Concepts Behind the Local Chatbot Project

This document explains the key concepts behind the chatbot project. It is designed as background knowledge for preparing a demo video or presentation.

1. Project Goal

The project builds a local chatbot using Hugging Face's `distilgpt2` model. It runs in the command line, maintains short-term memory with a sliding window, and exits cleanly when the user types `/exit`.

2. Sliding-Window Memory

The chatbot needs context from previous turns. Instead of keeping the whole history, we keep the last N turns (User+Bot). This is managed by the `ChatMemory` class. For example: If the user asks about France, and then 'What about Italy?', the bot remembers the France question in memory.

3. Pipeline and Model Loading

We use Hugging Face's `pipeline('text-generation')` to load both the model and tokenizer. The tokenizer splits/join text, and the model generates replies. If no pad token exists, we set it equal to the EOS token.

4. Prompt Engineering

The model generates text by continuing a prompt. Our prompt includes: a system preamble ('You are a helpful assistant...'), recent conversation turns, and the new user input followed by 'Bot:'. The model then completes the response.

5. Stopping and Cleaning Outputs

To avoid the bot generating unnecessary text, we stop output when sequences like '\nUser:' or '\nBot:' appear. This ensures clean replies.

6. CLI Interface

The chatbot runs in a loop: (1) take user input, (2) check if '/exit', (3) add user input to memory, (4) build a prompt, (5) generate a reply, (6) display the reply, (7) update memory.

7. Kaggle Integration

In Kaggle: (1) Enable Internet to download models, (2) install dependencies with `!pip install transformers accelerate`, (3) place files in `/kaggle/working/chatbot_cli`, (4) run `python interface.py` to start chatting interactively.

8. Example Interaction

User: What is the capital of France?
Bot: The capital of France is Paris.

User: And what about Italy?
Bot: The capital of Italy is Rome.

User: /exit
Bot: Exiting chatbot. Goodbye!

Conclusion

This chatbot shows how to integrate ML models into real applications. The main learning outcomes are: modular design, memory handling, prompt engineering, and practical deployment in Kaggle.