**REPORT**

**오픈소스SW활용 최종 보고서**

**과 목 명 : 오픈소스SW활용**

**담당교수 : 송 인 식**

**학 과 : 소프트웨어**

**학 번 : 32144794**

**이 름 : 최 원 제**

**제 출 일 : 06. 07.**

**목차**

1. **프로젝트 배경**
2. **제공 기능**
3. **시스템 구조 및 설계**
4. **구현 과정**
5. **테스트 결과 및 분석**
6. **후기**
7. **참고 문헌**
8. **프로젝트 배경**

최근 OpenAI사에서 제공하는 다양한 AI 모델들에 대해 관심이 있었고 이를 API의 형태로 활용하기 위해 Spring boot를 통한 웹 서비스 프로젝트를 해본 경험이 있다.

하지만 OpenAI에서 제공하는 공식 기술 문서와 인터넷 상의 많은 자료들에서는 python언어를 사용한 레퍼런스들이 대부분이었고 JAVA를 사용하는 Spring boot에서는 전용으로 만들어진 의존성을 주입하거나 보다 복잡한 과정을 거쳐야만 했다.

따라서 이번 프로젝트에서는 python을 활용해 보다 쉽고 간편하게 API를 연동해보고 이를 활용한 프로그램을 PyQt5f를 통해 GUI로 제작해보는 것이 목표였다.

GitHub 주소: <https://github.com/1J-Choi/ChatGPTTranslateGUI>

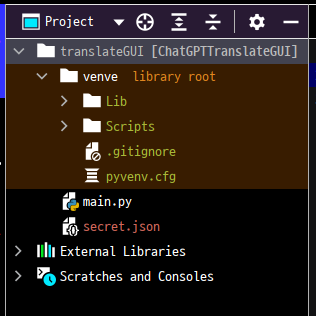
1. **제공 기능**

해당 프로그램은 GPT-3.5 모델의 API 기능을 통해 입력한 문장을 각각 다른 나라의 언어로 번역해주는 GUI 형태의 서비스이다.

해당 프로그램에서는 선택 가능한 언어로 GPT-3.5가 지원하는 언어인 영어, 일본어, 중국어, 한국어, 스페인어, 프랑스어, 독일어, 이탈리아어, 포르투갈어, 네덜란드어, 러시아어를 입력하거나 해당 언어로 번역이 가능하다.

이 외에도 GPT-3.5 모델은 현대에 사용되는 언어 뿐만 아니라 사어[[1]](#footnote-1), 우리나라의 하오체와 같은 고어투, 심지어 암호문이나 모스부호까지 해석할 수 있기 때문에 해당 항목을 추가하게 되면 번역 기능을 추가할 수 있다.

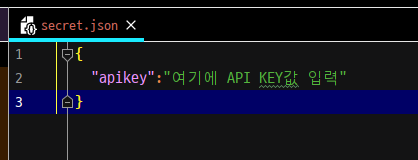
1. **시스템 구조 및 설계**



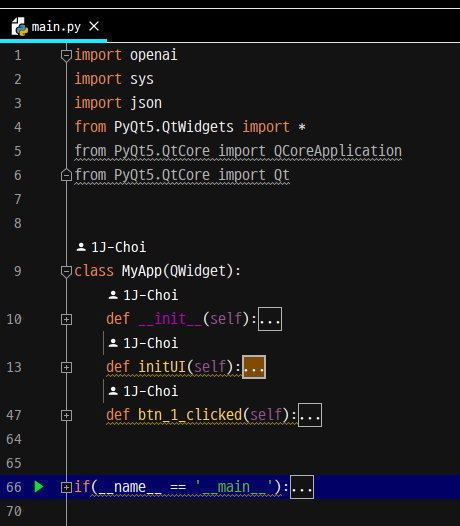
해당 프로그램은 PyQt5를 통해 만들어진 GUI 프로그램이다.

따라서 main.py라는 하나의 Python 파일에서 기능 및 GUI 구성 함수를 코딩하였다. 해당 프로그램에서는 OpenAI의 API 서비스를 활용하기 때문에 OpenAI에서 제공하는 API Key 값이 필요한데 이를 main.py에 작성한 뒤 GitHub에 올리게 되면 해당 코드가 노출 되기 때문에 아래의 secret.json 파일의 형태로 따로 뺀 뒤 GitHub에 올라가지 않도록 하였다.

따라서 해당 프로그램을 clone 하여 사용하려는 경우 개인이 직접 API Key값을 발급받은 뒤 secret.json 파일을 작성 해야하며, 이는 아래의 사진과 같다.



1. **구현 과정**



앞서 설명하였듯이 해당 프로그램은 PyQt5를 사용한 GUI 프로그램이기 때문에 이를 만들기 위한 기본적인 함수 구조를 따라갔다. initUI에서 작성한 UI 요소들의 경우 원하는 언어를 선택할 수 있는 QComboBox, 텍스트를 입력받는 QTextEdit, 번역 기능을 실행하는 QPushButton, 번역 결과물을 출력하는 QTextBrowser로 구성되어있다.

또한 위에서 설명한 PyQt5와 함께 사용된 모듈로는 OpenAI API 기능을 사용하기 위한 openai, API key값 관리를 위한 json, GUI 프로그램 실행을 위한 sys 등이 있다.



위의 함수들 중에서 가장 중요한 부분은 번역 기능을 실행하는 함수인 btn\_1\_clicked 함수였다.

