Experiment No.1
Study various applications of NLP and Formulate the Problem Statement for Mini Project based on chosen real world NLP applications
Date of Performance:
Date of Submission:

Group Members:

Vaishnavi Saindane(74)
Chetana Manjarekar(70)
Janvi Chavan(06)

Chatbot in Dialogflow

Introduction

Chatbots have become an essential part of modern business operations. They offer a cost-effective way to provide instant support and information to customers. Dialogflow, a powerful natural language processing (NLP) platform by Google, enables developers to create conversational interfaces for websites, mobile applications, and other platforms. In this article, we'll walk you through the process of building a chatbot using Dialogflow and demonstrate how you can enhance its capabilities by integrating it with ChatGPT, a state-of-the-art language model.

Problem Statement

Many businesses face the challenge of providing efficient and round-the-clock customer support. Human agents have limitations, such as being unavailable during non-working hours and the potential for human error. These limitations can result in customer dissatisfaction and lost business opportunities.

Scope of Work

To address the problem statement, our goal is to build a chatbot that can provide instant responses to customer queries, handle routine tasks, and offer a seamless user experience. Additionally, we want to leverage ChatGPT, a language model developed by OpenAI, to enhance the chatbot's capabilities.

Chatbot Building Steps

1. Reason for Choosing DialogFlow: Understanding why Dialogflow is an excellent choice for chatbot development.

2. Dialogflow Setup: A step-by-step guide on setting up a Dialogflow agent.

3. Dialogflow Intents: Creating intents to define how the chatbot should respond to user inputs.

4. Dialogflow Entities: Defining entities to extract specific information from user queries.

5. Dialogflow Contexts: Using contexts to maintain conversational context.

6. Fulfillment: Setting up fulfillment to handle backend interactions.

7. Database Setup: Creating a database to store and retrieve information.

8. Backend Setup: Preparing the backend for communication with Dialogflow.

9. FastAPI Python Backend Coding: Building the backend logic using FastAPI.

Reason for Choosing DialogFlow

1. Powerful NLP Capabilities: Dialogflow's NLP engine can understand user input in multiple languages and contexts, making it suitable for a wide range of applications.

2. Ease of Use: Its intuitive interface allows developers to design and train chatbots quickly, with minimal coding effort.

3. Integration with Other Services: Dialogflow seamlessly integrates with Google Cloud services, making it an attractive choice for businesses already using Google's ecosystem.

4. Versatility: Dialogflow can be used in various applications, including web and mobile apps, voice assistants, and more.

FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.6+ based on standard Python type hints. We will use FastAPI to build the backend logic, enabling seamless communication between Dialogflow and external services.