Chatbot Development Using AI

Abstract

Artificial Intelligence has revolutionized human–computer interaction by enabling systems to understand and respond to natural language queries. One of the most prominent applications of Artificial Intelligence in this domain is the development of intelligent chatbots capable of engaging in meaningful conversations with users.

This project focuses on the development of an Artificial Intelligence–based chatbot using Generative AI technology. The chatbot utilizes Google's Generative Artificial Intelligence platform and the Gemini Large Language Model to process user queries and generate intelligent responses. Unlike traditional rule-based chatbots, the system leverages natural language processing capabilities of modern AI models to interpret user input dynamically.

To ensure reliable and domain-specific responses, the chatbot incorporates a structured knowledge base containing detailed product and service information. Prompt engineering techniques are used to guide the model in generating relevant responses while restricting answers to the domain defined in the knowledge base.

The system is developed using Python programming language and implemented in the Google Colab environment. A web-based conversational interface is created using the Gradio framework, allowing users to interact with the chatbot through a browser. The chatbot is deployed using Gradio's public sharing mechanism, enabling real-time access through a generated web link.

The project demonstrates how Generative Artificial Intelligence models can be integrated with domain knowledge to create intelligent conversational agents capable of assisting users with information retrieval, product support, and troubleshooting guidance.

1. Introduction

Artificial Intelligence (AI) is a rapidly evolving field of computer science that focuses on developing intelligent systems capable of performing tasks that typically require human intelligence. These tasks include language understanding, reasoning, problem-solving, and learning from data.

In recent years, significant advancements in Natural Language Processing (NLP) and deep learning have led to the development of Large Language Models (LLMs). These models are capable of understanding and generating human-like text by analyzing vast amounts of linguistic data.

Chatbots are one of the most widely used applications of Natural Language Processing. A chatbot is a software system that can communicate with users through text or voice interfaces. Early chatbots were rule-based systems that relied on predefined scripts and keyword matching. However, modern chatbots utilize machine learning and deep learning models to understand user queries and generate dynamic responses.

Generative Artificial Intelligence has further transformed chatbot development by enabling systems to produce coherent and context-aware responses without relying on rigid rules. These systems can interpret natural language queries and generate responses based on contextual understanding.

This project focuses on developing a chatbot powered by Generative Artificial Intelligence. The chatbot integrates a knowledge base with a Large Language Model to provide intelligent and domain-specific responses. The system demonstrates how AI-powered conversational agents can be used to enhance user interaction and provide automated assistance.

2. Problem Statement

Organizations frequently provide technical support and product documentation through manuals, websites, and help centers. However, users often face difficulties navigating these resources efficiently.

Common problems include:

• Users struggle to locate relevant information within large documentation systems.
• Customer support teams spend significant time answering repetitive queries.
• Traditional FAQ systems cannot adapt to different ways users phrase their questions.
• Rule-based chatbots require extensive manual configuration and cannot handle complex queries.
• Static information systems lack conversational interaction.

These limitations create a need for an intelligent conversational system capable of understanding natural language queries and providing relevant responses.

The objective of this project is to design and implement an Artificial Intelligence–based chatbot that can interpret user queries and generate meaningful responses using a Generative AI model integrated with a domain-specific knowledge base.

3. Objectives of the Project

The primary objectives of the project are:

• To study the concepts of Artificial Intelligence and Generative AI.
• To develop a conversational chatbot capable of interacting with users using natural language.
• To integrate a domain-specific knowledge base with the AI model.
• To implement prompt engineering techniques for response control.
• To create a web-based chatbot interface using Gradio.
• To deploy the chatbot for real-time interaction.

4. Tools and Technologies Used

The chatbot system utilizes the following technologies:

Programming Language
Python

Artificial Intelligence Platform
Google Generative AI

Language Model
Gemini Large Language Model

Development Environment
Google Colab

User Interface Framework
Gradio

Libraries Used

• google-genai
• gradio

These tools enable the chatbot to process natural language queries and generate responses using AI-based models.

5. Methodology

The development of the chatbot follows a structured methodology consisting of multiple stages.

Requirement Analysis

The requirements of the system were identified, including user interaction, query processing, knowledge base integration, and response generation.

System Design

A modular architecture was designed to integrate the chatbot interface, knowledge base, and Generative AI model.