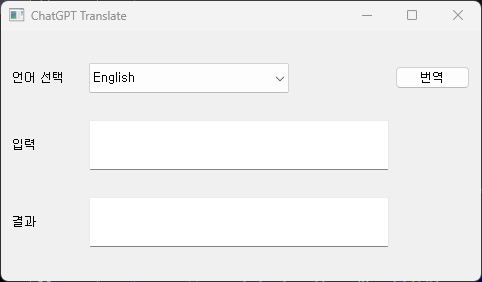
해당 함수가 실행되면 ComboBox와 TextEdit에서 입력된 값을 받아와 OpenAI로 보낼 지시어인 prompt를 생성한다. 이후 분리되어 저장된 API Key 값을 받아 온 뒤 openai의 text-davinci-003 모델에 prompt를 보내 원하는 결과값을 response에 받아온다.

이후 response에 담긴 번역 결과를 TextBrowser 항목에 출력하면 되는데

response는 JSON의 형태이기 때문에 원하는 항목을 추출하고 보기 좋게 수정하여 출력 값으로 보내준다.

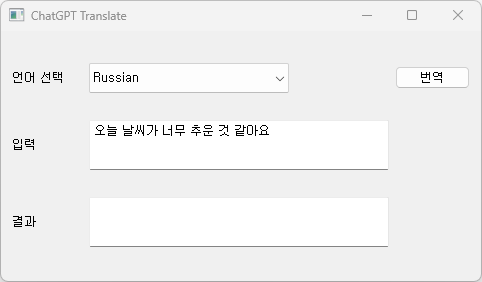
1. **테스트 결과 및 분석**

시연 동영상: <https://youtu.be/Q7jevApOwJQ>

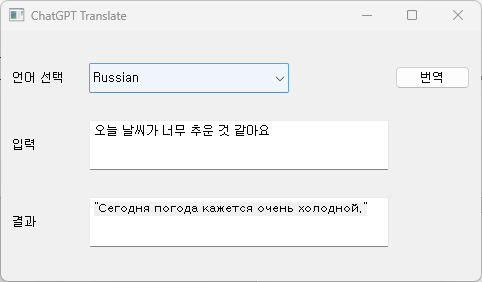


완성된 프로그램을 실행하게 되었을 시 위 그림과 같은 화면이 출력된다.

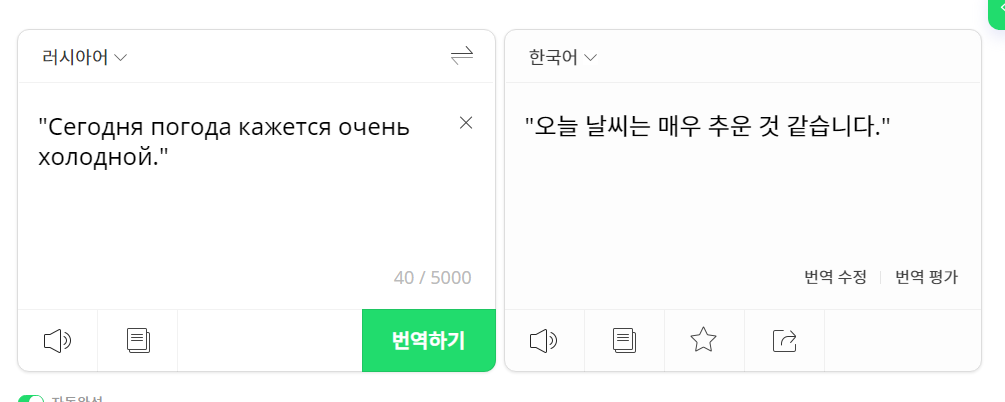
맨 위쪽의 언어 선택 항목을 통해 내가 입력한 문장을 어떤 언어로 번역할 것인지를 선택할 수 있다.



해당 예시에서는 러시아어를 선택하였으며 번역을 원하는 한국어 문장을 입력 항목에 작성한 뒤 번역 버튼을 누른다.



잠시 후 결과 항목에 번역 된 문장이 출력되게 된다.



번역된 문장을 papago와 같은 번역 서비스에 입력해본 결과 입력한 한국어와 일치하는 것을 확인하였다.

이 외의 다른 언어의 입력 및 번역을 테스트한 결과 정상적으로 작동하였다.

1. **후기**

본래 해당 수업에서는 OpenAI사에서 오픈소스로 공개한 3D 객체 생성 AI 모델인 Point-E를 사용한 프로그램을 만드려 했다.

하지만 로컬 컴퓨터에서 해당 모델을 구동하는 데는 지나치게 긴 로딩 시간이 요구되어 테스트 및 구현에 어려움이 있어 포기하게 되었다.

또한 배경에서 서술한 바와 같이 python을 사용하게 되면서 python의 대표적인 웹 프레임워크인 Django를 사용해보려 했으나, 해결되지 않는 오류로 인해 이 또한 포기하게 되었다.

때문에 기존에 구상하였던 프로젝트 보다 작은 규모의 결과물이 나온 것 같아 아쉽다.

하지만 해당 프로젝트를 진행하면서 python을 활용한 개발 환경에 보다 친숙해졌고 REST API와 복잡한 형태의JSON 자료를 다뤄볼 수 있었다.

이를 바탕으로 이후에도 해당 프로그램을 발전시켜 보다 많은 기능을 넣고 가능하다면 Django를 사용해 웹 서비스의 형태로도 구현하여 프로젝트를 발전시켜나갈 예정이다.

1. **참고 문헌**

* “OpenAI API”, OpenAI, <https://platform.openai.com/>

1. 과거에는 쓰였으나 현재는 사용되지 않는 말 ex) 고대 중국어, 라틴어, 산스크리트어 [↑](#footnote-ref-1)