**Building AI Chatbots: A Comprehensive Guide**

This document outlines two methods for building AI chatbots: using Botpress for static FAQ responses and fine-tuning the LLaMA 3.2 model for custom AI interactions. Both approaches cater to different needs and provide robust chatbot solutions.

**Method 1: Creating a Chatbot with Botpress**

* **Problem**

The objective was to create a chatbot capable of answering user queries based on predefined FAQs and handling out-of-scope queries gracefully.

* **Solution**

We utilized Botpress, a user-friendly platform for building conversational chatbots, to design and deploy a chatbot.

1. **Approach:**

* Used Botpress's web application for rapid development without requiring local installation.
* Built the chatbot by uploading a document containing FAQ data.
* Configured static responses for FAQs while integrating fallback options for queries outside the provided dataset.

1. **Deployment:**

* Deployed the [Botpress AIchatbot](https://cdn.botpress.cloud/webchat/v2.2/shareable.html?configUrl=https://files.bpcontent.cloud/2024/11/21/17/20241121173505-LS2FEGE2.json) chatbot directly via its web application:

**Method 2: Building a Custom AI Chatbot with LLaMA 3.2 and Ollama**

* **Problem**

We needed a chatbot tailored to a specific dataset, requiring a custom AI model capable of handling complex queries and dynamic responses.

* **Solution**

We fine-tuned the LLaMA 3.2 model on a custom dataset and used Ollama to run the model locally. A Python-based API was built using FastAPI for integration.

* **Steps Taken**

#### **Downloading and Running LLaMA 3.2 Locally with Ollama**

* Ollama was installed as the local runtime to manage and serve the LLaMA model.
* The latest version of Ollama was installed and configured.
* The LLaMA 3.2 model was pulled into Ollama and run using its built-in serve command.

#### **Fine-Tuning the Model**

* The LLaMA model was fine-tuned with a dataset specific to the chatbot's target domain.
* This involved preprocessing the dataset and creating a fine-tuned version of the model.

#### **Creating an API with FastAPI**

* Built an API to interact with the fine-tuned model using Ollama's Python Library.
* The API enabled user queries to be processed dynamically by the model and returned meaningful responses.

#### 

#### **Deployment and Hosting**

* The solution was deployed locally, allowing full control over data privacy and processing.

#### 

#### **Key Benefits:**

* + The model could answer specific domain-related queries effectively after fine-tuning.
  + Using Ollama ensured efficient local hosting without external dependencies.
* **GitHub Repository:** [**AIChatbot Demo**](https://github.com/Arpit-K-Sharma/AI_Chatbot.git)

## **Conclusion**

This document outlines two distinct approaches to creating AI chatbots:

1. Botpress for quick, FAQ-based solutions.
2. LLaMA and Ollama for custom AI chatbots fine-tuned to specific datasets.

Both methods demonstrate flexibility in chatbot development, providing scalable solutions depending on project needs.