Giới thiệu về Khóa học LLM Engineering (8 Tuần)

Khóa học LLM Engineering là chương trình đào tạo chuyên sâu trong 8 tuần, giúp người học hiểu rõ và áp dụng các kỹ thuật về mô hình ngôn ngữ lớn (LLM). Ngay từ buổi đầu tiên, học viên sẽ được thực hành trước khi bước vào giới thiệu khóa học và giảng viên.

Thực hành đầu tiên: Cài đặt và sử dụng Ollama

- Hướng dẫn cài đặt Ollama trên Windows và MacBook.
- Mở PowerShell (Windows) hoặc Terminal (Mac) và chạy lệnh:

  ollama run llama3.2
- Chạy các ví dụ bằng tiếng Tây Ban Nha: Hola. ¿Cómo estás?

Giới thiệu về chương trình 8 tuần

- **Tuần 1:** Khám phá các mô hình tiên phong.
- **Tuần 2:** Xây dựng giao diện người dùng với Gradio.
- **Tuần 3:** Làm việc với mã nguồn mở tại Hugging Face.
- **Tuần 4:** Lựa chọn mô hình AI phù hợp.
- **Tuần 5:** RAG (Retrieval-Augmented Generation).
- **Tuần 6:** Dự án thực tế - Phân tích dữ liệu và tạo mô hình cơ sở.
- **Tuần 7:** Cải tiến mô hình mã nguồn mở.
- **Tuần 8:** Xây dựng hệ thống AI tự động (Agentic AI)

Hướng dẫn phương pháp học:

Phương pháp học: Khuyến khích sử dụng GitHub để tạo Pull Request và chia sẻ giải pháp.

Giới thiệu dự án:

- Dự án Chatbot trợ lý hãng hàng không.
- Dự án hệ thống RAG.
- Dự án AI tác nhân tự động.

Giới thiệu giảng viên

- Kinh nghiệm làm việc và khởi nghiệp trong AI, LLM.
- Chia sẻ câu chuyện về chặng đường phát triển sự nghiệp.

=> Khẳng định chuyên môn về AI và LLM.

Lộ trình Tuần 1

Ngày 1: Tạo giải pháp LLM đầu tiên từ đầu.
Ngày 2: Định vị tốt nhất để thành công trong 8 tuần.
Ngày 3: So sánh các mô hình Frontier tiên tiến nhất.
Ngày 4: Tìm hiểu về Transformers.
Ngày 5: Xây dựng giải pháp mạnh mẽ trong vài phút.

Hướng dẫn cài đặt môi trường

1. Clone repo từ GitHub và thiết lập môi trường theo Readme.
2. Tạo môi trường ảo:

conda env create -f environment.yml # Anaconda

python -m venv venv # Virtualenv

3. Kích hoạt môi trường:

Mac: source venv/bin/activate

Windows: venv\Scripts\activate

4. Thiết lập API Key OpenAI, tạo file .env chứa khoá API.
5. Cài đặt JupyterLab và chạy môi trường.

Giới thiệu về GitHub của khóa học

- GitHub của khóa học là nơi chứa toàn bộ mã nguồn (source code) phục vụ cho quá trình học tập.
- Hướng dẫn cách tải mã nguồn từ GitHub về máy tính.
- Cài đặt các công cụ cần thiết như Anaconda để thiết lập môi trường làm việc.

Hướng dẫn cài đặt Git và tải mã nguồn về máy

Cài đặt Git trên Windows

- Truy cập trang "LM Engineering" (đã cung cấp đường dẫn) để tải Git và cài đặt.
- Sau khi cài đặt xong, mở PowerShell và kiểm tra bằng lệnh:

```
git --version
```

Tải mã nguồn từ GitHub về máy

- Mở PowerShell hoặc Terminal, tạo thư mục chứa mã nguồn:

```
mkdir MyProject && cd MyProject
```

- Clone repository về máy:

```
git clone <URL_GITHUB_REPO>
cd <Tên_Repo>
```

Hướng dẫn cài đặt Anaconda và thiết lập môi trường

Cài đặt Anaconda

Truy cập trang "LM Engineering" để tải Anaconda từ nguồn chính thức.

