Yêu cầu đồ án

I. Yêu cầu cho đồ án PROMPTING

1. Mục tiêu

Các đề tài nghiên cứu này nhằm mục đích tìm hiểu cách áp dụng các kỹ thuật gợi ý (prompting) khác nhau trong Mô hình ngôn ngữ lớn (LLM), bao gồm các phương pháp gợi ý cơ bản (basic prompting), chuỗi suy nghĩ (Chain-of-thoughts) và cây suy nghĩ (Tree-of-thoughts). Bằng cách khai thác các kỹ thuật này, chúng ta có thể ra lệnh cho các LLM thực hiện tác vụ chính xác hơn rất nhiều.

2. Tài liệu tham khảo

- Chuỗi suy nghĩ: https://www.promptingguide.ai/techniques/cot
- Cây suy nghĩ: https://www.promptingguide.ai/techniques/tot
- ChatGPT for Startup: https://drive.google.com/file/d/13QMy4jiUiIcCFPQyxTcldl4g812xws5J/view?usp=sharing

3. Yêu cầu cụ thể

Trong quyển ebook ChatGPT for Startups (Link trong Tài liệu tham khảo) có các kỹ thuật gợi ý cơ bản (basic prompting) cho các lĩnh vực sau:

- Finance And Administration Applications (Ứng Dụng Tài Chính Và Hành Chính);
- Product Development Applications (Ứng Dụng Phát Triển Sản Phẩm);
- Operations And Management Applications (Ứng Dụng Vận Hành Và Quản Lý);
- Marketing And Pr Applications (Ứng Dụng Marketing Và Pr);
- Business Development And Sales Applications (Ứng Dụng Phát Triển Kinh Doanh Và Bán Hàng);
- Customer Service Applications (Ứng Dụng Dịch Vụ Khách Hàng);
- Human Resources Applications (Ứng Dụng Nhân Sự).

Mỗi lĩnh vực trong đây có nhiều mục con (ví dụ: lĩnh vực "Ứng Dụng Tài Chính Và Hành Chính" có 16 mục con). Học viên chỉ chọn **tối thiểu 04 mục con** (ví dụ: II.1->II.4) để xây dựng các chiến lược gợi ý chuyên sâu gồm chuỗi suy luận (CoT) và cây suy luận (ToT).

Ngoài quyển ebook trên, còn có các lĩnh vực ở đường link: https://www.aiforeducation.io/prompt-library. Việc chọn lựa các mục con cũng tương tự như sau: ví dụ trong mục "Special Education" có nhiều mục con gồm: "Transform an Existing Lesson", "Summarize Meeting Notes"... Học viên cần chọn **tối thiểu 04 mục con**

để xây dựng các chiến lược gợi ý chuyên sâu gồm chuỗi suy luận (CoT) và cây suy luận (ToT).

Lưu ý: nếu có nhiều hơn một nhóm chọn cùng 1 chủ đề, giảng viên sẽ sắp xếp các mục con cho từng nhóm để tránh làm trùng nhau.

4. Kết quả đầu ra

Yêu cầu kết quả đầu ra phải có một báo cáo (tập tin MS Word) chi tiết gồm:

- Miêu tả các rõ ràng các tình huống xây dựng gợi ý;
- Chiến lược gợi ý CoT, ToT có đầu vào và đầu ra cụ thể;
- Việc thiết kế gợi ý cần làm cho cả tiếng Anh và tiếng Việt;
- Nêu ra một số trường hợp sử dụng (use-case, best practice) khi sử dụng CoT, ToT cho các tình huống trong đề tài nghiên cứu;
- Đề xuất các hướng phát triển trong tương lai cho các tình huống trong đề tài nghiên cứu;
- Việc thiết kế gợi ý phải được khảo sát trên ít nhất ChatGPT (bao gồm gpt-3.5, gpt-4.0), việc khảo sát thêm trên Gemini, Claude là điểm cộng;
- Việc đưa ra số lượng các tình huống cần được sự đồng ý của GV (sẽ phụ thuộc vào mức độ phức tạp của chiến lược gợi ý học viên đưa ra): sẽ có phần nộp giữa kỳ, lúc đó học viên sẽ được yêu cầu bổ sung hoặc không tùy theo độ phức tạp đã làm được.

