Rule-Based Chatbot

Directed by: Kisor.G

Project Overview

The project focuses on developing an intelligent chatbot for Amrita Vishwa Vidyapeetham. The chatbot is designed to address frequently asked queries related to admissions, courses, campus locations, and more. By leveraging machine learning (ML) and natural language processing (NLP), the chatbot delivers accurate and contextually relevant responses. The application is deployed as a web-based platform for easy accessibility.

Objectives

- Create a user-friendly chatbot capable of understanding and responding to user queries.
- Utilize ML to classify user messages into predefined intents.
- Implement NLP techniques for robust preprocessing of text inputs.
- Develop and deploy the chatbot as an interactive web application using Flask.

Project Structure

Key Components

1. app.py:
   A Python-based Flask application that serves as the backend of the chatbot. It handles user input, model inference, and response generation.
2. chatbot_intents.csv:
   A structured dataset containing predefined intents, user query patterns, and corresponding responses.
3. Machine Learning Module:
   - A TensorFlow/Keras-based neural network for intent classification.
   - Preprocessing pipelines for converting text data into numerical formats suitable for training.

4. Frontend:
   A simple HTML/CSS interface for user interaction. Users can type queries and receive real-time responses.
5. Dependencies:
   - Flask: Web framework for deployment.
   - TensorFlow/Keras: For model development and prediction.
   - NLTK: For text preprocessing.
   - CSV: For managing the intents dataset.

Implementation Details

A. Natural Language Processing (NLP)

1. Text Preprocessing:
   - Tokenization: Splitting sentences into words using nltk.word_tokenize.
   - Lemmatization: Reducing words to their base forms (e.g., "running" → "run").
   - Stopword Removal: Eliminating common words like "is" and "the" that don't contribute to meaning.
   - Special Character Removal: Cleaning text by removing punctuation and non-alphanumeric characters.
2. Bag of Words (BoW):
   - Converts textual data into numerical arrays for machine learning compatibility.
   - Ensures uniform vector representation of text inputs.

B. Machine Learning Model

1. Data Preparation:

    - Inputs: Bag of Words representations of user queries (patterns).
    - Outputs: One-hot encoded labels representing predefined intents.
2. Model Architecture:
    - Input Layer: Accepts BoW vectors.
    - Hidden Layers: Two dense layers with ReLU activation.
    - Output Layer: A dense layer with Softmax activation for multiclass classification.
3. Training Configuration:
    - Optimizer: Adam for adaptive learning rates.
    - Loss Function: Categorical Crossentropy for classification.
    - Training Parameters:
        - Epochs: 200
        - Batch Size: 5
4. Model Deployment:
    - The trained model is saved as a .h5 file and loaded in the Flask application for inference.

