Bangla Chat Bot Documentation

Introduction

This documentation provides detailed guidelines for installing, using, and understanding the Bangla Chat Bot. This chatbot is implemented using PyTorch and provides a basic yet effective approach for natural language understanding and response generation in the Bengali language.

Features

- Simple chatbot implementation with a feedforward neural network having two hidden layers.
- Easy customization for different use cases by modifying intents.json.
- Language support for Bangla and English, with capabilities for understanding and generating responses in Bengali and English. Also easily customizable for adding new languages.

Project Structure

- **app.py:** Main application file for the chatbot interface.
- **chat.py:** Handles chat functionality and response generation.
- **model.py:** Defines the neural network model.
- **nltk_utils.py:** Contains utility functions for text processing.
- **train.py:** Script for training the neural network model.
- **intents.json:** Contains intents and patterns for training.
- **data.pth:** The trained model file.

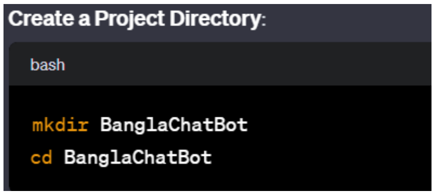
Installation Guide

Prerequisites

- Python 3.x
- Pip package manager
- Virtual environment (optional but recommended)

Setting Up the Environment

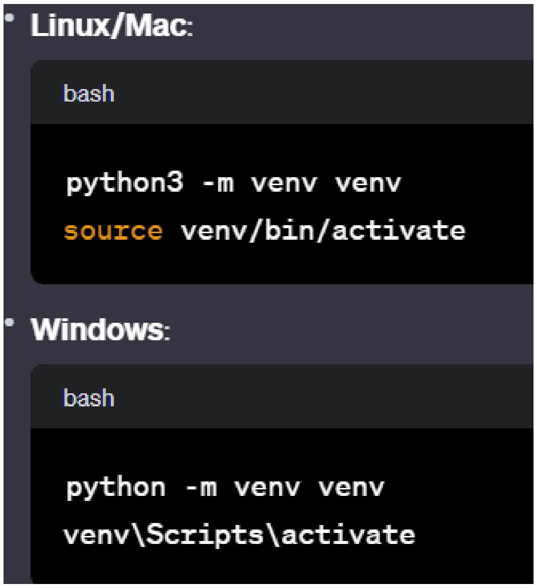
1. *Create a Project Directory:*



Create a Project Directory:

```bash
mkdir BanglaChatBot
cd BanglaChatBot
```

2. *Create and Activate a Virtual Environment:*



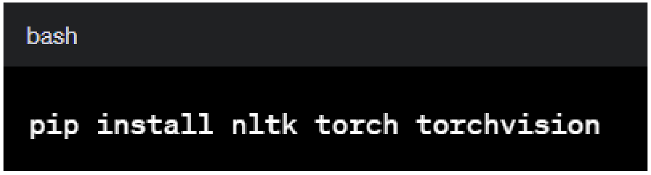
- **Linux/Mac:**

```bash
python3 -m venv venv
source venv/bin/activate
```

- **Windows:**

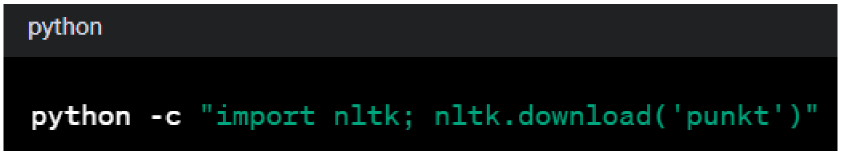
```bash
python -m venv venv
venv\Scripts\activate
```

3. **Install Dependencies:**
   - **PyTorch:** Visit the official PyTorch website for installation instructions.(https://pytorch.org/get-started/locally/)
   - **Install NLTK and other dependencies:**



