

Detail Design

산업보안학과 32200492 김명현

1. Coding

빠대 소스코드 (추가중입니다.)

```
from googletrans import Translator
import openai
import re

# Google Translate API를 사용하여 번역을 가져옵니다.
translator = Translator()
translated_text = translator.translate("번역할 문장", src='원본 언어', dest='번역할 언어').text

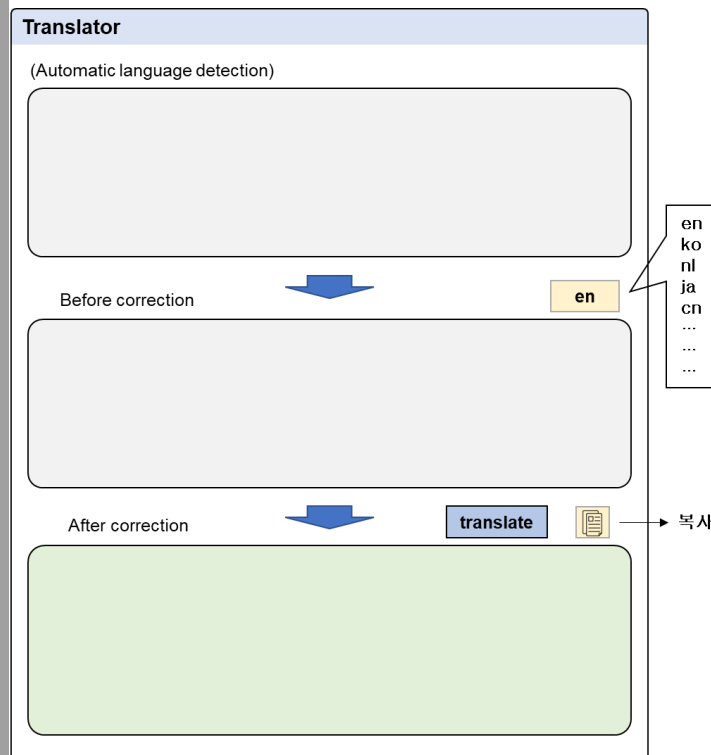
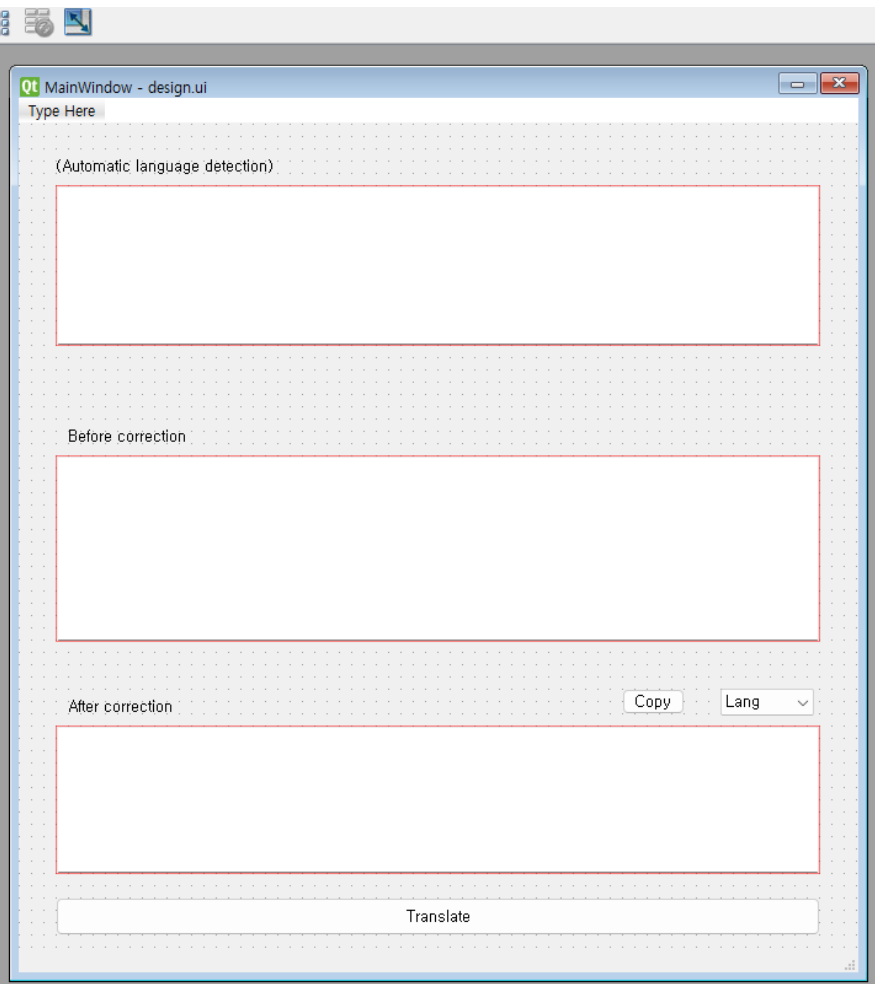
# openai의 GPT 모델을 사용하여 문장을 교정합니다.
openai.api_key = "YOUR_API_KEY"
model_engine = "gpt-3.5-turbo"
corrected_text = openai.Completion.create(
    engine=model_engine,
    prompt=(f"Please correct the following sentence: {translated_text}"),
    temperature=0.5,
    max_tokens=60,
    n = 1,
    stop=None,
)

# 교정된 문장을 추출합니다.
corrected_text = corrected_text.choices[0].text

# 필요한 경우 문장 끝의 공백 및 줄바꿈 문자를 제거합니다.
corrected_text = re.sub(r'[\n\s]+', ' ', corrected_text.strip())

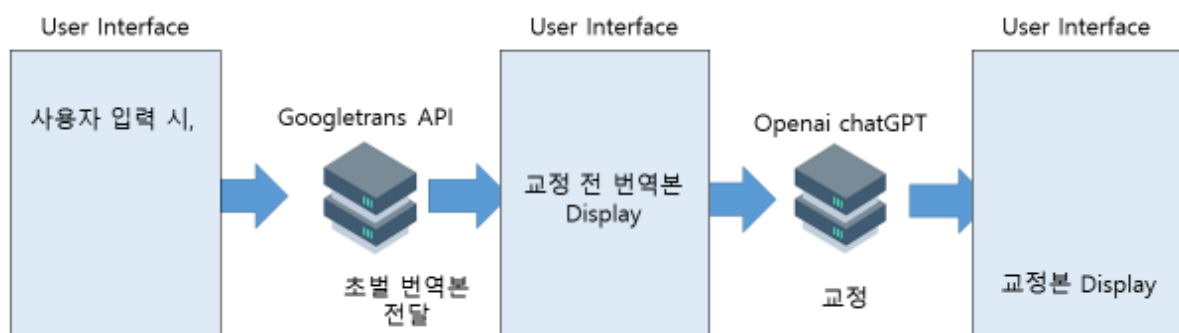
print(f"원본 문장: {translated_text}")
print(f"교정된 문장: {corrected_text}")
```