II. Yêu cầu cho đồ án DATA

1. Mục tiêu

Để tạo tập ngữ liệu chất lượng cao phù hợp với các tác vụ Xử lý ngôn ngữ tự nhiên (NLP) có thể được sử dụng để đào tạo và đánh giá các mô hình NLP. Cụ thể các bước như sau:

- Xác định Phạm vi: Xác định tác vụ NLP cụ thể (ví dụ: phân loại văn bản, nhận dạng thực thể được đặt tên, dịch máy, phân tích tình cảm). Tác vụ này được đưa ra trong đề tài.
- Thu thập dữ liệu: Thu thập dữ liệu liên quan cho tác vụ.
- Chú thích dữ liệu: Đảm bảo dữ liệu được dán nhãn chính xác theo nhu cầu của tác vụ.
- Xử lý trước dữ liệu: Làm sạch và xử lý trước dữ liệu để nâng cao chất lượng và khả năng sử dụng.

2. Tài liệu tham khảo

- Data sets for NLG: https://aclweb.org/aclwiki/Data_sets_for_NLG

3. Yêu cầu cụ thể

a) Nguồn dữ liệu:

- Xác định các nguồn đáng tin cậy để thu thập dữ liệu (ví dụ: website, API, bộ dữ liệu có sẵn công khai).
- Ví dụ: Dữ liệu Twitter để phân tích cảm xúc, Wikipedia để tạo mô hình ngôn ngữ.

b) Hướng dẫn chú thích (guideline):

- Xây dựng các hướng dẫn rõ ràng cho người chú thích để đảm bảo ghi nhãn nhất quán.
- Ví dụ: Xác định điều gì tạo nên cảm xúc tích cực hoặc tiêu cực trong phân tích cảm xúc.

c) Công cụ:

- Sử dụng các công cụ để thu thập và chú thích dữ liệu (ví dụ: tập lệnh Python để crawl web, các nền tảng chú thích như Labelbox hoặc Prodigy).

d) Những cân nhắc về mặt đạo đức:

- Đảm bảo quyền riêng tư của dữ liệu và sử dụng dữ liệu có đạo đức, đặc biệt nếu sử dụng nội dung do người khác tạo ra.
- Ví dụ: Ẩn danh thông tin cá nhân khỏi nội dung do người dùng tạo.

4. Kết quả đầu ra

a) Bộ dữ liệu chất lượng:

- Một tập dữ liệu rõ ràng và được chú thích rõ ràng phù hợp với tác vụ NLP cụ thể.
- Ví dụ: Tập dữ liệu gồm 10.000 tweet được gắn nhãn để phân tích tình cảm.

b) Tài liệu:

- Tài liệu toàn diện trình bày chi tiết về quy trình tạo tập dữ liệu, nguyên tắc chú thích và mọi bước tiền xử lý đã thực hiện.
- Ví dụ: Tệp README giải thích cách thu thập, chú thích và xử lý trước dữ liệu.
- Ngoài ra, trong tài liệu cũng phải bao gồm bảng thống kê chi tiết về mặt ngữ liệu như số câu, số nhãn, số từ, số từ trung bình...

c) Phân chia dữ liệu:

- Chia tập dữ liệu hợp lý thành các tập huấn luyện (70%), xác nhận (15%) và kiểm tra (15%).
- Ví dụ: Đối với tập dữ liệu 10.000 mẫu, 7.000 mẫu để đào tạo, 1.500 mẫu để xác thực và 1.500 mẫu để thử nghiệm.

d) Đánh giá:

- Độ đo trên các mô hình cơ sở để xác thực tính phù hợp của tập dữ liệu cho tác vụ.
- Ví dụ: Chạy mô hình phân tích cảm xúc **đơn giản** trên tập dữ liệu và báo cáo độ chính xác.

Lưu ý: Học viên dùng công cụ **Label Studio** (https://github.com/HumanSignal/label-studio) để gán nhãn dữ liệu (nếu không có yêu cầu dùng công cụ khác trong đề tài).

