

Lab 4: Tìm hiểu và trải nghiệm về Prompt Engineering

Phần này sẽ được hướng dẫn tại lớp lý thuyết: 8/3/2025

Yêu cầu: TỰ CODE các bài tập này (sẽ có bài kiểm tra trên giấy tại lớp)

Quy tắc nộp bài:

- Nén thành file MSSV.zip chứa source code
- Mỗi kỹ thuật làm trên 1 file py/ipynb riêng biệt
- Đặt tên file theo format: MSSV_[tên kỹ thuật].ipynb/py
Ví dụ: 123456_fewShot.ipynb, 123456_zeroShot.py

Deadline: 12/3/2025 10:00 PM

I. Giới thiệu

Prompt engineering là nghệ thuật và khoa học về việc thiết kế các câu lệnh (prompts) hiệu quả để tương tác với các mô hình ngôn ngữ lớn (LLMs) như GPT và các mô hình AI khác. Lab này sẽ giúp bạn hiểu các nguyên tắc cơ bản của prompt engineering và thực hành từ việc thiết lập môi trường đến tương tác với mô hình OpenAI trong Visual Studio Code.

Tạo tài khoản OpenAI và lấy API Key

1. Truy cập trang web [OpenAI Platform](https://platform.openai.com/)
2. Đăng ký tài khoản mới hoặc đăng nhập với tài khoản hiện có
3. Sau khi đăng nhập, vào mục **API Keys** (<https://platform.openai.com/api-keys>)
4. Nhấp vào **Create new secret key**
5. Đặt tên cho key (ví dụ: "Prompt Engineering Lab") và nhấp vào **Create secret key**
6. **Quan trọng:** Sao chép và lưu API key vào một nơi an toàn ngay lập tức, vì bạn sẽ không thể xem lại key này sau khi đóng cửa sổ

Thiết lập trong Visual Studio Code

- ☐ Mở Visual Studio Code
- ☐ Tạo một thư mục mới cho dự án "TenDuAn"
- ☐ Mở thư mục vừa tạo trong VS Code
- ☐ Tạo file môi trường .env để lưu API key:

```
OPENAI_API_KEY=your_api_key_here
```
- ☐ Tạo file requirements.txt với nội dung:

```
openai>=1.0.0  
python-dotenv>=0.19.0
```
- ☐ Mở terminal trong VS Code (Terminal > New Terminal):

```
pip install -r requirements.txt
```

The screenshot shows a VS Code editor with a file named `requirements.txt` containing the following content:

```
1 openai>=1.0.0
2 python-dotenv>=0.19.0
3
```

The terminal output shows the execution of `python -m venv venv` and `pip install -r requirements.txt`. The output includes the following information:

- Collecting openai>=1.0.0 (from -r requirements.txt (line 1))
- Obtaining dependency information for openai>=1.0.0 from https://files.pythonhosted.org/packages/ba/db/7bab832be24631a793492c1c61ecbf029018b99696f435db3b63d690bf1c/openai-1.65.4-py3-none-any.whl.metadata (27 kB)
- Downloading openai-1.65.4-py3-none-any.whl.metadata (27 kB)
- Collecting python-dotenv>=0.19.0 (from -r requirements.txt (line 2))
- Obtaining dependency information for python-dotenv>=0.19.0 from https://files.pythonhosted.org/packages/6a/3e/b68c118422ec867fa7ab88444e1274aa40681c606d59ac27de5a5588f082/python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
- Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
- Collecting anyio<5,>=3.5.0 (from openai>=1.0.0->-r requirements.txt (line 1))
- Obtaining dependency information for anyio<5,>=3.5.0 from https://files.pythonhosted.org/packages/46/eb/e7f063ad1fec6b3178a3cd82d1a3c4de82ccc283fc42746168188e1cdd5/anyio-4.8.0-py3-none-any.whl.metadata (4.6 kB)
- Downloading anyio-4.8.0-py3-none-any.whl.metadata (4.6 kB)
- Collecting distro<2,>=1.7.0 (from openai>=1.0.0->-r requirements.txt (line 1))
- Obtaining dependency information for distro<2,>=1.7.0 from https://files.pythonhosted.org/packages/12/b3/231ffd4ab1fc9d679809f356cebee130ac7daa00d6d6f3206dd4fd137e9e/distro-1.9.0-py3-none-any.whl.metadata (6.8 kB)
- Downloading distro-1.9.0-py3-none-any.whl.metadata (6.8 kB)
- Collecting httpx<1,>=0.23.0 (from openai>=1.0.0->-r requirements.txt (line 1))
- Obtaining dependency information for httpx<1,>=0.23.0 from https://files.pythonhosted.org/packages/2a/39/e50c7c3a983047577ee07d2a9e53faf5a69493943ec3f6a384bdc792deb2/httpx-0.28.1-py3-none-any.whl.metadata (7.1 kB)
- Downloading httpx-0.28.1-py3-none-any.whl.metadata (7.1 kB)