기존에 구상했던 user interface에 맞추어 "Qt designer"라는 gui making tool로 제작하고 있습니다. Gui 만드는 것을 처음해봐서 python code와 ui를 연결하고 기능을 추가하는 작업을 진행 중에 있습니다.



2. Architecture Design

전체적인 architecture design은 다음과 같습니다.



또한 구현해야 할 기능은 다음과 같이 정리해보았습니다.

1) 편의 기능

- 언어 선택 버튼: 내가 번역하고자 하는 언어를 선택할 수 있는 버튼으로 기존에 많이 사용하는 언어 위주로 선택 가능하게 할 생각이다.
- 번역 버튼: 버튼을 누르면 번역이 실행되서 교정 전 초벌 번역본과 교정 후 최종 번역본을 같이 볼 수 있다.
- 복사 버튼: 교정 후 최종 번역본을 복사할 수 있는 버튼이다.

2) 추가 기능

- 번역이 진행되는 동안 모든 버튼 비활성화
- (more else?)

3. Software Used

이전에 설명했던 기술들을 정리해놓았습니다.

(1) Googletrans

1) 번역(translate)

translator.translate(src, dest, text, pronunciation)

요소	의미
src	번역 전 언어
dest	번역 후 언어
text	번역 결과
pronunciation	번역 결과의 발음 (영어로 표기)

Ex) KOR to ENG

```
Translator.translate('안녕', src='ko', dest='en').text
```

```
>>> 'Hi'
```

2) 탐지(detector)

translator.detect (lang, confidence)

요소	의미
----	----

lang	감지한 언어
confidence	결과의 신뢰도 (0~1.0)

감지 가능한 언어 리스트

LANGUAGES = {


<https://py-googletrans.readthedocs.io/en/latest/> 참조


}

(2) openai

Api key 발급 필요

<https://platform.openai.com/account/api-keys>


[Overview](#)
[Documentation](#)
[API reference](#)
[Examples](#)
[Playground](#)

[Upgrade](#)
[Help](#)

Personal

ORGANIZATION

Personal

Settings

Usage

Members

Billing

USER

API Keys

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED
sk-...D19M	2023년 3월 28 일	Never

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

ChatGPT API 사용

- **model:** 'gpt-3.5-turbo' 모델을 설정
- **query:** 질문을 입력, 예를 들어, 텍스트로 이미지를 생성하는 방법 등