Ngoài ra: học viên có thể vào trang https://catalog.ldc.upenn.edu/byproject để xem các bộ ngữ liệu chuẩn và các làm tài liệu chuẩn. Các bước chi tiết như sau:

- Bước 1: vào đường dẫn: https://catalog.ldc.upenn.edu/byproject

- Bước 2: chọn một dự án (ví dụ: Abstract Meaning Representation (AMR) Annotation Release 1.0)

- Bước 3: chọn Samples (Please view this sample) để xem dữ liệu mẫu

- Bước 4: chọn Online Documentation để xem hướng dẫn đánh nhãn dữ liệu (tập tin README.txt)

Học viên có thể dùng các kỹ thuật PROMPTING như few-shot prompting để sử dụng ChatGPT gán nhãn trước và sau đó chỉnh sửa lại (sẽ đỡ mất thời gian hơn làm từ đầu)

Link tham khảo: https://drive.google.com/file/d/18qIFGQ6SEIsSUXDXzDPkYwFVQCieDwHa/view?usp=share_link.

III. Yêu cầu cho đồ án DEMO

1. Mục tiêu

Để phát triển một ứng dụng web demo cho Xử lý ngôn ngữ tự nhiên (NLP) có giao diện dễ thao tác với người dùng. Cụ thể các bước như sau:

- Xác định Chức năng: Chức năng sẽ được nêu cụ thể trong đề tài (ví dụ: phân loại văn bản, nhận dạng thực thể được đặt tên, phân tích tình cảm).
- Thiết kế giao diện người dùng (frontend): tạo giao diện người dùng tương tác và thân thiện với người dùng.
- Phát triển phần cuối (backend): triển khai API để xử lý quá trình xử lý NLP.
- Tích hợp Frontend và Backend: Đảm bảo liên lạc liền mạch giữa giao diện và API.

2. Tài liệu tham khảo

- https://developer.nvidia.com/blog/building-a-machine-learning-microservice-with-fastapi/

3. Yêu cầu cụ thể

a) Công nghệ sử dụng (bắt buộc):

- Ngôn ngữ lập trình: Python (phiên bản 3.10 trở lên);
- Frontend: streamlit;
- Backend: FastAPI;

b) Yêu cầu frontend:

- Người dùng nhập vào văn bản;
- Hiển thị kết quả từ API (backend);
- Đơn giản, dễ sử dụng.

c) Yêu cầu backend:

- Xử lý các yêu cầu đến và xử lý văn bản bằng mô hình NLP;
- Trả về kết quả ở định dạng có cấu trúc (JSON);
- Đảm bảo khả năng mở rộng và hiệu quả.

4. Kết quả đầu ra

a) Ứng dụng website: Một ứng dụng web đầy đủ chức năng mà người dùng có thể tương tác để thực hiện nhiệm vụ NLP được chỉ định.

b) Giao diện người dùng: Giao diện thân thiện với người dùng được xây dựng bằng Streamlit cho phép người dùng nhập văn bản và xem kết quả.

Ví dụ: mã nguồn giao diện streamlit cho ứng dụng phân tích cảm xúc:

```
import streamlit as st
import requests

st.title("Sentiment Analysis Web Application")

text_input = st.text_area("Enter text for analysis:")

if st.button("Analyze"):
    response = requests.post("http://localhost:8000/process_text", json={"text": text_input})
    result = response.json()
    st.write(f"Sentiment: {result['sentiment']}")
    st.write(f"Score: {result['score']}")
```

c) API phụ trợ: API xử lý văn bản và trả về kết quả một cách hiệu quả.

