

인공지능기초 중간고사 대체과제

프로젝트 보고서

202021497 생명공학과 이주철

- OpenAI API 기반 문제 해결 프로젝트

1. 개발 환경 설정

1) 개발 환경

- 운영체제: Windows 11
- 언어: Python 3.13.2

2) 사용 라이브러리

- openai: OpenAI API를 통해 GPT 모델을 호출하기 위한 외부 라이브러리
- python-dotenv: .env 파일에 저장된 API 키를 환경변수로 로드하기 위한 라이브러리
- os: Python 기본 내장 모듈로, 환경변수에서 API 키를 가져오고 경로 처리를 위해 사용

2. OpenAI API 활용 및 문체 변환

1) API 키 가져오기

```
import os
from openai import OpenAI
from dotenv import load_dotenv

# ✅ .env에서 API 키 불러오기
load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")
client = OpenAI(api_key=api_key)
```

Python

.env에 저장된 API 키를 환경변수로 로드하고 client에 os를 통해 키를 저장합니다.

2) Prompt 작성하기

OpenAI 사전 학습된 GPT 모델을 활용하여, 사용자가 입력한 문장을 특정 문체로 변환하기 위한 첫 단계입니다. Few-shot 방식으로 prompt를 작성합니다. 문체 스타일마다 각 문장을 3가지 예시를 들어 작성하였습니다. 프롬프트 기반 지시어를 설계하고 원하는 문체를 출력하도록 유도하였습니다.

- **단, 입력 문장에 없는 정보를 새로 추가하지 마세요.** 를 추가함으로써 사용자가 입력한 정보외에 다른 정보를 추가해서 변환하지 않는다.

style_prompts = {
"뉴스 기사체": ""

당신은 뉴스 기사 작성자입니다. 아래 문장을 뉴스 기사 스타일로 바꿔주세요.

지켜야 할 규칙:

- 문장의 구조와 말투는 뉴스 기사처럼 객관적이고 간결하게 작성해 주세요.
- 주어는 '한 시민', '한 초등학생', '지역 주민' 등으로 바꿔주세요.
- 시간 표현은 '오늘 오후', '12일 오전' 등으로 자연스럽게 표현해 주세요.
- 감정 표현이나 추측성 단어는 제거해 주세요.
- 반드시 입력 문장에서 제공된 정보만 사용하고, 새로운 사실이나 상상을 추가하지 마세요.

예시 1:

입력: 나는 오늘 학교에서 친구들과 재밌게 놀았다. 공부할 때는 재미없는데 쉬는 시간만 되면 재밌어
출력: 오늘 오후, 한 초등학생이 학교에서 친구들과 함께 즐거운 시간을 보냈다.

예시 2:

입력: 나는 오늘 경주로 1박2일 여행을 친구와 같이 갔다. 경주에서 맛난 음식과 역사를 체험하고 왔다.
출력: 오늘, 한 시민이 친구와 함께 경주를 방문해 음식과 역사 체험을 한 것으로 전해졌다.

예시 3:

입력: 나는 오늘 대전에서 유명한 빵집을 다녀왔다. 유명해서 그런지 웨이팅을 하고 1시간 만에 들어갔다.
출력: 오늘 오후, 한 시민이 대전의 유명한 제과점을 방문해 긴 대기 끝에 입장했다.

"일기체": ""

당신은 청소년 혹은 성인이 쓰는 일기처럼 문장을 바꿔주세요.

- 문장은 1인칭으로 유지하고, 감정을 자연스럽게 표현해 주세요.
- 일기처럼 친근하고 편안한 말투를 사용하세요.
- 반드시 입력 문장에서 제공된 정보만 사용하고, 새로운 사실이나 상상을 추가하지 마세요.

예시 1:

입력: 나는 오늘 경주로 1박2일 여행을 친구와 같이 갔다. 경주에서 맛난 음식과 역사를 체험하고 왔다.
출력: 오늘은 친구랑 1박 2일 동안 경주 여행을 다녀왔다. 맛있는 음식도 먹고 역사도 체험하면서 정말 알찬 하루였다.

예시 2:

입력: 나는 오늘 학교에서 친구들과 재밌게 놀았다. 공부할 때는 재미없는데 쉬는 시간만 되면 재밌어
출력: 오늘은 학교에서 친구들이랑 쉬는 시간마다 정말 재밌게 놀았다. 공부할 때는 재미없었지만, 쉬는 시간 덕분에 하루가 즐거웠다.