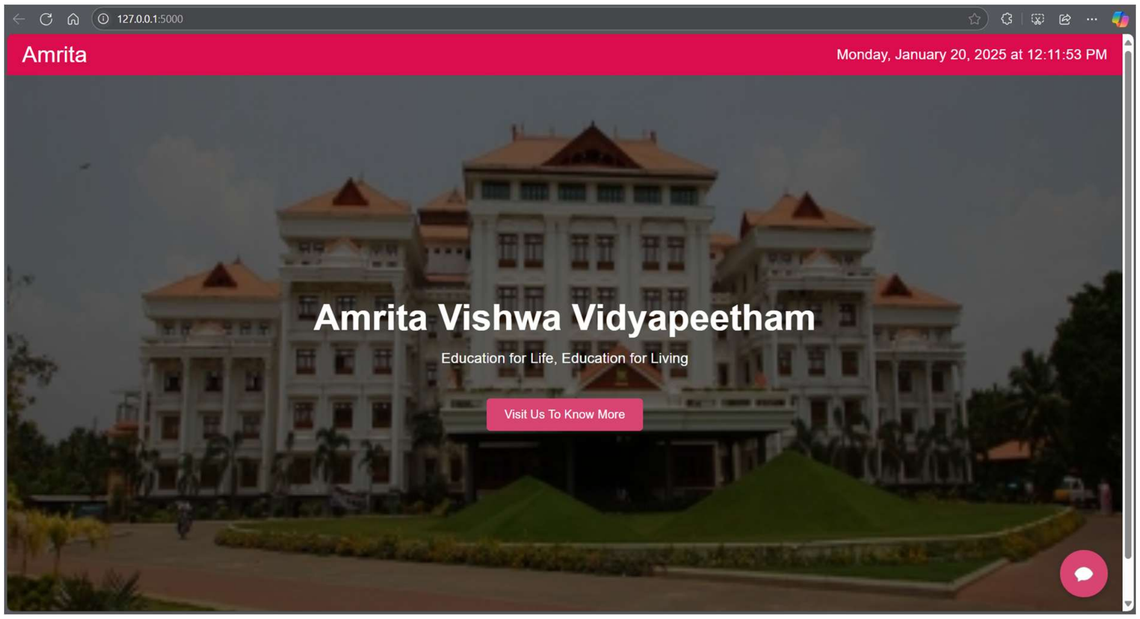
C. Flask-Based Web Application

1. Endpoints:
    - /: Renders the home page with the chatbot interface.
    - /get: Handles POST requests, processes user queries, and returns appropriate responses.
2. Response Generation Logic:
    - User inputs are preprocessed using NLP techniques.
    - The processed text is passed to the ML model for intent classification.

- The corresponding response is retrieved from chatbot_intents.csv.

3. Fallback Responses:
   - If the model fails to classify a query, a generic fallback message is provided:
     - Example: "I'm sorry, I don't understand your question. Please try rephrasing."

Interface


127.0.0.1:5000
Amrita
Monday, January 20, 2025 at 12:11:53 PM
Amrita Vishwa Vidyapeetham
Education for Life, Education for Living
Visit Us To Know More

This is a web application layout, likely a homepage for Amrita Vishwa Vidyapeetham:

Header

- **Top Left:** "Amrita" is displayed on a pink background, representing the brand.
- **Top Right:** Current date and time are displayed for real-time updates.

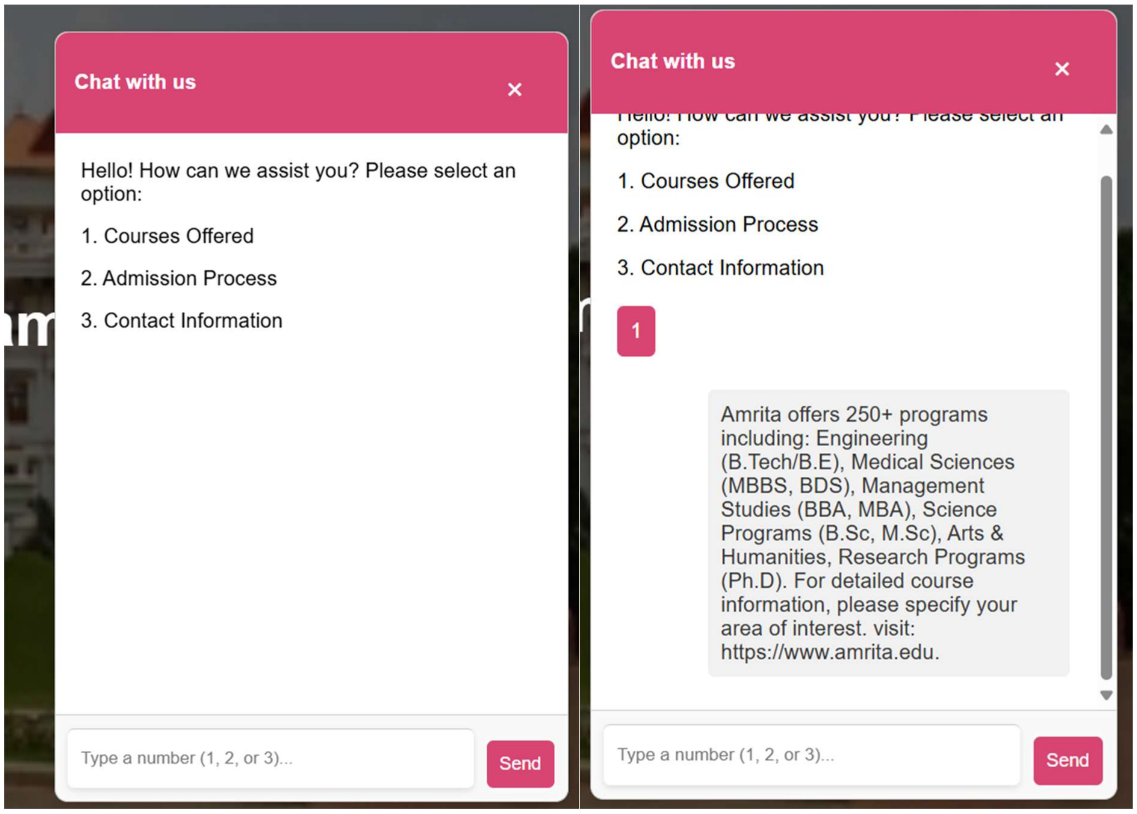
Main Section

- **Background Image:** Features the campus building, creating a professional appearance.