II. Thực hành

Phần 1: Cơ bản về tương tác với OpenAI API

Bài tập 1.1: Thiết lập và kiểm tra kết nối API

The screenshot shows a VS Code editor with a file named `test_connection.py` containing the following content:

```
1 import os
2 from openai import OpenAI
3 from dotenv import load_dotenv
4
5 # Load API key from file .env
6 load_dotenv()
7
8 # Initialize client
9 client = OpenAI(api_key=os.environ.get("OPENAI_API_KEY"))
10
11 def test():
12     try:
13         response = client.chat.completions.create(
14             model="gpt-3.5-turbo",
15             messages=[
16                 {"role": "system", "content": "You are a helpful AI assistant."},
17                 {"role": "user", "content": "Hello, I'm testing the API connection!"}
18             ]
19         )
20         print("Connection successful!")
21         print("Response:", response.choices[0].message.content)
22     except Exception as e:
23         print("An error occurred:", e)
24
25 test()
26
```

The terminal output shows the execution of `python test_connection.py`. The output includes the following information:

- Connection successful!
- Response: That's great! If you need any assistance with testing or connecting to an API, feel free to ask.

Bài tập 1.2: Tạo hàm gửi prompt cơ bản

```

prompt_utils.py > ...
[Codeium] Explain the function: send_prompt

2  from openai import OpenAI
3  from dotenv import load_dotenv
4
5  load_dotenv()
6
7  client = OpenAI(api_key=os.environ.get("OPENAI_API_KEY"))
8
9  def send_prompt(prompt, system_message="You are a helpful AI assistant.", model="gpt-4"):
10
11     """
12     Send a prompt to the OpenAI API and return a response
13
14     Args:
15     prompt (str): Prompt content
16     system_message (str): System message (default: "You are a helpful AI assistant.")
17     model (str): AI model to use (default: "gpt-3.5-turbo")
18
19     Returns:
20     str: Response from AI
21
22     """
23     response = client.chat.completions.create(
24         model=model,
25         messages=[
26             {"role": "system", "content": system_message},
27             {"role": "user", "content": prompt}
28         ]
29     )
30     return response.choices[0].message.content
31
32 if __name__ == "__main__":
33     test_prompt = "Giải thích prompt engineering bằng 1 câu."
34     result = send_prompt(test_prompt)
35     print(result)

```

OUTPUT PROBLEMS SPELL CHECKER 10 COMMENTS TERMINAL DEBUG CONSOLE

• (venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 % python prompt_utils.py
 Prompt engineering là quá trình thiết kế và tinh chỉnh các câu hỏi hoặc lệnh để tối ưu hóa hiệu suất của hệ thống trí tuệ nhân tạo.

○ (venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 %

Phần 2: Thực hành các kỹ thuật prompt engineering

Đọc tài liệu tại: <https://www.promptingguide.ai/> các kỹ thuật mà chúng ta được cung cấp

Prompting Techniques

Zero-shot Prompting

Few-shot Prompting

Chain-of-Thought
Prompting

Ý nghĩa các kỹ thuật này là gì?

Phần trả lời:

Bài tập 2.1: Zero-shot prompting

```

zero_shot.py > ...
1  from prompt_utils import send_prompt
2
3  zero_shot_prompt = """
4  Phân loại đoạn văn sau vào một trong các thể loại: Thể thao, Công nghệ, Giải trí, Chính trị, hoặc Khoa học.
   Đoạn văn: 'Apple vừa công bố iPhone mới với chip A16 và camera 48MP cải tiến.'
5  """
6
7  result = send_prompt(zero_shot_prompt)
8  print("Zero-shot classification result:")
9  print(result)
10
11
12 # Bài tập thực hành:
13 # 1. Viết một zero-shot prompt yêu cầu AI tóm tắt một đoạn văn bản
14 # 2. Viết một zero-shot prompt yêu cầu AI phân tích tình cảm (tích cực/tiêu cực) của một bình luận

```

OUTPUT PROBLEMS SPELL CHECKER 128 COMMENTS TERMINAL DEBUG CONSOLE

• (venv) (3.11.5) macintoshhdd@Macintoshs-MacBook-Pro untitled folder 2 % python zero_shot.py
Zero-shot classification result:
Đoạn văn này thuộc thể loại Công nghệ.