Implementation

The chatbot was implemented using Python programming language. The Gemini AI model was accessed through the Google Generative AI API.

Deployment

The chatbot was deployed using the Gradio framework to create an accessible web-based interface.

Testing

Various user queries were tested to evaluate the chatbot's response accuracy and reliability.

6. System Architecture

The chatbot system consists of the following components:

User Interface
Provides the interaction platform where users enter queries.

Prompt Engineering Layer
Constructs prompts containing system instructions, knowledge base information, and user input.

Knowledge Base
Stores domain-specific product and support information.

AI Model
Processes the prompt using a Large Language Model and generates responses.

Response Module
Displays the generated response to the user.

System Workflow

User Query
↓
Prompt Construction
↓
Knowledge Base Integration
↓
AI Model Processing
↓
Response Generation
↓
Display Response

7. Development Process

The development of the chatbot involved several steps.

Step 1: Creating the Development Environment

Google Colab was used as the development platform because it provides a cloud-based environment with Python support.

Step 2: Installing Required Libraries

The required libraries were installed in the Colab environment.

Example command:

pip install google-genai gradio

Step 3: Configuring the Generative AI API

An API key was generated from the Google Generative AI platform and used to configure the AI client.

Step 4: Creating the Knowledge Base

A structured knowledge base containing product descriptions, features, troubleshooting information, and ordering details was created.

Step 5: Designing the Prompt Structure

Prompt engineering techniques were used to combine system instructions, knowledge base content, and user queries.

Step 6: Implementing Chatbot Logic

Python functions were written to process user input, send prompts to the AI model, and

display responses.

Step 7: Creating the Chat Interface

A conversational interface was built using the Gradio framework.

Step 8: Deploying the Chatbot

The chatbot was deployed using Gradio's public sharing feature, generating a public link for user interaction.

8. Implementation

The chatbot was implemented using Python and integrated with Google Generative AI.

The following code represents the complete implementation of the chatbot system.

```
from google import genai
import gradio as gr
API_KEY = "AIzaSyBos-ergGuZ8a2ItA5y-TSxCQ8tDliJZ9s"
MODEL = "models/gemini-2.5-flash"
client = genai.Client(api_key=API_KEY)
print("\nAI TECHNICAL SUPPORT ASSISTANT READY")
print("Type 'exit' to stop\n")
history = []
knowledge_base = """
========================================
  COMPANY TECH SUPPORT KNOWLEDGE BASE
========================================
Company Overview
----------------
NovaTech Systems develops consumer electronics and smart automation
products for home and office environments. Products are designed for
reliability, remote management, and seamless integration with mobile
```

applications.
All NovaTech devices support firmware updates and cloud synchronisation through the NovaTech mobile application.

Official Company Website

NovaTech Products can be purchased directly from the official company website.
Website:
https://www.novatechsystems.com
Ordering Information:
-Customers can browse all Novatechproducts on the Website
-Orders can be placed directly through the online Store
-The Website provides product specifications, pricing, and availability.
-Customers can track orders through their online account
Shipping and Delivery:
-Standard delivery typically takes 3-5 business days.
-Express shipping options maybe available depending on location.
Customer Support Portal:
The website also includes:
-Product documentation
-Setup Guides
-Firmware downloads
-Troubleshooting Resources
-Customer Support Ticket Submission

Product: SmartHome Hub X1

Description:
SmartHome Hub X1 is a central automation controller designed to manage multiple smart devices within a home network.
Key Features:
- Wifi and Zigbee connectivity
- Centralised Device Control

- Automation Routines and Scheduling
- Voice Assistant integration
- Remote Access through mobile App
- Multi Room Device Grouping

Typical Use Cases:

-Automating Lighting Schedules

-Controlling Thermostats remotely

-Monitoring Security Devices

-Managing Energy Consumption

Trouble Shooting:

-Restart the Hub if connection drops

-Ensure firmware is up to date

-Place hub within router range

Product: Smart Secure Camera C200

Description:

Indoor Security Monitoring Camera Designed for Home Surveillance

Key Features:

-1080p HD Video Recording

-Night Vision infrared sensors

-Motion Detection Alerts

-Mobile Live Streaming

-Cloud Video Backup

-Two way Audio Communication

Typical Use Cases:

-Home Monitoring

-Pet Monitoring

-Child Safety Monitoring

Troubleshooting:

-Verify Motion Detection Sensitivity

-Check Camera Wifi Connection

-Restart Device if video feed freezes

Product:NovaTherm Smart Thermostat T500

Description:

A Programmable Thermostat Designed to optimise indoor temperature and reduce energy consumption.