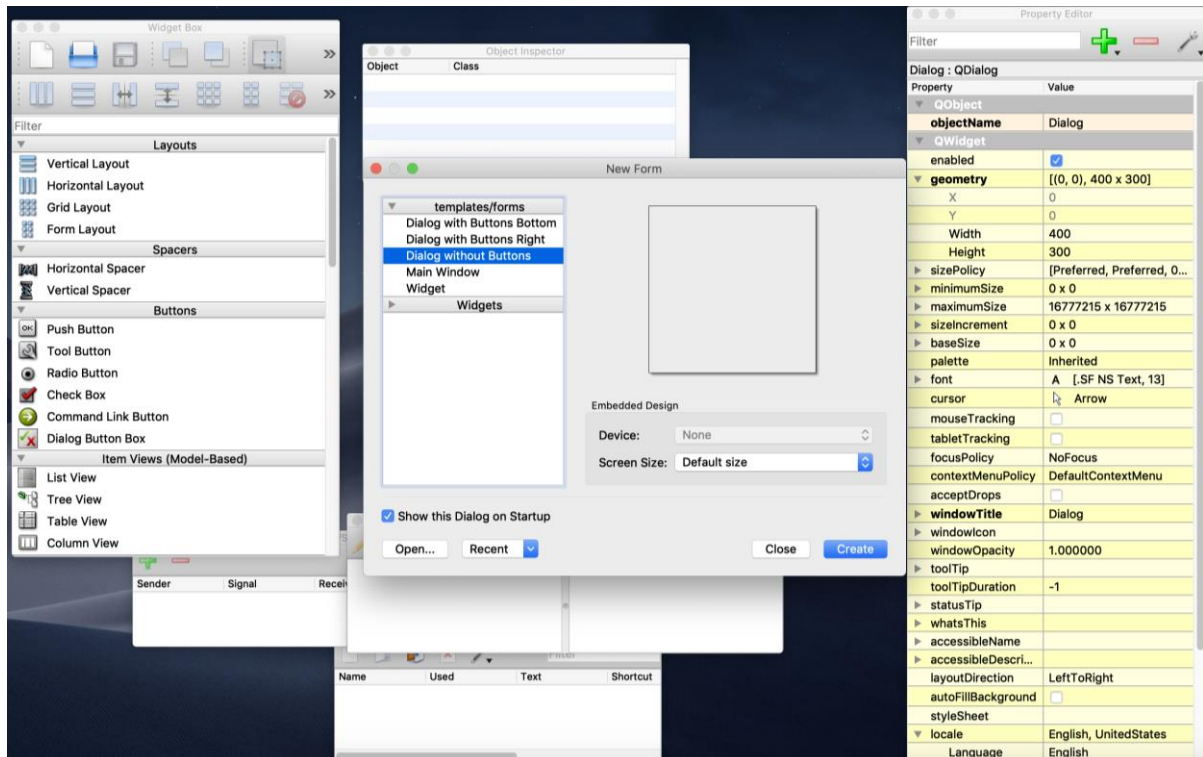
- **messages:** 대화에 참여하는 여러 **역할**과 **메시지 내용**을 설정할 수 있다. 보통 대화는 먼저 시스템 메시지로 형식을 정의하고 사용자와 도우미의 메시지를 번갈아가며 정의한다.
 - **시스템 메시지:** 'You are a helpful assistant.'와 같은 메시지로 도우미에게 지시할 수 있습니다. 시스템이 챗봇에게 일종의 역할을 부여.
 - **사용자 메시지:** 도우미에게 직접 전달하는 내용
 - **도우미 메시지:** 이전에 응답했던 결과를 저장해 대화의 흐름을 유지할 수 있도록 설정 가능

openai.ChatCompletion.create()에 파라미터 값 입력하고 response 에 할당

'response['choices'][0]['message']['content']'를 호출 후 응답결과를 'answer'에 저장한다음 출력한다.

(3) Qt Designer

Qt Designer란, PyQt를 이용하여 GUI프로그래밍을 할 때 손쉽게 프로그램의 레이아웃을 편집할 수 있도록 해주는 편집기



UI파일과 Python코드를 연결시키는데는 UI파일을 통채로 Python코드로 변환시킨 후 그 파일에 코드를 작성하는 방법, UI파일을 Python 코드에서 import하는 방법 이렇게 두가지가 있다. 그중 UI파일을 Python에 Import하여 사용하는 방법을 찾아봤다.

UI파일과 동일한 위치에 Python파일을 하나 만든 후, 아래의 코드를 그대로 복사/붙여넣기를 합니다. 중간에 있는 "UI파일이름.ui"는 자신이 가지고 있는 UI의 이름으로 수정한다.

```
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

#UI파일 연결
#단, UI파일은 Python 코드 파일과 같은 디렉토리에 위치해야한다.
form_class = uic.loadUiType("UI파일이름.ui")[0]

#화면을 띄우는데 사용되는 Class 선언
class WindowClass(QMainWindow, form_class) :
    def __init__(self) :
        super().__init__()
        self.setupUi(self)

if __name__ == "__main__" :
    #QApplication : 프로그램을 실행시켜주는 클래스
    app = QApplication(sys.argv)

    #WindowClass의 인스턴스 생성
    myWindow = WindowClass()

    #프로그램 화면을 보여주는 코드
    myWindow.show()

    #프로그램을 이벤트루프로 진입시키는(프로그램을 작동시키는) 코드
    app.exec_()
```

4. Reference

구글 번역관련

<https://py-googletrans.readthedocs.io/en/latest/>

<https://cloud.google.com/translate/docs/basic/translating-text?hl=ko>

chatgpt 관련

<https://teddylee777.github.io/python/chatgpt-blog-automation/>

<https://platform.openai.com/account/api-keys>