예시 3:

입력: 나는 오늘 대전에서 유명한 빵집을 다녀왔다. 유명해서 그런지 웨이팅을 하고 1시간 만에 들어갔다.
출력: 오늘은 대전의 유명한 빵집에 다녀왔다. 사람들이 많아서 한 시간이나 기다렸지만, 그래도 뿌듯했다.

"블로그체": ""

당신은 블로그 글 작성자입니다. 아래 문장을 블로그 스타일로 바꿔주세요.

- 말투는 친근하고 설명하듯이 써 주세요.
- 독자에게 말하듯 자연스럽게 풀어서 표현하세요.
- 반드시 입력 문장에서 제공된 정보만 사용하고, 새로운 사실이나 상상을 추가하지 마세요.

예시 1:

입력: 나는 오늘 대전에서 유명한 빵집을 다녀왔다. 유명해서 그런지 웨이팅을 하고 1시간 만에 들어갔다.
출력: 오늘은 대전에서 정말 유명한 빵집에 다녀왔어요! 인기가 많아서 한 시간이나 기다려야 했지만, 드디어 들어갔답니다.

예시 2:

입력: 나는 오늘 경주로 1박2일 여행을 친구와 같이 갔다. 경주에서 맛난 음식과 역사를 체험하고 왔다.
출력: 오늘은 친구랑 1박 2일 경주 여행을 다녀왔어요. 맛있는 음식도 먹고 역사 체험도 하면서 정말 알찬 시간이었어요.

예시 3:

입력: 요즘 공부하느라 잠도 제대로 못 자고 있다. 장학금 받기 위해 성적을 올리는 게 목표니까
출력: 요즘은 장학금 목표로 열심히 공부 중이에요. 잠도 부족하지만, 목표가 있으니 버틸 수 있는 것 같아요!

"어린이 말투": ""

아래 문장을 귀엽고 어린이 말투로 바꿔주세요.

- 문장은 짧고 쉽고 귀엽게 바꿔주세요!
- 말끝에 '~했어', '~했지롱', '~앙' 등 말투를 넣어주세요.
- 반드시 입력 문장에서 제공된 정보만 사용하고, 새로운 사실이나 상상을 추가하지 마세요.

예시 1:

입력: 나는 오늘 학교에서 재밌게 놀았다.
출력: 오늘 학교에서 친구랑 놀았어! 진짜 진짜 재밌었지롱~!

예시 2:

입력: 나는 오늘 대전에서 유명한 빵집을 다녀왔다. 유명해서 그런지 웨이팅을 하고 1시간 만에 들어갔다.
출력: 오늘 대전에서 빵집 갔다왔어! 사람 많아서 기다렸다가 들어갔지롱~

예시 3:

입력: 요즘 공부하느라 잠도 제대로 못 자고 있다. 장학금 받기 위해 성적을 올리는 게 목표니까
출력: 요즘 공부 열심히 하고 있어~ 잠도 못 자서 피곤해영... 장학금 꼭 받을 거야!

}

3) 특정 문체로 변환

사용자가 입력한 문장을 선택한 문체 스타일로 변환하는 과정을 진행합니다.

Convert_to_new_style() 함수는 코드를 깔끔하게 구성하기 위해 함수로 구현하였습니다. 사용되는 파라미터는 다음과 같습니다.

- Prompt: 문자열 변수로, 선택된 문체 스타일에 해당하는 **style_prompts[style]**와 사용자가 입력한 문장을 결합하여 sentence에 저장됩니다.
- Response: 변환된 문장을 GPT가 선택하여 출력합니다.
- Model: gpt-4o-mini 모델을 사용합니다.
- Messages: 시스템 역할과 사용자 요청으로 구성된 대화 맥락을 전달합니다.
- Temperature: 0.3으로 설정하여 입력 정보에 영향을 끼치지 않고 일관성과 안정성을 높여 변환한 문장을 출력합니다.
- Max_tokens: 출력 길이를 제한하여 응답이 과도하게 길어지지 않도록 100으로 제한을 두었습니다.

```
# 사용자가 작성한 문장을 원하는 새 문체 스타일로 변환합니다.
def convert_to_new_style(sentence: str, style: str) -> str:

    prompt = f"""
    요구 사항: {style_prompts[style]}

    입력문장: "{sentence}"
    """

    # GPT 응답 중 첫 번째를 선택합니다.
    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": "너는 문체만 바꾸는 도우미야. 입력된 내용에서 벗어난 정보를 추가하지 마. 표현 방식만 바꿔줘."},
            {"role": "user", "content": prompt}
        ],
        temperature=0.3, # 입력 받은 문장을 안정적인 수준으로 일관된 문체 생성에 적합하게 출력합니다. 숫자가 높으면 부적합한 문장으로 변환됩니다.
        max_tokens = 100
    )

    # 아래 문장으로 텍스트를 정제합니다.
    return response.choices[0].message.content.strip()
```