- **Title:** "Amrita Vishwa Vidyapeetham" in bold white text.
- **Subtitle:** "Education for Life, Education for Living" below the title.
- **Button:** A pink button labeled **"Visit Us To Know More"**, prompting user engagement.

Footer

- **Chat Icon (Bottom Right):** Suggests chatbot functionality for queries.


Chat with us

Hello! How can we assist you? Please select an option:

1. Courses Offered

2. Admission Process

3. Contact Information

Type a number (1, 2, or 3)...

Send

Chat with us

Hello! How can we assist you? Please select an option:

1. Courses Offered

2. Admission Process

3. Contact Information

1

Amrita offers 250+ programs including: Engineering (B.Tech/B.E), Medical Sciences (MBBS, BDS), Management Studies (BBA, MBA), Science Programs (B.Sc, M.Sc), Arts & Humanities, Research Programs (Ph.D). For detailed course information, please specify your area of interest. visit: https://www.amrita.edu.

Type a number (1, 2, or 3)...

Send

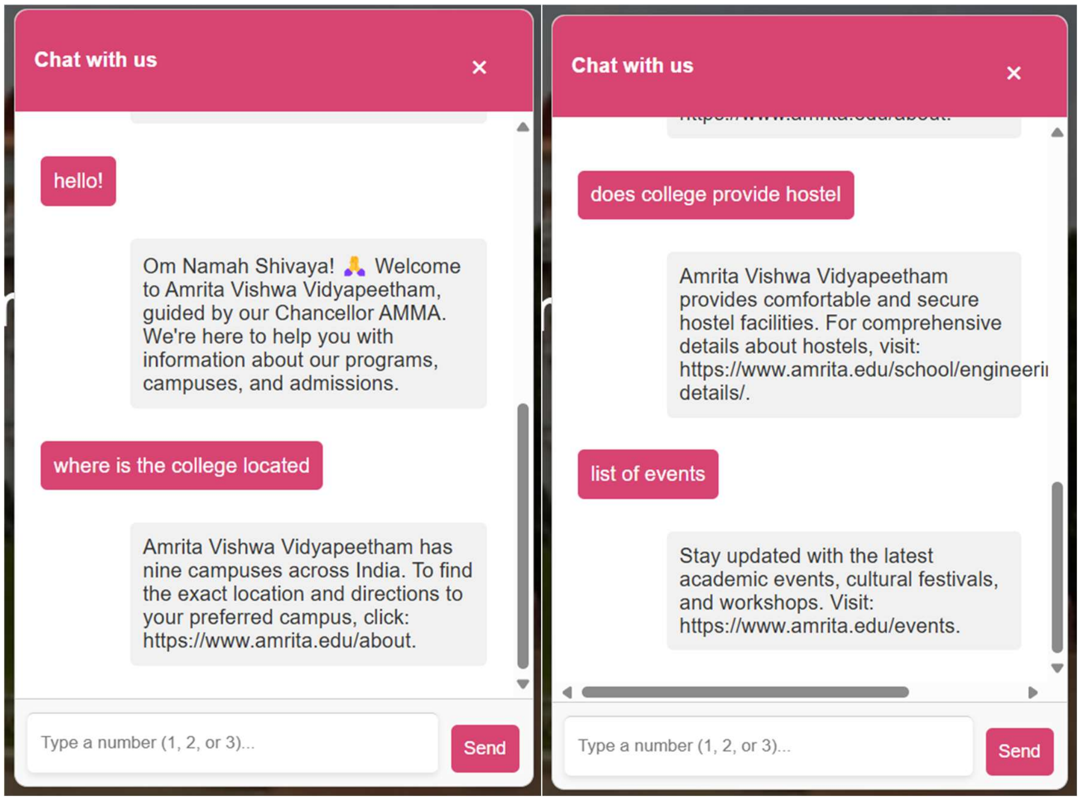
This is a **chat window layout** for a chatbot interface and included Menu Driven feature. Below is a breakdown of the elements:

Components:

1. **Header**:
   - Title: "Chat with us".
   - Close Button (X): Exits/minimizes the chat.
2. **Body**:
   - Greeting message: Offers assistance.