Cài đặt Anaconda theo hướng dẫn.

Thiết lập môi trường bằng Anaconda

Mở **PowerShell** và chạy lệnh để tạo môi trường:

```
conda env create -f environment.yml
```

Kích hoạt môi trường:

```
conda activate <Tên_Môi_Trường>
```

Thiết lập API Key OpenAI và kết nối với JupyterLab

Tạo và bảo quản API Key

- Truy cập trang OpenAI để lấy API Key.
- Tạo file .env để lưu trữ API Key:

```
echo "OPENAI_API_KEY=your_key_here" > .env
```

Cài đặt và khởi động JupyterLab

Kích hoạt môi trường và chạy JupyterLab:

jupyter lab

File Troubleshooting – Hỗ trợ khắc phục lỗi

Mục đích

- Hỗ trợ người học khi gặp lỗi trong quá trình cài đặt và chạy dự án.

Cách sử dụng

- Nếu lỗi import thư viện, kiểm tra xem môi trường Conda đã kích hoạt chưa.
- Nếu lỗi API Key không hợp lệ, mở file .env để kiểm tra giá trị.
- Nếu không tự khắc phục được lỗi, có thể liên hệ giảng viên qua email hoặc LinkedIn.

Giới thiệu dự án "Instant Gratification"

Mục tiêu: **Tích hợp API OpenAI để xử lý dữ liệu từ website.**

Bắt đầu triển khai dự án:

- Viết class Website để thu thập nội dung trang web.
- Sử dụng API OpenAI để tóm tắt nội dung.
- Lưu ý: Nếu gặp lỗi, kiểm tra troubleshooting file hoặc restart Kernel.

Import các thư viện cần thiết

```
import os
import requests
from dotenv import load_dotenv
from bs4 import BeautifulSoup
from IPython.display import Markdown, display
from openai import OpenAI
```

Lưu ý: Nếu gặp lỗi khi chạy đoạn code này, hãy xem notebook Troubleshooting để khắc phục.

Mục đích của đoạn code này

Đoạn code trên chuẩn bị các công cụ cần thiết để:

Quản lý biến môi trường: dotenv, os

Gửi và xử lý yêu cầu HTTP: requests, BeautifulSoup

Hiển thị nội dung Markdown: IPython.display

Làm việc với OpenAI API: openai

Tải API Key từ file .env

```
# Load environment variables from .env file
load_dotenv(override=True)
api_key = os.getenv('OPENAI_API_KEY')
# Kiểm tra API Keyif not api_key:
    print("❌Không tìm thấy API key! Hãy xem notebook Troubleshooting để kiểm tra & khắc phục.")elif not api_key.startswith("sk-proj-"):
    print("❌ API key có vẻ không đúng định dạng (phải bắt đầu bằng 'sk-proj-'). Kiểm tra lại notebook Troubleshooting.")elif api_key.strip() != api_key:
```

print("❌ API key có dấu cách thừa ở đầu hoặc cuối. Hãy xóa chúng và thử lại.")else:

print("✅API key hợp lệ và sẵn sàng sử dụng!")

Mục đích của đoạn code này

Đọc API key từ file .env.

Kiểm tra **lỗi thường gặp:**

- API key chưa thiết lập
- API key sai định dạng
- API key có dấu cách thừa

Nếu hợp lệ, thông báo rằng API key đã sẵn sàng sử dụng.

Khởi tạo OpenAI API

Khởi tạo API OpenAI

openai = OpenAI()

- **Lưu ý**: Nếu gặp lỗi khi khởi tạo API, hãy thử:
- Kernel menu > Restart Kernel and Clear Outputs, sau đó chạy lại từ đầu.
- Nếu vẫn không hoạt động, hãy xem notebook Troubleshooting để khắc phục.

Gửi tin nhắn đến GPT-4o-mini

message = "Hello, GPT! This is my first ever message to you! Hi!"

response = openai.chat.completions.create(

model="gpt-4o-mini",

messages=[{"role": "user", "content": message}]

)print(response.choices[0].message.content)

Mục đích của đoạn code này

- Gửi tin nhắn đến mô hình GPT-4o-mini thông qua OpenAI API.
- Nhận phản hồi từ AI và in ra màn hình.

