NLP

Section 4

LLM APIs

- Multiple APIs are available (OpenAI, Gemini, Claude, Deepseek, .. etc)
- Gemini is free (rate limited)
- APIs are directly connected to commercial LLMs (mostly closed-source)
- These closed source LLMs are typically state-of-the-art and may outperform the model you trained for a specific task.

Gemini API

- You can access a free rate-limited API on aistudio.google.com
- It will require a google cloud project setup
- Typically used with google-genai library

```python
from google import genai

# The client gets the API key from the environment variable `GEMINI_API_KEY`.
client = genai.Client()

response = client.models.generate_content(
    model="gemini-2.5-flash", contents="Explain how AI works in a few words"
)
print(response.text)
```

LangChain

- An open-source framework for managing LLMs with their APIs

- Connects LLMs to external data, tools, and agentic features.

LangChain Core Components

- LLMs & Chat Models – Interface with models like GPT, Gemini, Claude, etc.

- Prompts – Templates that structure how models receive inputs.

- Chains – Combine multiple components to create logical pipelines.

- Memory – Store and use conversation history for contextual responses.

- Tools & Agents – Allow LLMs to take actions (e.g., search, calculate, query DB).

LangChain Adv.

- Unified interface to work with different LLM providers (e.g., OpenAI, Gemini).

- Easily switch models without changing your logic.

- Templates structure how data is passed to LLMs which ensures consistency and reusability

- Can link multiple components into a single workflow.

LangChain Adv. Continued

- Multiple memory types (buffer, summary, vector).

- Tools give LLMs the ability to act (e.g., search, calculate, query DB) and the

  agent can choose which tool to use

- Enable autonomous task execution.

Code

- Call APIs using LangChain

- Use Prompts and Chains

- Use LCEL to enable tools usage

- Use LangGraph to enable more complicated tools and workflows

- Memory example