

# 중간 보고서

산업보안학과 32200492 김명현

주제: 향상된 번역 프로그램

## 1. Detail Design

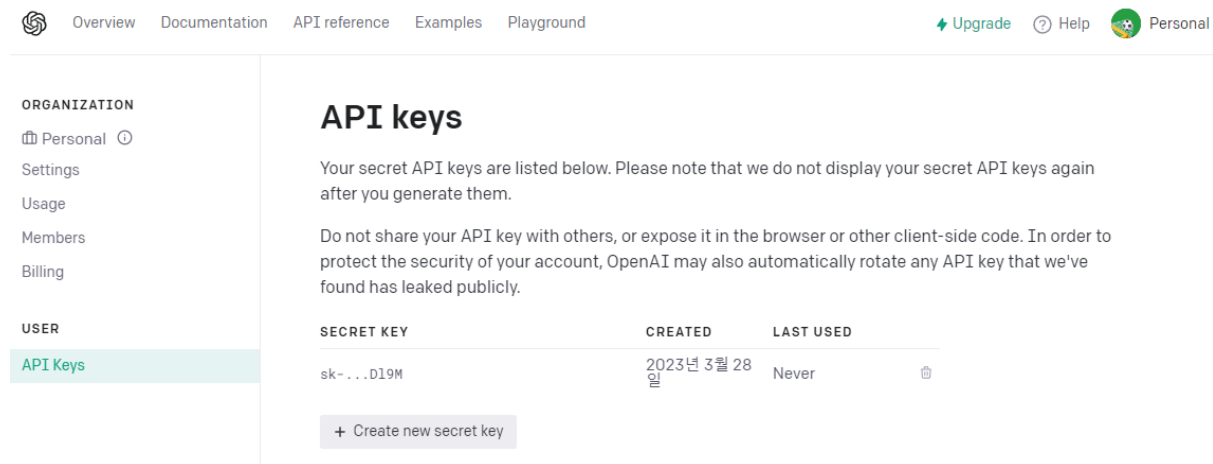
전체적인 ui는 다음과 같이 설계했다.

The image displays a user interface for a translation application. It features a light gray background with a dotted grid pattern. The interface is organized into several sections:

- (Automatic language detection):** A large, empty rectangular box with a red border, intended for the user's input text.
- Before correction:** A second large, empty rectangular box with a red border, positioned below the first one.
- After correction:** A third large, empty rectangular box with a red border, located below the 'Before correction' box.
- Controls:** To the right of the 'After correction' box, there is a 'Copy' button and a 'Lang' dropdown menu.
- Translate Button:** A wide, horizontal button labeled 'Translate' is positioned at the bottom center of the interface.

- **API 키 입력 박스 및 확인 버튼(추가 예정):** chatgpt에게 요청을 위한 필요한 API키를 사용자에게 입력 받아 값을 저장한다. (사용자는 API Key 발급 필요)

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



- **Auto Language Detection:** 번역하고자 하는 텍스트를 입력한다. 언어는 자동으로 번역을 실행할 때 자동으로 탐지할 수 있다.

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

}

- **Before Correction:** 초벌 번역된 텍스트를 출력하는 칸이다. Googletrans API로 자동으로 입력받은 언어를 탐지하고 사용자가 설정한 언어로 번역한 결과를 출력한다.

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'
```

- **After Correction:** 교정된 텍스트를 출력하는 칸이다. 교정된 텍스트 출력 박스에는 chatgpt에게 교정받은 결과를 출력한다.

#### 1. 모델 "text-davinci-003" 사용

