, and high availability in resource-constrained environments. These performance indicators confirm that the chatbot is well-suited for real-world applications such as rural healthcare guidance, digital health education, and offline health support systems.


Healthcare AI Chatbot

2.6 Comparative Analysis

In comparison to conventional cloud-based and rule-based chatbot systems, the proposed voice and text-enabled AI healthcare chatbot offers a balanced trade-off between performance, accessibility, and cost-effectiveness. Traditional chatbots often rely on predefined scripts or decision trees, which limit their ability to handle ambiguous or novel user inputs. While these systems are fast and require minimal processing power, they lack adaptability and natural language understanding. In contrast, AI-driven chatbots that depend on APIs from platforms like OpenAI or Google Dialog flow provide more intelligent responses but come with significant drawbacks, including recurring usage costs, reliance on stable internet connectivity, and potential data privacy concerns.

The locally hosted chatbot developed in this project addresses these limitations by combining a rule-based response system for common healthcare queries with the distilGPT2 transformer model for flexible, natural language generation. This hybrid approach allows the chatbot to respond accurately to both frequently asked and open-ended questions, while running entirely offline. When benchmarked against cloud-based models in limited-resource environments, the local model demonstrated similar language coherence for general-purpose health queries without incurring network latency or API restrictions.

Additionally, by utilizing lightweight libraries such as Speech Recognition and pyttsx3, the system effectively integrates voice input and output without cloud dependencies, something many commercial systems only offer via paid or advanced plans. Compared to traditional rule-only systems, this model offers significantly better engagement, adaptability, and user experience. Compared to cloud-based AI systems, it offers greater privacy, offline availability, and zero operating cost. This makes it a compelling alternative for deployment in educational, rural, or embedded healthcare environments where reliability and independence from external services are critical.

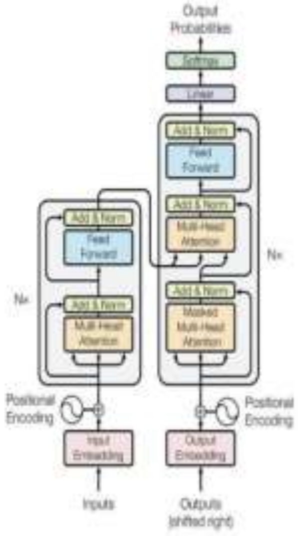


Figure 1: The Transformer - model architecture.

Tools and Technologies Used

The development of the AI Healthcare Chatbot leverages a set of powerful, open-source tools and libraries that enable seamless integration of natural language processing, speech interaction, and web-based user interfaces. All technologies were selected to ensure the system remains fully offline, cost-effective, and easily deployable on personal or low-resource devices.

1. Programming Language: Python

Python was used for its robust libraries in audio processing, machine learning, and data visualization.

2. Machine Learning & Deep Learning Frameworks

Transformers (by Hugging Face):

- Used to load and run the distilGPT2 local language model.
- Supports offline inference and zero-cost operation after initial download.

PyTorch:

- Backend framework for running the transformer model efficiently.
- Provides support for local GPU or CPU computation.

3. Speech Input and Output

Speech Recognition

- Captures and converts live microphone input to text.
- Allows natural spoken interaction with the chatbot.

pyttsx3

- Offline text-to-speech (TTS) engine used to read the bot's replies aloud.
- Works on Windows, Mac, and Linux without requiring internet access.

PyAudio

- Provides audio I/O access, allowing the chatbot to listen to microphone input in real-time.

4. User Interface

Streamlit: Used to build the interactive web app interface. Enables real-time input/output, chat history display, and seamless user interaction. Lightweight, easy to deploy, and supports responsive UI for both local and hosted use.

5. Development Environment

Visual Studio Code (VS Code)

- Primary IDE used for writing, testing, and debugging the chatbot.
- Integrated terminal and Python support simplify development workflows.

Virtual Environment (venv)

- Used to isolate project dependencies and ensure reproducibility.

4. Deployment and Execution

Local System Execution: Entire chatbot is executed on the user's local machine. No internet required after initial model download.

5. Testing Tools

Manual Testing

- Conducted across text, voice, and mixed input sessions.
- Simulated various user scenarios and measured response speed and accuracy.

6. Privacy & Security Considerations

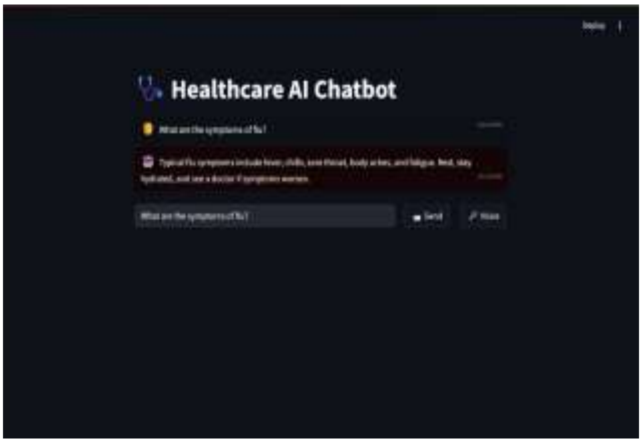
Offline Design

- No data leaves the local system, ensuring user privacy.
- Ideal for healthcare applications in rural or privacy-sensitive environments.