Ví dụ: mã nguồn FastAPI cho ứng dụng phân tích cảm xúc:

```
from fastapi import FastAPI
from pydantic import BaseModel
from transformers import pipeline

app = FastAPI()

sentiment_analyzer = pipeline("sentiment-analysis")

class TextRequest(BaseModel):
    text: str

@app.post("/process_text")
def process_text(request: TextRequest):
    result = sentiment_analyzer(request.text)[0]
    return {"sentiment": result['label'], "score": result['score']}
```

```
if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

d) Kiểm tra và đánh giá:

- Kiểm tra ứng dụng để đảm bảo ứng dụng hoạt động như mong đợi.
- Xác thực các phản hồi API và đảm bảo phản hồi chính xác: viết các testcase với tập ngữ liệu mẫu để có thể kiểm tra bằng cách chỉ chạy API trên bộ test;

e) Tài liệu: Tài liệu toàn diện nêu chi tiết về:

- Cách thiết lập: Cách chạy chương trình, cách huấn luyện, cách kiểm tra/đánh giá (có câu lệnh để huấn luyện và câu lệnh đánh giá);
- Cách sử dụng: cách sử dụng chi tiết;
- Thông số kỹ thuật API: miêu tả API, các tham số, định dạng đầu vào và đầu ra.

Lưu ý: Tổ chức mã nguồn nên được xây dựng theo cấu trúc sau:

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── run.py             <- Make this project runnable by starting gui.py and api.py
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   ├── visualization  <- Scripts to create exploratory and results oriented visualizations
│   │   └── visualize.py
│   │
│   ├── api            <- Scripts to provide api (i.e. fastapi) for gui (streamlit)
│   │   └── api.py
│   │
│   └── gui            <- Scripts to create GUI (use streamlit)
│       └── gui.py
```

Mã nguồn tham khảo:

https://drive.google.com/file/d/11TG0KiPvt68AUIXripamV3YAoq_EQhng/view?usp=sharing

Nếu sử dụng docker sẽ có thêm điểm cộng.

Tham khảo: https://github.com/kurtispykes/car-evaluation-project/tree/Main/packages

IV. Yêu cầu cho đồ án TOOL

1. Mục tiêu

Mục tiêu của dự án này là cung cấp cho sinh viên một hiểu biết sâu sắc về các công cụ AI khác nhau, bao gồm công cụ thương mại, công cụ miễn phí và công cụ mã nguồn mở. Sinh viên sẽ nghiên cứu cách các công cụ này hoạt động, những công nghệ và thuật toán nào được triển khai trong chúng, và cách chúng có thể được áp dụng để giải quyết các vấn đề thực tế. Dự án này nhằm phát triển kỹ năng nghiên cứu, kiến thức kỹ thuật và khả năng tư duy phản biện của sinh viên trong lĩnh vực trí tuệ nhân tạo.

2. Tài liệu tham khảo

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig
- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- Online documentation and user guides for selected AI tools (e.g., TensorFlow, PyTorch, IBM Watson, OpenAI GPT)
- Các bài báo nghiên cứu và bài viết liên quan đến việc triển khai và ứng dụng các công cụ AI

3. Yêu cầu cụ thể

- Nghiên cứu và Phân tích:
    - Nghiên cứu các công nghệ và thuật toán cơ bản được sử dụng trong công cụ đã chọn.
    - Khám phá kiến trúc, chức năng và các tính năng chính của công cụ.
    - Phân tích các điểm mạnh, điểm yếu và các ứng dụng tiềm năng của công cụ.
- Xem xét tài liệu:
    - Xem xét tài liệu chính thức, hướng dẫn và tài liệu người dùng của công cụ đã chọn.
    - Tóm tắt quá trình cài đặt, thiết lập và hướng dẫn sử dụng cơ bản.
- Triển khai và Thử nghiệm:
    - Cài đặt và thiết lập công cụ đã chọn trên môi trường cục bộ hoặc đám mây.
    - Thực hiện các thí nghiệm để kiểm tra khả năng của công cụ (ví dụ: huấn luyện mô hình, xử lý dữ liệu, tạo ra kết quả).

    - Ghi chép chi tiết thiết lập thí nghiệm, quy trình và kết quả.
- So sánh và Đánh giá:
    - So sánh công cụ đã chọn với ít nhất hai công cụ tương tự khác.
    - Đánh giá công cụ dựa trên các tiêu chí như hiệu suất, dễ sử dụng, hỗ trợ cộng đồng và chi phí.
- Báo cáo và Thuyết trình:
    - Chuẩn bị một báo cáo chi tiết về các phát hiện nghiên cứu, kết quả thí nghiệm và đánh giá.
    - Tạo một bài thuyết trình để chia sẻ kết quả của dự án với lớp, nhấn mạnh những điểm chính và kết luận.

4. Kết quả đầu ra

- Hướng dẫn sử dụng (bao gồm cả quá trình chuẩn bị dữ liệu, huấn luyện kiểm tra).
- Báo cáo chi tiết về các phát hiện nghiên cứu, kết quả thí nghiệm và đánh giá.