Key Features:

- Smart Temperature Scheduling
- Energy Consumption Analytics
- Mobile Remote Control
- Voice Assistant Integration
- Adaptive Learning of User Preferences

Benefits:

-Energy Savings

-Comfortable Climate Control

- Automated Heating and cooling

Product:NovaLight Smart LED Panel L100

Description:

Energy Efficient Smart Lighting Panel with Remote Brightness and Color Temperature Control

Key Features:

-Adjustable Brightness Levels

-Warm to Cool Light Spectrum

-Mobile App Control

-Automation Scheduling

-Energy Efficient LED Technology

Applications:

- Home Lighting
- Office Lighting
- Studio Environments

Customer Support Services

```
NovaTech offers technical assistance for:
-Device Setup and installation
-Firmware Updates
-Troubleshooting Connectivity Issues
-Software Configuration
-Device Compatibility Guidance
Support Channels:
-Email support
-Live chat Support
-Mobile App Ticket system
-Online Trouble shooting Documentation
"""
system_prompt = """
You are NovaTech's official AI Technical Support Assistant.
Your Responsibilities:
-Help Users understand NovaTech Products
-Assist with Troubleshooting and setup
-Provide clear explanations of Product Features
-Offer Guidance based on Knowledge Base
Behavior guidelines:
1. Answer ONLY questions related to NovaTech Products, Services, or
support information in the Knowledge Base.
2. If the question is unrelated to the knowledge base , respond with:
"I can provide support only for NovaTech Products and services."
3.Provide Professional, clear and structured responses.
4.When helpful,expand explanations using logical context while staying
consistent with the knowledge base.
5. Avoid speculation or unsupported claims.
6.Maintain a professional and helpful tone.
7.When users ask about purchasing or ordering products or request order
info or summary, guide them directly to official company website.
"""
def chatbot(user_input, chat_history):
    prompt = f"""
```

```
{system_prompt}
Knowledge Base:
{knowledge_base}
User Question:
{user_input}
"""
    response = client.models.generate_content(
        model=MODEL,
        contents=prompt,
        config={
            "temperature": 0.35,
            "top_p": 0.9,
            "max_output_tokens": 500
        }
    )
    reply = response.text
    chat_history.append({"role": "user", "content": user_input})
    chat_history.append({"role": "assistant", "content": reply})
    return "", chat_history
with gr.Blocks() as demo:
    gr.Markdown(" NovaTech AI Support Assistant")
    chatbot_ui = gr.Chatbot(type="messages")
    msg = gr.Textbox(
        placeholder="Ask a question about NovaTech products..."
    )
    msg.submit(chatbot, inputs=[msg, chatbot_ui], outputs=[msg,
chatbot_ui])
demo.launch(share=True)
```

9. Testing and Debugging

Several test cases were used to verify system functionality.

Test Case 1
User asks about product features.

Expected Result
Chatbot provides relevant product information.

Test Case 2
User asks where to purchase products.

Expected Result
Chatbot directs user to official website.

Test Case 3
User asks unrelated question.

Expected Result
Chatbot responds that it only supports NovaTech products.

10. Results and Performance Analysis

The chatbot demonstrated strong performance in handling domain-specific queries.

Response Accuracy
High accuracy for queries related to the knowledge base.

Response Speed
Responses were generated within a few seconds.

User Experience
The Gradio interface provided smooth and interactive communication.

The system successfully integrated Generative AI with domain knowledge to produce meaningful conversational responses.

11. Future Enhancements

Future improvements may include:

- Integration with vector databases for intelligent knowledge retrieval.
- Voice-based interaction using speech recognition.
- Multi-language support for global users.
- Integration with enterprise databases and APIs.
- Enhanced conversation memory for long interactions.

12. Conclusion

This project demonstrates the development of a conversational chatbot using Generative Artificial Intelligence. By integrating a Large Language Model with a structured knowledge base, the system can interpret user queries and generate intelligent responses.

The project highlights the potential of AI-powered conversational systems in improving user interaction and automating information retrieval. With further development and integration, such systems can play a significant role in customer support, education, healthcare, and enterprise automation.