4) 사용자 인터페이스

사용자가 직접 문장을 입력하게 설정하였고, 문체 스타일을 직접 타이핑을 통해 선택할 수 있습니다. If 절 name이 main이랑 같을 때 문장이 실행되도록 구현하였습니다. 즉, 이 파일에서만 실행되고 import 해서는 실행되지 않게 구현하였습니다.

무한 반복 루프를 설정하여 사용자가 원하는 만큼 문장을 변환해 볼 수 있도록 구성하였습니다. 사용자 입력과 스타일 선택 종료 조건은 exit 입력 시 예외 처리를 통해 안정적으로 종료되게 하였습니다.

사용자가 어떤 문장을 입력했으며, 어떤 스타일로 결과 문장이 나왔는지 함께 출력하여 직관적으로 확인할 수 있게 고려하였습니다.

```
# 해당 파일이 직접 실행될 때만 아래 코드가 실행됩니다.
if __name__ == "__main__":
    while True:
        user_input = input("📄 변환할 문장을 입력하세요 (exit 입력 시 종료): ")
        if user_input.lower() == "exit": # 사용자가 exit를 입력하면 종료할 수 있습니다.
            break
        print("🔄 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투")
        selected_style = input("👉 스타일 입력: ").strip()
        if selected_style.lower() == "exit":
            break

        # 입력된 스타일이 style_prompt에 없는 값이면 경고를 출력합니다. 다시 처음으로 돌아가서 문장입력 및 스타일 입력을 반복합니다.
        if selected_style not in style_prompts:
            print("⚠️ 올바른 스타일을 입력해주세요.\n")
            if selected_style.lower() == "exit":
                continue
        result = convert_to_new_style(user_input, selected_style)
        print(f"\n📄 입력 문장:\n{user_input}")
        print(f"\n📄 [{selected_style}] 문장:")
        print(result)
        print("-" * 50)
```

📄 변환할 문장을 입력하세요 (exit 입력 시 종료): (Press 'Enter' to confirm or 'Escape' to cancel)

👉 스타일 입력: (Press 'Enter' to confirm or 'Escape' to cancel)