```
openai.api_key = "PUT YOUR KEY"
model_engine = "text-davinci-003"
corrected_text = openai.Completion.create(
    engine=model_engine,
    prompt=(f"Insert the text"),
    temperature=0.5,
    max_tokens=1000,
    n = 1,
    stop=None,
)
```

#### 2. 모델 "gpt-3.5-turbo" 사용

- **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

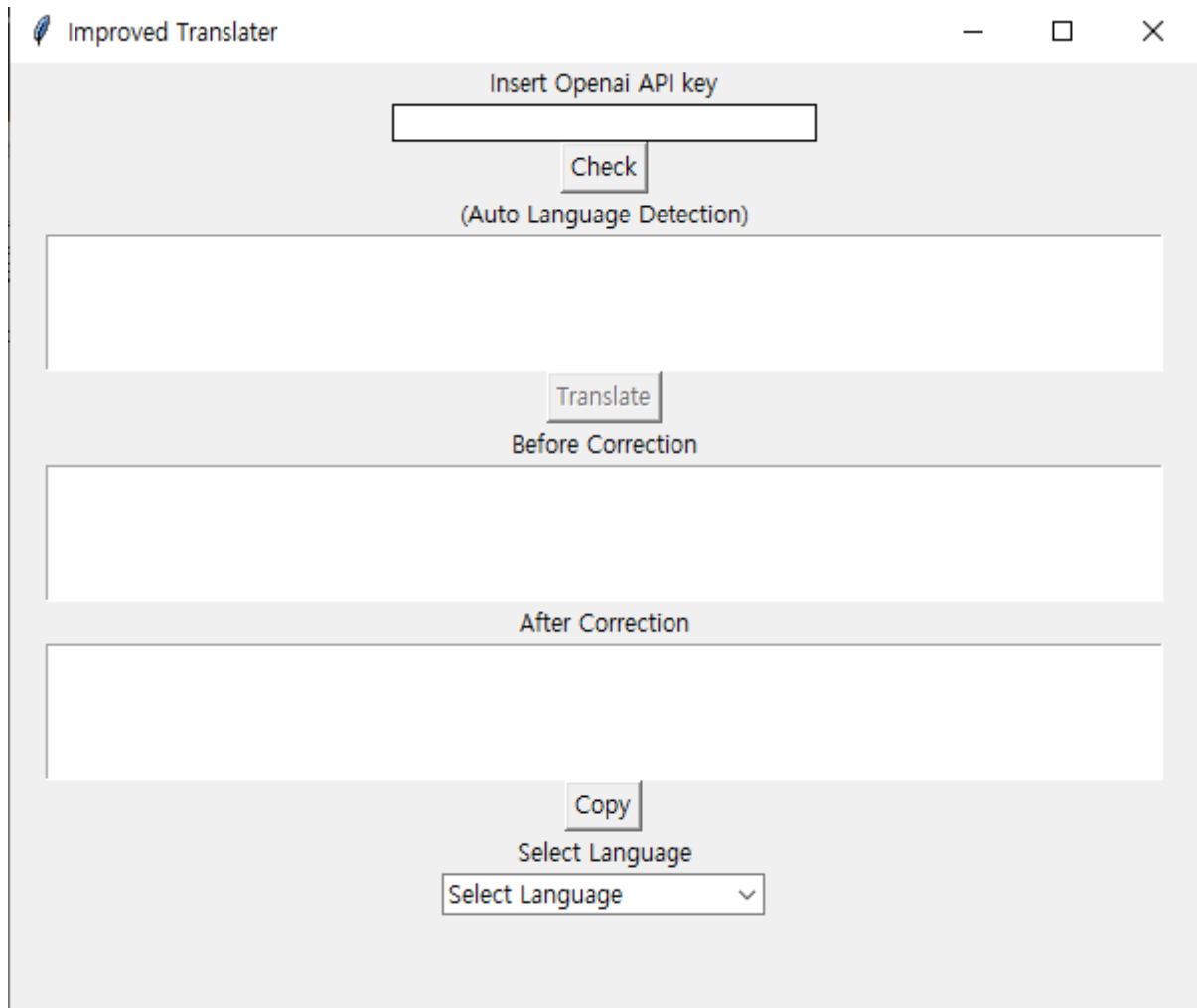
'response[\'choices\'][0][\'message\'][\'content\']'를 호출 후 응답결과를 'answer'에 저장한다음 출력한다.

- **Copy:** 교정된 텍스트 출력 박스의 출력 결과들을 복사하여 붙여넣기할 수 있는 상태로 만들어준다.

UI관련 패키지 PyQt의 함수 활용: QApplication.clipboard()

- **Translate:** 이 버튼을 누르면 번역 및 교정이 실행되며 진행 중일 때는 다른 버튼들은 비활성화 된다.

(간단히 만들어 놓은 UI)



## 2. Test Plan

사전지식이 필요한 문장이나 긴 글과 일반적으로 진입장벽이 낮은 글의 번역결과에 대해 비교하고 분석해보면서 교정의 성능이 어느 정도인지 확인하여 “향상된 번역 프로그램이 유의미한지”에 대해 test해볼 예정이다. (영문을 한글로 번역해볼 예정)

예문 1: Chow et. al의 “White-Box Cryptography and an AES Implementation”라는 논문의 Abstract (공백 포함: 1010자, 공백 제외: 872자)

<Conventional software implementations of cryptographic algorithms are totally insecure where a hostile user may control the execution environment, or where co-located with malicious software.

Yet current trends point to increasing usage in environments so threatened. We discuss encrypted-composed-function methods intended to provide a practical degree of protection against white-box (total access) attacks in untrusted execution environments. As an example, we show how aes can be implemented as a series of lookups in key-dependent tables. The intent is to hide the key by a combination of encoding its tables with random bijections representing compositions rather than individual steps, and extending the cryptographic boundary by pushing it out further into the containing application. We partially justify our aes implementation, and motivate its design, by showing how removal of parts of the recommended implementation allows specified attacks, including one utilizing a pattern in the aes SubBytes table.>

예문 2: 소설 '위대한 개츠비'의 첫 문단 (공백 포함: 271자, 공백 제외: 221자)

<In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. "Whenever you feel like criticizing anyone," he told me, "just remember that all the people in this world haven't had the advantages that you've had.">

또한 지금까지 작동하는지에 대한 test를 했을 때 text-davinci-003 모델을 사용했었는데, gpt-3.5-turbo 모델을 사용하면 1K 토큰당 0.002달러로 10배 저렴하기 때문에 chat 형식으로 태스크 변환을 하고 gpt-3.5-turbo 모델을 사용하는 test 또한 진행해 볼 예정이다.

마지막으로 test를 진행하는 동안 겪은 오류 과정이나 추가했으면 좋을 것 같다고 생각한 것들에 대해 기록할 것이며 내가 프로그램의 사용자가 되어 기존에 만든 데모 버전을 사용하면서 보완할 부분에 대해 찾아보는 test까지 진행해 볼 예정이다.