○ (venv) (3.11.5) macintoshhdd@Macintoshs-MacBook-Pro untitled folder 2 %

Bài tập 2.2: Few-shot prompting

```

few_shot.py > ...
1  from prompt_utils import send_prompt
2
3
4  few_shot_prompt = """
5  Phân loại đoạn văn sau vào một trong các thể loại: Thể thao, Công nghệ, Giải trí, Chính trị, hoặc Khoa học.
6
7  Ví dụ 1:
8  Đoạn văn: 'Đội tuyển Việt Nam đã giành chiến thắng 2-1 trước Malaysia trong trận đấu tối qua.'
9  Phân loại: Thể thao
10
11 Ví dụ 2:
12 Đoạn văn: 'NASA phóng tàu vũ trụ mới để khám phá sao Hỏa, dự kiến sẽ đáp xuống bề mặt trong 6 tháng tới.'
13 Phân loại: Khoa học
14
15 Đoạn văn cần phân loại: 'Chính phủ vừa thông qua dự luật mới về cải cách thuế trong phiên họp quốc hội.'
16 Phân loại: """
17
18 result = send_prompt(few_shot_prompt)
19 print("Few-shot classification result:")
20 print(result)
21 print("\n" + "-"*50 + "\n")
22
23 # Bài tập thực hành:
24 # 1. Viết một few-shot prompt để dạy AI cách chuyển đổi câu từ thể bị động sang thể chủ động
25 # 2. Viết một few-shot prompt để dạy AI cách tạo tiêu đề bài báo hấp dẫn từ nội dung

```

OUTPUT PROBLEMS SPELL CHECKER 130 COMMENTS TERMINAL DEBUG CONSOLE

• (venv) (3.11.5) macintoshhdd@Macintoshs-MacBook-Pro untitled folder 2 % python few_shot.py
Few-shot classification result:
Chính trị

Bài tập 2.3: Chain-of-Thought prompting

```
chain_of_thought.py > ...
1  from prompt_utils import send_prompt
2
3
4  cot_prompt = """Giải bài toán sau đây, hãy suy nghĩ từng bước:
5
6  Một cửa hàng bán 5 loại trái cây: táo, cam, nho, chuối và dưa hấu. Giá của mỗi loại lần lượt là 15.000đ, 20.000đ, 35.000đ, 12.000đ và 40.000đ mỗi kg. Nếu mua 2kg táo, 1.5kg cam, 0.5kg nho, 3kg chuối và 4kg dưa hấu, tổng số tiền cần trả là bao nhiêu?"""
7
8  result = send_prompt(cot_prompt)
9  print("Chain-of-Thought result:")
10 print(result)
11 print("\n" + "-"*50 + "\n")
12
13 # Bài tập thực hành:
14 # 1. Viết một chain-of-thought prompt để giải một bài toán xác suất đơn giản
15 # 2. Viết một chain-of-thought prompt để phân tích lợi ích và rủi ro của một quyết định kinh doanh
```

OUTPUT PROBLEMS SPELL CHECKER 123 COMMENTS TERMINAL DEBUG CONSOLE

● (venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 % python chain_of_thought.py

Chain-of-Thought result:

Để giải quyết bài toán này, ta cần nhân giá của mỗi loại trái cây với số kg cần mua, sau đó cộng tất cả các số tiền đó lại.

Cụ thể:

- 2 kg táo: $2\text{kg} \times 15.000\text{đ/kg} = 30.000\text{đ}$
- 1.5 kg cam: $1.5\text{kg} \times 20.000\text{đ/kg} = 30.000\text{đ}$
- 0.5 kg nho: $0.5\text{kg} \times 35.000\text{đ/kg} = 17.500\text{đ}$
- 3 kg chuối: $3\text{kg} \times 12.000\text{đ/kg} = 36.000\text{đ}$
- 4 kg dưa hấu: $4\text{kg} \times 40.000\text{đ/kg} = 160.000\text{đ}$

Bây giờ cộng tất cả lại: $30.000\text{đ} + 30.000\text{đ} + 17.500\text{đ} + 36.000\text{đ} + 160.000\text{đ} = 273.500\text{đ}$

Vậy, tổng số tiền cần trả là 273.500đ.