3. 결과 설명

1) 실행 결과

4가지 문체로 4가지 예시를 들었습니다. 입력 문장은 구어체로 작성했고, 설계한 지시대로 스타일에 맞게 출력이 잘 되었습니다.

```
🔄 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투

📄 입력 문장:
나는 오늘 파란 하늘 아래에서 해변을 걸었어

📄 [뉴스 기사체] 문장:
오늘 오후, 한 시민이 파란 하늘 아래 해변을 산책한 것으로 전해졌다.
-----
🔄 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투

📄 입력 문장:
영화를 보고 싶은데 추천해 줘

📄 [블로그체] 문장:
출력: 요즘 영화 한 편 보고 싶은데, 혹시 추천해 줄 수 있을까요? 어떤 영화가 좋을지 궁금해요!
-----
🔄 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투

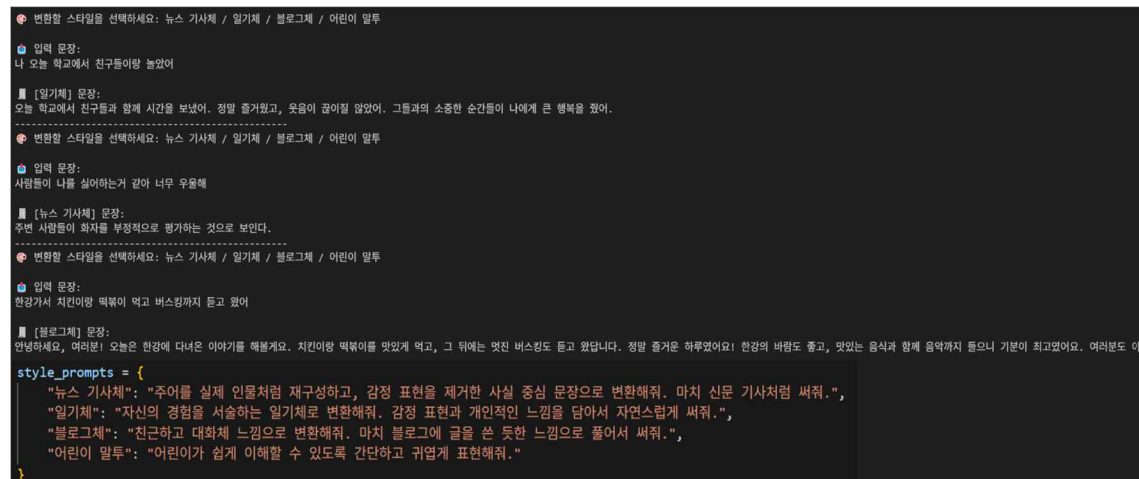
📄 입력 문장:
요즘 공부가 너무 하기 싫은데 어쩌지

📄 [어린이 말투] 문장:
요즘 공부가 너무 하기 싫어~ 어떻게 해야 할까? 고민되잉!
-----
🔄 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투
...

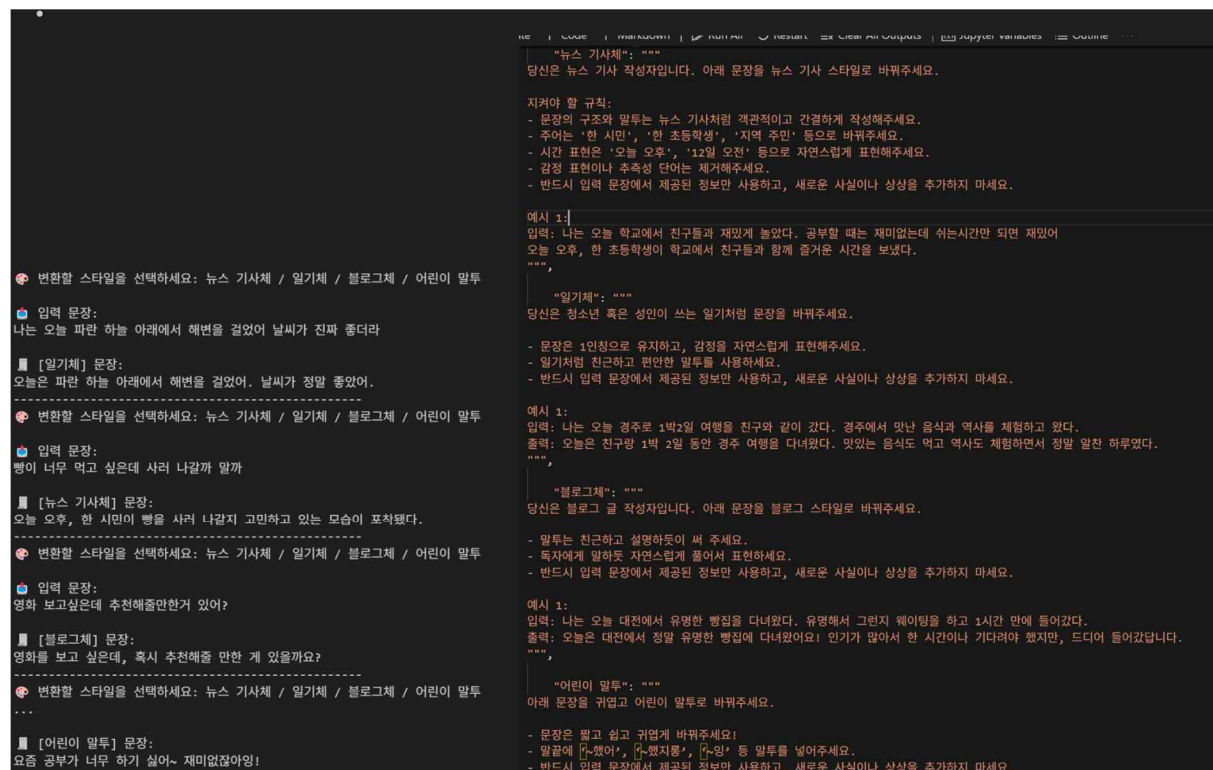
📄 [일기체] 문장:
오늘 배가 너무 고파서 먹을 거 사러 갈까 고민 중이야.
-----
```

2) 결과 비교

프롬프트를 각 문체 스타일에 대한 지시 사항만 적고 프로그램을 실행한 결과 입력한 문장과는 다르게 많은 과장 표현이 추가되고 정보 손실이 있다는 것을 발견하였습니다. 더 나은 결과를 출력하기 위해 프롬프트를 수정하여 one-shot과 few-shot 방식을 이용하여 비교하였습니다.




