West Visayas State University

COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

La Paz, Iloilo City, Philippines

"Chatbot"

A Software Documentation

Presented to the Faculty of the

College of Information and Communications Technology

West Visayas State University

La Paz, Iloilo City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

by

Fatima Grace T. Apinan

Gloria Marie P. Lamsin

Princes Rose G. Manuel

Windy C. Sabolbora

Nelwin J. Serra

May 2024

West Visayas State University

COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

La Paz, Iloilo City, Philippines

Overview

This document provides an overview of the chatbot's architecture, including the backend, frontend, and the libraries used in its development.

Githublink : https://github.com/Hime-chann/Chatbot

Backend

Architecture

The backend of the chatbot is intricately designed to handle several crucial functions, including natural language processing (NLP), conversation management, and integration with external APIs. At its core, the NLP component is responsible for understanding and processing user inputs, allowing the chatbot to interpret and respond to a wide variety of language patterns and nuances. This involves leveraging advanced machine learning algorithms and language models to accurately decode the intent behind user queries and generate appropriate responses.

To manage conversations effectively, the backend employs robust conversation management techniques. This ensures the chatbot can maintain context, track the flow of interactions, and provide coherent and relevant replies even over extended exchanges. The

conversation management system keeps track of previous interactions, user preferences, and context, thereby enabling the chatbot to deliver a more personalized and seamless user experience.

Technologies

- Language: Python

- Framework: Flask

Key Components

1. API Gateway: Manages incoming requests and routes them to appropriate services.

2. NLP Service: Processes user inputs, performs entity recognition, and determines intent.

3. Dialogue Manager: Manages the state of the conversation and generates responses.

4. Integration Service: Connects to external APIs and services.

Libraries

- Flask: Micro web framework for API development.

- spaCy: NLP library for entity recognition and intent classification.

- NLTK: Toolkit for working with human language data.

- TextBlob: Library for sentiment analysis.

- Requests: Library for making HTTP requests to external APIs.

- Random: Standard library for generating random responses.

Frontend

Architecture

The frontend of the chatbot is a web-based interface that facilitates interaction with the backend. Designed with responsiveness and user-friendliness in mind, it ensures that users can communicate seamlessly with the chatbot across various devices and screen sizes. This interface provides an intuitive and engaging experience, making it easy for users to input their queries and receive responses. By focusing on usability and accessibility, the frontend enhances the overall efficiency and effectiveness of the chatbot, ensuring smooth and effective communication between the user and the system.

Technologies

- Language: HTML, CSS, JavaScript

- Framework: Flask (for rendering templates)

Key Components

1. Chat Interface: The main user interface for interacting with the chatbot.

2. Message List: Displays the conversation history.

3. Input Field: Allows users to type and send messages.

4. Send Button: Facilitates sending messages to the chatbot

West Visayas State University

COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

La Paz, Iloilo City, Philippines

Libraries

- Flask: Used for rendering HTML templates.

- Fetch API: Used for making AJAX requests to the backend.

Libraries Overview

This section provides the overview of the libraries used for

creating the chatbot.

Backend Libraries

1. Flask: Provides the core framework for building the backend APIs.

2. spaCy: Powers the NLP capabilities for understanding user inputs.

3. NLTK: Complements spaCy for additional NLP functionalities.

4. TextBlob: Used for sentiment analysis.

5. Requests: Simplifies HTTP requests to external services.

6. Random: Facilitates generating random responses.

Frontend Libraries

1. Flask: Used for rendering HTML templates.

2. Fetch API: Used for making AJAX requests to the backend.

West Visayas State University

COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

La Paz, Iloilo City, Philippines

Conclusion

This documentation outlines the essential components and technologies used in the development of the chatbot. The backend leverages Python and Flask for robust API and NLP capabilities, while the frontend uses HTML, CSS, and JavaScript to create a responsive and interactive user experience. The integration of various libraries ensures efficient development and maintenance of the chatbot. The integration of various libraries ensures efficient development and maintenance of the chatbot.