- Menu:
    1. Courses Offered.
    2. Admission Process.
    3. Contact Information.

3. **Input Section**:
    - **Input Box**: Users type numbers (1, 2, or 3).
    - **Send Button**: Submits the query.



hello!

Om Namah Shivaya! 🙏 Welcome to Amrita Vishwa Vidyapeetham, guided by our Chancellor AMMA. We're here to help you with information about our programs, campuses, and admissions.

where is the college located

Amrita Vishwa Vidyapeetham has nine campuses across India. To find the exact location and directions to your preferred campus, click: https://www.amrita.edu/about.

does college provide hostel

Amrita Vishwa Vidyapeetham provides comfortable and secure hostel facilities. For comprehensive details about hostels, visit: https://www.amrita.edu/school/engineeri details/.

list of events

Stay updated with the latest academic events, cultural festivals, and workshops. Visit: https://www.amrita.edu/events.

This dataset is structured for training a chatbot to respond to specific queries related to **Amrita Vishwa Vidyapeetham**. It uses three key components for intent recognition:

Structure

1. **Tag**:
    - Represents the **intent** or topic of the query (e.g., greeting, goodbye, hours, etc.).

    - Helps categorize user inputs into predefined topics for better response accuracy.
2. **Patterns**:
    - A list of possible **user inputs** (phrases or keywords) that indicate the corresponding intent.
    - Includes variations of user queries to handle natural language diversity (e.g., for greeting, phrases like "Hi there", "Hello!").
3. **Responses**:
    - Predefined **replies** for each tag that the chatbot provides when a user query matches the intent.
    - Tailored to provide relevant information (e.g., operational hours, courses, contact numbers).

Purpose

- **Intent Matching**: Using patterns, the chatbot identifies the user's intent.
- **Dynamic Responses**: Provides informative and accurate replies based on the detected intent.

Applications

- **Admissions Assistance**: Responds to queries about courses, fees, and contact details.
- **General Queries**: Handles greetings, farewells, and operational information.
- **Event Promotion**: Directs users to information about events and facilities.

Features

- Dynamic Intent Recognition:
  The chatbot intelligently maps user inputs to predefined intents.
- Natural Responses:
  Predefined responses ensure clarity and relevance.
- Web-Based Deployment:
  Users can interact with the chatbot through a simple, responsive web interface.
- Fallback Mechanism:
  Handles unrecognized queries gracefully, improving user experience.

Challenges and Resolutions

1. Flask Integration with the ML Model:
   - Challenge: Loading the trained model into the Flask application.
   - Solution: Ensured proper directory structure and used TensorFlow's load_model function.
2. Dataset Limitations:
   - Challenge: Limited diversity in patterns for each intent.
   - Solution: Expanded the dataset with additional patterns and variations for better training.

Future Enhancements

1. Scalable Dataset:
   - Increase the number of intents and patterns to handle a broader range of queries.
2. Advanced Models:
   - Upgrade to transformer models like BERT for improved intent classification.

3. Cloud Deployment:
    - Host the chatbot on platforms like AWS, Azure, or Google Cloud for wider accessibility.
4. Multilingual Support:
    - Enable the chatbot to handle queries in multiple languages using translation APIs.

Evaluation Metrics

Functionality:

- Accurate classification of user intents.
- Responsive to variations in queries.

Code Quality:

- Modular design for easy maintenance and updates.
- Clean, well-documented code for clarity.

User Experience:

- Intuitive interface with real-time query handling.
- Meaningful fallback responses for unrecognized queries.

Error Handling:

- Robust mechanisms to manage unexpected inputs and application errors.

Conclusion

This project demonstrates the effective use of machine learning and NLP for chatbot development. By integrating a trained model, the chatbot delivers precise and context-aware responses, enhancing user experience. The application is well-suited for deployment in educational institutions, providing students and visitors with quick access to essential information.

GitHub Repository:
All project files, including datasets, code, and instructions, are available for reference and deployment.

Git Link: Click Me