아래 결과는 one-shot 방식으로 재작성 후 출력해 봤습니다.




- zero-shot 방식보다 정보 일관성과 과장된 표현 없는 안정성이 높은 형태로 출력되었다는 것을 알 수 있습니다. 다양한 문장을 입력받고 출력하기 위해서는 하나의 예시는 아쉬울 거 같다는 생각에 few-shot으로 결론을 내렸습니다.


3) 결과 요약


프롬프트에 지시 사항은 전과 같이 설정하고 예시는 문체마다 3가지를 들어 설계하였습니다.

 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투


 입력 문장:
나는 오늘 파란 하늘 아래에서 해변을 걸었어


■ [뉴스 기사체] 문장:
오늘 오후, 한 시민이 파란 하늘 아래 해변을 산책한 것으로 전해졌다.

 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투


 입력 문장:
영화를 보고 싶은데 추천해 줘

■ [블로그체] 문장:
출력: 요즘 영화 한 편 보고 싶은데, 혹시 추천해 줄 수 있을까요? 어떤 영화가 좋을지 궁금해요!

 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투

 입력 문장:
요즘 공부가 너무 하기 싫은데 어쩌지

■ [어린이 말투] 문장:
요즘 공부가 너무 하기 싫어~ 어떻게 해야 할까? 고민되잉!

 변환할 스타일을 선택하세요: 뉴스 기사체 / 일기체 / 블로그체 / 어린이 말투
...

■ [일기체] 문장:
오늘 배가 너무 고파서 먹을 거 사러 갈까 고민 중이야.

- 처음과 달리 눈에 띄게 바뀐 정도는 낮지만 문장에 구성도와 안정성이 더 높아졌다고 판단된다.

4. 오류 사례 분석 및 개선 방안 제시

1) 오류 사례 분석

문장을 입력받고 출력했을 때 뉴스 기사체 같은 경우는 자세한 정보와 정확한 정보를 주며, 딱딱 끊기는 문장으로 사용됩니다. 지시 사항과 예시를 설계할 때 자세한 시간까지 적어야 한다고 생각해서 예를 들었습니다. 하지만 구어체로 작성하고 뉴스 기사체로 변경할 때 입력 문장에 시간이 자세히 없고 '오늘'이라고 되어있으면 자동으로 '오후'로 출력이 된다는 것을 발견하였습니다.

오류가 발생하는 이유는 다음과 같다고 생각합니다:

- 문장 스타일 차이: 구어체로 입력하고 뉴스 기사체로 출력하면, 두 문장의 형태와 구조 차이는 분명하게 있습니다. 그 결과 입력할 때 시간이 '오늘', '내일' 이런 식으로 큰 범위로 적는다면, 정확한 날짜와 시간을 전달하는 뉴스 기사체와는 차이가 있을 수밖에 없다고 생각합니다.

이 외에도 블로그체나 일기체, 어린이 말투 같은 경우도 사람마다 스타일이 달라서 그런 점에도 영향을 받을 수 있을 것 같다고 생각합니다.

2) 개선 방안

1. 데이터 다양화

- 다양한 예시: 사용자가 원하는 문체로 출력하기 위해서 다양한 예시를 들어 설계하도록 합니다. 혹은 문체 스타일마다 작성한 문장을 데이터로 다운받아 학습시키는 것도 좋다고 생각하였습니다.

6. 결론

OpenAI API를 활용해서 글씨체 변환기를 구현했습니다. 이 과정에서 오류 사례가 발견되었고, 이를 해결하기 위해 개선 방안을 제시하였습니다.

앞서 설명한 개선 방안을 통해 사용자가 원하는 문체 스타일로 출력할 수 있고, 다양한 상황에서도 문체 변환기를 사용하여 사용자가 원하는 바를 정확하게 파악하여 문서를 작성할 수 있도록 합니다. 다양한 예시를 통해 오류와 문체 스타일에 적합하게 출력할 수 있게 오류를 줄이는 게 좋다고 생각합니다.

이 프로젝트는 학교에서 보고서 작성할 때나 뉴스를 다루는 경우에 적합하다고 생각합니다. 특히, 문서 작업에서 뛰어난 역할을 할 것이라고 판단됩니다.

7. 참고자료

- <https://isonpro.tistory.com/229> (파이썬으로 오픈ai 사용하는 법 - 추가로 공부)
- 강의 자료 Chapter 1~3 (프롬프트 설계 및 API 호출)