Xây dựng lớp Website để lấy nội dung trang web

 # Một số trang web yêu cầu sử dụng headers hợp lệ khi gửi request

headers = {

"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"

}

class Website:

def __init__(self, url):

"""

Khởi tạo đối tượng Website từ URL được cung cấp.

Sử dụng BeautifulSoup để phân tích nội dung trang web.

"""

self.url = url

response = requests.get(url, headers=headers)

```
        soup = BeautifulSoup(response.content, 'html.parser')

        # Lấy tiêu đề trang web
        self.title = soup.title.string if soup.title else "No title found"

        # Xóa các phần không cần thiết (script, style, img, input)
        for irrelevant in soup.body(["script", "style", "img", "input"]):
            irrelevant.decompose()

        # Lấy nội dung văn bản từ trang web
        self.text = soup.body.get_text(separator="\n", strip=True)
```

Mục đích của lớp Website

Khi tạo một đối tượng Website("https://example.com"), lớp này sẽ:

- Gửi yêu cầu đến https://example.com.
- Phân tích HTML trang web bằng BeautifulSoup.
- Lấy tiêu đề trang và lưu vào self.title.
- Loại bỏ các phần không liên quan như script, style, hình ảnh.
- Trích xuất nội dung văn bản và lưu vào self.text.

Thử nghiệm lấy dữ liệu từ trang web

```
# Tạo đối tượng Website và lấy nội dung trang web
ed = Website("https://edwarddonner.com")
# In tiêu đề và nội dung văn bản thu thập đượcprint(ed.title)print(ed.text)
```

Mục đích của đoạn code này

- Tạo một đối tượng Website để thu thập dữ liệu từ https://edwarddonner.com.
- In ra tiêu đề trang và nội dung văn bản.

Giới thiệu về System Prompt và User Prompt

- **System Prompt:** Thiết lập bối cảnh và quy tắc phản hồi cho AI.
- **User Prompt:** Nội dung nhập vào từ người dùng để yêu cầu phản hồi.

Cấu trúc tin nhắn gửi đến OpenAI: Sử dụng danh sách Python chứa các từ điển với hai trường:

- **role: Vai trò (system, user).**
- **content: Nội dung tin nhắn.**

Viết hàm tạo System Prompt và User Prompt từ nội dung website

1. System Prompt (system_prompt)

```
system_prompt = """Bạn là một trợ lý AI chuyên phân tích nội dung trang web và tạo bản tóm tắt ngắn gọn.
Vui lòng bỏ qua các nội dung điều hướng (navigation).
Định dạng phản hồi bằng Markdown để dễ đọc."""
```

Mục đích

- Xác định vai trò: Trợ lý AI phân tích nội dung trang web.
- Bỏ qua nội dung không quan trọng (menu, điều hướng).
- Phản hồi dưới dạng Markdown để có cấu trúc rõ ràng.

2. Hàm user_prompt_for(website)

def user_prompt_for(website):

return f"""

Tiêu đề trang web: {website.title}

Nội dung chính:{website.text}

Mục đich

Nếu trang web chứa tin tức hoặc thông báo, vui lòng tóm tắt những thông tin quan trọng.
"""

- Lấy tiêu đề trang web (website.title).
- Thêm nội dung chính (website.text).
- Nếu là trang tin tức, yêu cầu AI tóm tắt những thông tin quan trọng.

Viết hàm tạo danh sách tin nhắn gửi đến OpenAI

def messages_for(website):

return [

{"role": "system", "content": system_prompt},

{"role": "user", "content": user_prompt_for(website)}

]

Trả về danh sách tin nhắn:

- **System Prompt** để thiết lập AI.
- **User Prompt** chứa nội dung cần tóm tắt.