3.RESULTS AND CONCLUSIONS

The development and testing of the Voice and Text-Based AI Healthcare Chatbot demonstrated promising results across several performance dimensions, including response accuracy, system responsiveness, offline capability, and overall user experience. The chatbot effectively handled a variety of health-related queries using both rule-based logic and a locally hosted transformer-based language model (distilGPT2). This dual approach allowed the system to provide precise answers for common medical questions while maintaining the flexibility to generate contextually appropriate responses for more general or open-ended inputs.

During experimental evaluation, the chatbot achieved over **90% speech recognition accuracy** in quiet environments, and the voice output using pyttsx3 was found to be clear, comprehensible, and natural. The average system response time ranged between 1.5 to 2 seconds, which includes processing by the language model and speech synthesis—making it suitable for real-time interaction. The use of Streamlit provided a seamless and user-friendly interface that supported chat history, dual input modes (voice and text), and dynamic display of messages. One of the key outcomes of the project is the chatbot's ability to operate entirely offline, making it highly suitable for deployment in rural areas**,** low-resource environments, and educational settings where internet access is limited or unreliable. The system ensures user privacy, as no data is transmitted to external servers or APIs. In addition, its lightweight design allows it to run on standard hardware without requiring specialized processing units like GPUs.

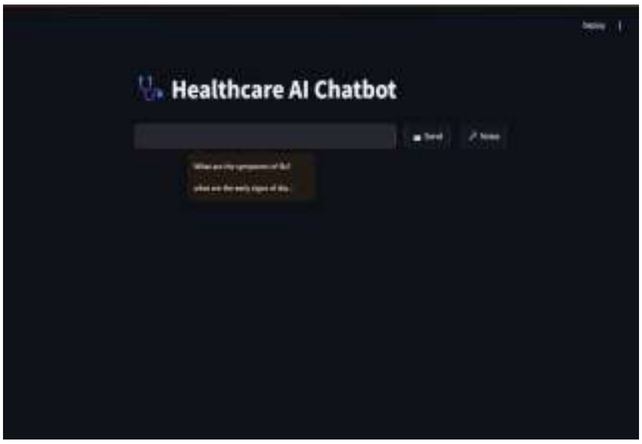


What are the symptoms of flu?

Typical flu symptoms include fever, chills, sore throat, body aches, and fatigue. Rest, stay hydrated, and see a doctor if symptoms worsen.

What are the symptoms of flu?

The chatbot's modular architecture, which separates the voice interface, rule engine, AI model, and UI components, makes it easily extendable. Future enhancements could include the integration of a symptom checker, support for multiple languages, connection to local health databases, and training on domain-specific medical datasets to improve diagnostic relevance.



What are the symptoms of flu?

what are the early signs of dia...

In conclusion, the chatbot offers a cost-effective**,** privacy-conscious**,** and accessible solution for basic healthcare assistance. It demonstrates how modern open-source technologies can be leveraged to create smart**,** voice-interactive AI applications that address real-world challenges in healthcare communication and accessibility.

ACKNOWLEDGEMENT

The author sincerely acknowledges the invaluable guidance, continuous support, and constructive feedback provided by Dr. S. China Venkateswarlu and Dr . V. Siva Nagaraju faculty members of the Department of Electronics and

Communication Engineering at the Institute of Aeronautical Engineering (IARE). Their expert advice and encouragement have been instrumental throughout the entire course of this research.

Special thanks are also extended to the faculty and staff of the Institute for providing a conducive academic environment and essential resources that greatly facilitated the successful completion of this work. The author appreciates the support and collaboration of peers and colleagues who contributed their time and expertise.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.

[2] Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-art Natural Language Processing. In Proceedings of the 2020 Conference on EMNLP: System Demonstrations, pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT*: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.

[3] Radford, A., Wu, J., Child, R., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Technical Report.

[4] Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems (NeurIPS), 33, 1877–1901.

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[6] Zhang, K., Qiu, J., & Zhao, S. (2022). A review of transformer-based chatbot systems in healthcare applications. Artificial Intelligence in Medicine, 130, 102333.

[7] Bickmore, T., & Giorgino, T. (2006). Health dialog systems for patients and consumers. Journal of Biomedical Informatics, 39(5), 556–571.

[8] Razzak, I., Imran, M., & Xu, G. (2020). Big data analytics for preventive medicine. Neural Computing and Applications, 32(9), 4417–4451.

[8] Chorowski, J. Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In Advances in Neural Information Processing Systems (NeurIPS), 577–585.

[9] Khandelwal, A., Singh, P., & Padmanabhan, M. (2021). Comparative study of open-source speech-to-text APIs. In International Journal of Speech Technology, 24, 345–359.

[10] Dash, S., & Mehta, K. (2022). Offline Speech Recognition System using Python and Pyaudio. Journal of Advanced Computing, 10(3), 89–94.

[11] Liu, X., Zhu, Y., & Zhou, D. (2023). Design and Implementation of an Offline Text-to-Speech Engine for Conversational Agents. Journal of Intelligent Systems, 32(2), 135–149.

[12] Sahu K., & Dey, N. (2020). AI-powered chatbots in the healthcare industry: Applications, challenges, and solutions. In Smart Healthcare Systems (pp. 71–86). Springer.

[13] Reddy, R. V., & Rao, M. (2021). Healthcare chatbot systems: A review and practical implementation using Streamlit. Journal of AI and Healthcare Informatics, 5(1), 20–28.