NLP-Powered Calendar Scheduling Chatbot

NLP Internship Task at Archle Labs

Arjunaa Seshadri

Mail: krooks522@gmail.com

Phone: 9999144193

Objective:

Develop a chatbot in Python that can intelligently understand natural language messages involving event planning, extract relevant scheduling information, and integrate with the Google Calendar API to create events only after receiving user confirmation. The chatbot should simulate human-like interaction to make scheduling conversational and intuitive.

Dependencies:

- Python
- Hugging Face `transformers`
- PyTorch
- Google API Python Client (`google-api-python-client`)
- Google Authentication Library (`google-auth-oauthlib`)
- python-dateutil
- tzlocal
- python-dotenv
- Hugging Face Hub

Install:

pip install --upgrade transformers torch google-api-python-client google-auth-oauthlib python-dateutil tzlocal python-dotenv huggingface_hub sentencepiece accelerate

Intro:

This project implements a text-based chatbot in Python designed to simplify event scheduling through natural language interaction. It takes conversational input from the user via a command-line interface. This input is processed using the Mistral 7B Instruct Large Language Model to understand the scheduling intent and extract key details like the event title, date, start time, and end time. The system is designed to handle various phrasing and date/time formats. After extraction and parsing, the chatbot presents the interpreted details to the user for confirmation before proceeding to interact with the Google Calendar API to create the event. The codebase is organized into distinct modules for better structure and maintainability, with configuration handled via a `.env` file.

Functions:

The project's functionality is divided across several modules:

Configuration Loading (`config.py`): Loads necessary configurations like API keys, file paths, and the LLM model name from a `.env` file, keeping sensitive data separate from the code.

LLM Interaction (`llm_util.py`):

Initializes the Hugging Face `transformers` pipeline for the specified LLM (Mistral 7B Instruct).

Handles authentication with Hugging Face Hub for accessing gated models (using tokens).

Constructs carefully engineered prompts, including specific instruction formats (`<s>[INST]...[/INST]`), to guide the LLM.

Sends the user's request within the prompt to the LLM for processing.

Parses the LLM's text response to extract the structured JSON containing event details (Title, Date, Start Time, End Time).

Date/Time Parsing (`dt.py`):

Takes the raw date and time strings extracted by the LLM.

Uses `python-dateutil` to parse these strings into standard `datetime` objects, handling relative terms (e.g., "tomorrow", "next Friday") and various formats (e.g., "2 PM", "14:30").

Applies a default event duration (e.g., 1 hour) if an end time is not explicitly provided.

Uses `tzlocal` to make the `datetime` objects time zone-aware.

Google Calendar Utilities (`calendar_conn.py`):

Manages the OAuth 2.0 authentication flow with Google, handling user consent and token storage/refresh (`credentials.json`, `token.json`).

Builds the Google Calendar API service client.

Formats the parsed event details (title, start/end datetimes) into the structure required by the Google Calendar API.

Executes the API call (`events().insert()`) to create the event in the user's primary Google Calendar.

Main Chatbot Loop (`main.py`):

Provides the command-line interface for user interaction.

Orchestrates the workflow: receives input, calls the LLM utility, calls the date/time parser.

Presents the parsed event details clearly to the user for confirmation.

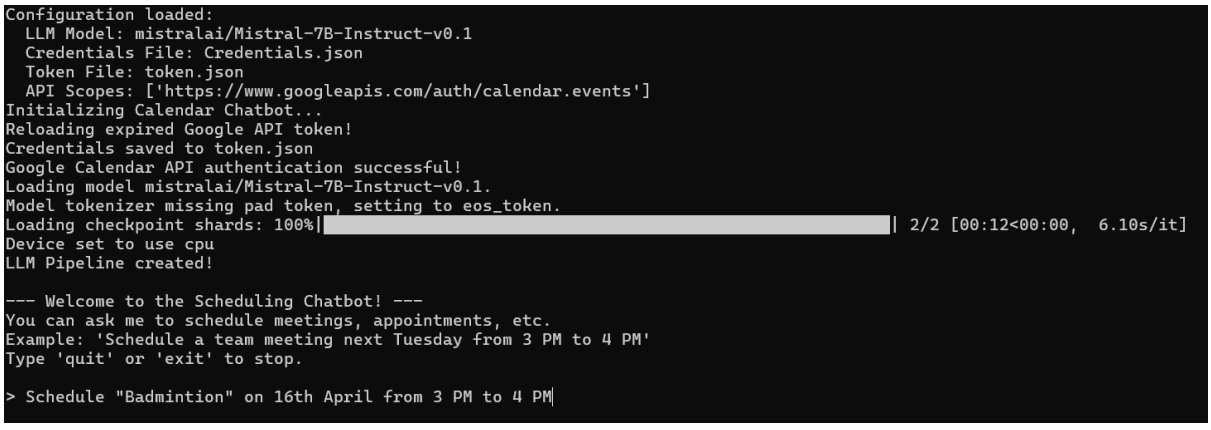
Handles the user's "yes" or "no" response.

Triggers the Google Calendar event creation function upon confirmation.

Includes basic error handling and user feedback messages.

Output:

The primary successful output of the chatbot is the creation of an event in the user's Google Calendar, corresponding to the details extracted from their natural language request and confirmed by them. This is accompanied by a confirmation message printed in the terminal, often including a direct link to the newly created calendar event.



```
Parsed & Normalized LLM Data: {'title': 'Badmintion', 'date': '16th April', 'start_time': '3 PM', 'end_time': None}
Parsed start datetime: 2025-04-16 15:00:00
End time not specified by LLM.
Parsed end datetime: 2025-04-16 16:00:00

Okay, I understood the following event details:
- Title: Badmintion
- Start: Wednesday, April 16, 2025 at 03:00 PM (Asia/Calcutta)
- End:   Wednesday, April 16, 2025 at 04:00 PM (Asia/Calcutta)

Shall I schedule this event in your Google Calendar? (yes/no): yes
```

Conclusion:

This project successfully delivers an NLP-powered chatbot that simplifies Google Calendar scheduling. By integrating the Mistral 7B Instruct LLM for advanced natural language understanding and entity extraction, and connecting with the Google Calendar API, the chatbot effectively translates conversational requests into concrete calendar events. It meets the core project objectives, demonstrating practical skills in applying modern NLP models, managing APIs, handling authentication, and building a functional Python application.