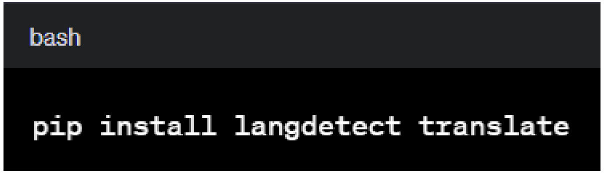
```bash
pip install nltk torch torchvision
```

   - **Download NLTK Tokenizer:**



```python
python -c "import nltk; nltk.download('punkt')"
```

   - **Install Language Detection and Translation Libraries:**

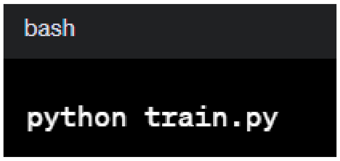


```bash
pip install langdetect translate
```

Usage Instructions

Training the Model

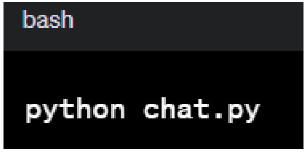
1. **Run the Training Script:**



```bash
python train.py
```

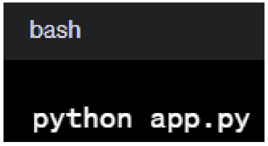
This will process the data in intents.json and train the neural network, saving the trained model to data.pth.

2. **Running the Chatbot**



```bash
python chat.py
```

3. **To interact with the chatbot using the graphical user interface, simply run the app.py file:**

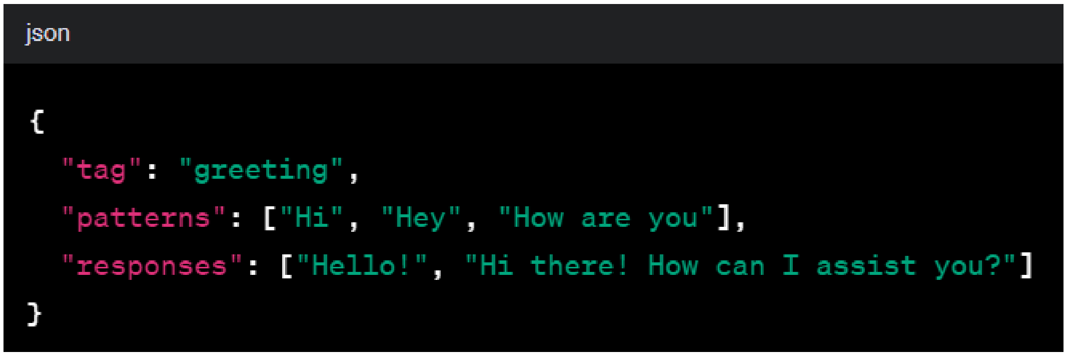


```bash
python app.py
```

This will open the Python GUI interface where you can directly interact with the chatbot.

Customization

- **Customizing Intents**:
  - Modify the **intents.json** file to include new patterns and responses.
  - Example intent structure:



```json
{
  "tag": "greeting",
  "patterns": ["Hi", "Hey", "How are you"],
  "responses": ["Hello!", "Hi there! How can I assist you?"]
}
```

FAQs

Q: How do I add a new language to the chatbot?

A: The current implementation primarily supports Bengali. To add support for a new language, you would need to provide data in the desired language in intents.json and possibly adjust the NLP preprocessing in nltk_utils.py.

Q: Can this chatbot handle context in conversations?

A: The current implementation has basic contextual understanding based on the defined intents and patterns. For advanced context handling, further development and integration of context-aware algorithms would be required.

Q: How do I add a new conversation topic?

A: Add a new intent in **intents.json** with patterns and responses, and retrain the model.

Q: What languages does this chatbot support?

A: Currently, it is designed for Bangla and English, but can be adapted for other languages by modifying the chat.py file.

Q: How can I improve the chatbot's accuracy?

A: Increase the dataset size, tweak the neural network architecture. Most importantly use language Models.