○ (venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 %

Bài tập 2.4: Role prompting

```

role_prompting.py > ...
1  from prompt_utils import send_prompt
2
3
4  role_prompt = """"Bạn là một chuyên gia dinh dưỡng với hơn 15 năm kinh nghiệm. Hãy đưa ra lời khuyên về chế độ
   ăn uống cho một người muốn tăng cường sức khỏe tim mạch.""
5
6  system_message = "Bạn là một chuyên gia dinh dưỡng với bằng Tiến sĩ Dinh dưỡng học và 15 năm kinh nghiệm làm
   việc với bệnh nhân tim mạch."
7
8  result = send_prompt(role_prompt, system_message)
9  print("Role prompting result:")
10 print(result)
11 print("\n" + "-"*50 + "\n")
12
13 # Bài tập thực hành:
14 # 1. role prompting là gì?
15 # 2. Viết một role prompt với vai trò "nhà văn nổi tiếng" để viết một đoạn mở đầu truyện ngắn
16 # 3. Viết một role prompt với vai trò "chuyên gia tâm lý" để đưa ra lời khuyên về cách đối phó với stress

```

OUTPUT PROBLEMS SPELL CHECKER 161 COMMENTS TERMINAL DEBUG CONSOLE

(venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 % python role_prompting.py

Role prompting result:
 Chắc chắn rằng có nhiều cách người có thể tăng cường sức khỏe tim mạch của mình thông qua chế độ ăn uống. Dưới đây là một số lời khuyên cơ bản:

1. **Thực phẩm giàu chất xơ**: Hiểu được vai trò của chất xơ trong việc giảm LDL (lipoprotein mật độ thấp - "cholesterol xấu") và tăng HDL (lipoprotein mật độ cao - "cholesterol tốt") đặc biệt quan trọng. Thực phẩm giàu chất xơ bao gồm quả, rau, hạt, và ngũ cốc nguyên hạt.
2. **Cắt giảm chất béo bão hòa và chất béo trans**: Chúng có thể tăng mức cholesterol trong máu và nguy cơ bị bệnh tim. Hạn chế sử dụng các loại thực phẩm chứa lượng chất béo bão hòa cao như bơ, mỡ động vật, kem... Và tránh các sản phẩm có chất béo trans như các loại thực phẩm chế biến sẵn, bánh ngọt...
3. **Ăn nhiều hải sản**: Cá béo như cá hồi, cá ngừ, cá saba... có chứa nhiều omega-3, một loại chất béo tốt có thể giảm nguy cơ bị bệnh tim.
4. **Hạn chế natri**: Natri quá nhiều có thể gây tăng huyết áp. Hạn chế lượng muối trong các bữa ăn và tránh các thức ăn chế biến sẵn.
5. **Duy trì cân nặng khỏe mạnh**: Bạn cần tìm hiểu và xác định mục tiêu cân nặng khỏe mạnh dựa trên chỉ số BMI (Body Mass Index), sau đó phác họa triển một chế độ ăn lành mạnh để duy trì cân nặng đó.
6. **Tăng cường vận động và tập thể dục**: Thêm các bài tập thể dục nhẹ nhàng vào trong cuộc sống hàng ngày của bạn, như đi bộ, yoga hoặc đạp xe cũng giúp tăng cường sức khỏe tim mạch.
7. **Giới hạn cồn**: Nếu bạn uống, hãy tiêu thụ cồn ở mức đồ hại. Đối với phụ nữ, điều này thường có nghĩa là không quá một đơn vị cồn mỗi ngày, và đối với nam giới, không quá hai đơn vị.

Nhớ rằng những thay đổi nhỏ từng ngày có thể tạo ra sự khác biệt lớn trong sức khỏe tim mạch của bạn. Luôn nên tìm kiếm sự hướng dẫn của chuyên gia dinh dưỡng hoặc bác sĩ trước khi thực hiện bất kỳ thay đổi lớn nào về dinh dưỡng.

(venv) (3.11.5) macintoshhd@Macintoshs-MacBook-Pro untitled folder 2 %