Ví dụ sử dụng:

messages_for(ed)

Kết quả: Danh sách tin nhắn gửi đến OpenAI API.

```
[{'role': 'system',
  'content': 'You are an assistant that analyzes the contents of a website and provides a short summary, ignoring text that might be navigation relate
d. Respond in markdown.'},
 {'role': 'user',
  'content': 'You are looking at a website titled Home - Edward Donner\nThe contents of this website is as follows; please provide a short summary of t
his website in markdown. If it includes news or announcements, then summarize these too.\n\nHome\nConnect Four\nOutsmart\nAn arena that pits LLMs again
st each other in a battle of diplomacy and deviousness\nAbout\nPosts\nWell, hi there.\nI'm Ed. I like writing code and experimenting with LLMs, and hop
efully you're here because you do too. I also enjoy DJing (but I'm badly out of practice), amateur electronic music production (\nvery\namateur) and lo
sing myself in\nHacker News\n, nodding my head sagely to things I only half understand.\nI'm the co-founder and CTO of\nNebula.io\n. We're applying AI
to a field where it can make a massive, positive impact: helping people discover their potential and pursue their reason for being. Recruiters use our
product today to source, understand, engage and manage talent. I'm previously the founder and CEO of AI startup untapt,\nacquired in 2021\n.\nWe work w
ith groundbreaking, proprietary LLMs verticalized for talent, we've\npatented\nour matching model, and our award-winning platform has happy customers a
nd tons of press coverage.\nConnect\nwith me for more!\nJanuary 23, 2025\nLLM Workshop - Hands-on with Agents - resources\nDecember 21, 2024\nWelcome,
SuperDataScientists!\nNovember 13, 2024\nMastering AI and LLM Engineering - Resources\nOctober 16, 2024\nFrom Software Engineer to AI Data Scientist -
resources\nNavigation\nHome\nConnect Four\nOutsmart\nAn arena that pits LLMs against each other in a battle of diplomacy and deviousness\nAbout\nPosts
\nGet in touch\ned [at] edwarddonner [dot] com\nwww.edwarddonner.com\nFollow me\nLinkedIn\nTwitter\nFacebook\nSubscribe to newsletter\nType your email…
\nSubscribe'}]
```

Hàm gọi OpenAI API để tóm tắt trang web

def summarize(url):

website = Website(url)

response = openai.chat.completions.create(

model="gpt-4o-mini",

```
        messages=messages_for(website)
    )
    return response.choices[0].message.content
```

Mục đích:

- Tạo đối tượng Website để thu thập nội dung trang.
- Gửi yêu cầu đến OpenAI API với model "gpt-4o-mini".
- Nhận phản hồi và trả về kết quả tóm tắt.

Ví dụ sử dụng:

summarize("https://edwarddonner.com")

Kết quả: Một bản tóm tắt ngắn gọn về nội dung trang web.


```
'# Summary of Edward Donner\'s Website\n\nThe website is operated by Ed Donner, who has a background in coding and experimentation with
large language models (LLMs). He is the co-founder and CTO of Nebula.io, which utilizes AI to enhance talent discovery and management fo
r recruiters. Ed has previously founded another AI startup, untapt, which was acquired in 2021. \n\n## Key Features:\n- **Outsmart**: An
arena for LLMs focused on diplomacy and strategic interactions.\n- **Personal Interests**: Ed is interested in DJing, electronic music p
roduction, and engaging with the tech community through platforms like Hacker News.\n\n## Recent Posts:\n- **October 16, 2024**: "From S
oftware Engineer to AI Data Scientist - resources"\n- **August 6, 2024**: "Outsmart LLM Arena - a battle of diplomacy and deviousness"\n
- **June 26, 2024**: "Choosing the Right LLM: Toolkit and Resources"\n- **February 7, 2024**: "Fine-tuning an LLM on your texts: a simul
```


Kết luận:

Tóm tắt các điểm chính đã học trong ngày đầu tiên:

- Sử dụng Ollama để chạy LLM cục bộ trên máy tính của bạn.
- Viết code để gọi các mô hình tiên tiến của OpenAI.
- Phân biệt giữa lời nhắc Hệ thống (System prompt) và lời nhắc Người dùng (User prompt).
- Tóm tắt - áp dụng cho nhiều vấn đề thương mại.

Những gì bạn sẽ có thể làm vào ngày mai:

- Biết chính xác các bước để đạt được sự thành thạo LLM.
- Được thiết lập để thành công.
- Nhận biết các Mô hình Tiên phong hàng đầu và cách sử dụng chúng.