

Code Alpha

Artificial Intelligence

Internship Report

Task 2: Chatbot for FAQs

Student name: Saira Arshad

STUDENT ID: CA/SE1/16114

Date: October 2025

Duration : 1 month

Table of Contents

1. Introduction

1.1 Overview

1.2 Objective of the Project

2. Literature Review / Background

2.1 What is a Chatbot?

2.2 Role of NLP in Chatbots

2.3 Use of Cosine Similarity and TF-IDF

3. System Requirements

3.1 Hardware Requirements

3.2 Software Requirements

4. Tools and Technologies Used

4.1 Python Programming Language

4.2 Tkinter Library

4.3 Scikit-learn Library (TF-IDF, Cosine Similarity)

5. System Design and Working

5.1 Architecture of the Chatbot

5.2 Data Flow and Processing

5.3 Chatbot Algorithm

6. Implementation

6.1 Preprocessing the FAQs

6.2 Matching User Queries

6.3 GUI Design Using Tkinter

6.4 Python Code Overview

7. Results and Output

7.1 Sample Chat Interactions

7.2 Screenshots of Chatbot Interface

8. Advantages and Limitations

8.1 Advantages

8.2 Limitations

9. Applications

9.1 Customer Support

9.2 E-commerce

9.3 Education and Helpdesks

10. Conclusion and Future Scope

10.1 Conclusion

10.2 Future Enhancements

11. References

TASK 2: CHATBOT FOR FAQS

Project Title: FAQ Chatbot Using NLP

1. Introduction

1.1 Overview

This project aims to develop a chatbot that can automatically respond to Frequently Asked Questions (FAQs) related to a particular topic or product. The chatbot uses Natural Language Processing (NLP) to understand user queries and provide accurate answers. It matches user input with a predefined list of questions and retrieves the most relevant response. By using TF-IDF vectorization and cosine similarity, the chatbot ensures intelligent matching between user queries and stored FAQs.

1.2 Objective of the Project

The main objective of this project is to create an intelligent FAQ chatbot capable of:

- Understanding user queries in natural language.
- Matching questions with stored FAQs using NLP techniques.
- Providing instant and accurate answers automatically.
- Offering a simple and user-friendly chat interface for easy interaction.

2. Literature Review / Background

2.1 What is a Chatbot?

A chatbot is an artificial intelligence program that simulates human conversation through text or voice interactions. It is designed to understand user queries, process them, and deliver relevant responses. Chatbots are commonly used in customer service, websites, and mobile apps.

2.2 Role of NLP in Chatbots

Natural Language Processing (NLP) is the core technology behind chatbots. It allows the system to interpret, analyze, and understand human language. NLP techniques like tokenization, stemming, and vectorization enable the chatbot to process textual data effectively.

2.3 Use of Cosine Similarity and TF-IDF

The TF-IDF (Term Frequency–Inverse Document Frequency) model is used to convert text into numerical vectors, representing the importance of words. Cosine similarity then measures how closely the user's question matches each FAQ, selecting the best response based on similarity scores.

3. System Requirements

3.1 Hardware Requirements

- Processor: Intel Core i3 or higher
- RAM: 4 GB or above
- Storage: Minimum 500 MB free space
- Display: Standard monitor for GUI display

3.2 Software Requirements

- Operating System: Windows 10 or higher
- Python 3.8+
- Libraries: Tkinter, scikit-learn
- IDE: VS Code, PyCharm, or IDLE

4. Tools and Technologies Used

4.1 Python Programming Language

Python is chosen for its simplicity and extensive support for machine learning and NLP libraries. It allows fast development and easy integration with GUI frameworks like Tkinter.

4.2 Tkinter Library

Tkinter is a standard GUI library in Python used to create desktop applications. In this project, it provides a simple chat interface where users can type and receive chatbot replies.

4.3 Scikit-learn Library (TF-IDF, Cosine Similarity)

Scikit-learn offers tools for text processing and similarity calculations. The TF-IDF vectorizer converts text into numerical form, and cosine similarity identifies the most relevant FAQ.

5. System Design and Working

5.1 Architecture of the Chatbot

1. User enters a question through the chat interface.
2. The system preprocesses the text and converts it into a vector.
3. The chatbot compares it with stored FAQ vectors using cosine similarity.
4. The most similar FAQ answer is selected and displayed.

5.2 Data Flow and Processing

Input text → Preprocessing → TF-IDF Vectorization → Similarity Calculation → Response Generation → Output on GUI.

5.3 Chatbot Algorithm

1. Load predefined FAQs and answers.
2. Convert all questions into TF-IDF vectors.
3. Accept user input.
4. Compute cosine similarity between input and FAQs.
5. Retrieve and display the best-matched answer.

6. Implementation

6.1 Preprocessing the FAQs

Each question is cleaned, lowercased, and vectorized using TF-IDF. This helps the model understand word importance.

6.2 Matching User Queries

When a user types a query, it is transformed into the same TF-IDF space, and cosine similarity finds the closest match.

6.3 GUI Design Using Tkinter

Tkinter provides a user-friendly window with a chat area, input box, and send button. Messages from both the user and chatbot appear in a conversational format.

6.4 Python Code Overview

The code imports libraries, defines FAQs, builds TF-IDF vectors, and starts a Tkinter interface. The chatbot continuously listens for user input and responds accordingly.

7. Results and Output

7.1 Sample Chat Interactions

User: What is your return policy?

Chatbot: Our return policy allows returns within 30 days of purchase.

User: How can I contact support?

Chatbot: You can contact our support team via email or live chat.

7.2 Screenshots of Chatbot Interface

(Insert screenshots of your chatbot window here)

8. Advantages and Limitations

8.1 Advantages

- Provides instant answers to FAQs.
- Reduces human workload in customer support.

- Uses NLP for intelligent text understanding.
- Easy to use and modify for other topics.

8.2 Limitations

- Limited to predefined questions.
- Cannot handle complex, multi-sentence conversations.
- Requires manual updating of FAQs.

9. Applications

9.1 Customer Support

Used to answer customer queries 24/7 efficiently.

9.2 E-commerce

Helps users check order status, policies, and payment methods.

9.3 Education and Helpdesks

Can be integrated into academic systems to assist students with common queries.

10. Conclusion and Future Scope

10.1 Conclusion

The FAQ Chatbot successfully demonstrates how NLP and machine learning can automate question answering. It provides quick, accurate, and consistent responses through an easy-to-use chat interface.

10.2 Future Enhancements

- Adding voice interaction using speech recognition.
- Integrating with databases for dynamic FAQ updates.
- Using deep learning models (like BERT) for better understanding.
- Deploying as a web-based chatbot.

11. References

- Python Official Documentation – <https://www.python.org>
- Scikit-learn Library – <https://scikit-learn.org>
- Tkinter GUI Docs – <https://docs.python.org/3/library/tkinter.html>
- NLP Tutorials – Towards Data Science, GeeksforGeeks