

OPEN AI

# 파파고(OPEN AI)를 이용한 동시 통역 프로그램 개발

배누리, 김호정, 전사영, 박현아

2024.12.09

# 목차

- ① 기획 및 구성
- ② 사용 기술
- ③ 코드 구현
- ④ 최종 결과



# 기획 및 구성

---



## Point

Papago를 활용하여 동시 통역 프로그램 개발 계획

그러나 Papago API 제공 서비스 종료에 따라 Open AI API로 대체

# 사용 기술



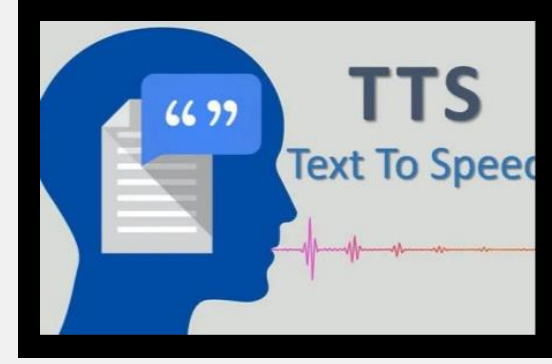
## ASR

입력된 음성을 Text로 변환



## Open AI

Open AI에서 제공하는 GPT 4.0를  
활용하여 번역된 응답 생성



## TTS

Text를 음성으로 변환

# 서비스 시나리오

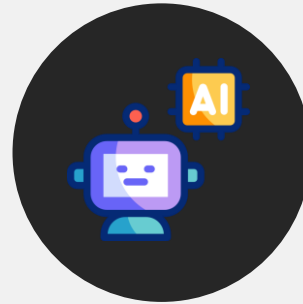
---



**사용자 음성 입력**



**텍스트 변환**



**AI 기반 번역**



**음성 변환**

# 코드 구현

```
import streamlit as st
import openai
import speech_recognition as sr
from gtts import gTTS
import os
import playsound
import threading

def select_language_ui():
    languages = {
        "한국어": "ko",
        "영어": "en",
        "일본어": "ja",
        "중국어": "zh",
        "독일어": "de",
        "스페인어": "es"
    }
    input_lang_name = st.selectbox("입력 언어를 선택하세요:", options=list(languages.keys()))
    target_lang_name = st.selectbox("번역 언어를 선택하세요:", options=list(languages.keys()))

    return languages[input_lang_name], languages[target_lang_name], input_lang_name, target_lang_name
```

# 코드 구현

---

```
def transcribe_speech_to_text(input_lang_code):  
    recognizer = sr.Recognizer()  
    with sr.Microphone() as source:  
        try:  
            audio = recognizer.listen(source, timeout=5)  
            text = recognizer.recognize_google(audio, language=input_lang_code)  
            return text  
        except sr.UnknownValueError:  
            st.error("음성을 인식하지 못했습니다. 다시 시도해주세요.")  
        except sr.RequestError as e:  
            st.error(f"Google Speech Recognition 서비스에 문제가 발생했습니다: {e}")  
    return None
```

# 코드 구현

---

```
def translate_text_with_openai(text, target_language="en"):
    openai.api_key = os.getenv("OPENAI_API_KEY")
    prompt = f"Translate the following text to {target_language}:\n{text}"
    try:
        response = openai.ChatCompletion.create(
            model="gpt-4.0",
            messages=[
                {"role": "system", "content": "You are a helpful translation assistant."},
                {"role": "user", "content": prompt}
            ],
            max_tokens=100
        )
        translation = response['choices'][0]['message']['content'].strip()
        return translation
    except Exception as e:
        st.error(f"번역 중 오류가 발생했습니다: {e}")
        return None
```



# 코드 구현

---

```
def text_to_speech(text, lang="en"):
    try:
        tts = gTTS(text=text, lang=lang)
        filename = "output.mp3"
        tts.save(filename)

        # 별도의 스레드에서 재생
        def play_audio():
            playsound.playsound(filename)
            os.remove(filename)

        thread = threading.Thread(target=play_audio)
        thread.start()
    except Exception as e:
        st.error(f"TTS 생성 중 오류가 발생했습니다: {e}")
```

# 코드 구현

```
def main():
    st.title("🌐 동시통역 프로그램")

    with st.sidebar:
        st.header("🌐 언어 설정")
        input_lang_code, target_lang_code, input_lang_name, target_lang_name = select_language_ui()

    st.markdown(f"""
<style>
.chat-container {{
    display: flex;
    flex-direction: column;
    gap: 10px;
}}
.chat-message {{
    border-radius: 10px;
    padding: 10px;
    max-width: 80%;
    margin-bottom: 10px;
}}
.user-message {{
    align-self: flex-end;
    background-color: #0078D7;
    color: #FFFFFF;
}}
.bot-message {{
    align-self: flex-start;
    background-color: #F1F0F0;
    color: #333;
}}
</style>
""", unsafe_allow_html=True)
    chat_messages = st.empty()
```

# 코드 구현

```
chat_messages = [] # 채팅 메시지 저장용 리스트

if st.button("말하기"):
    input_text = transcribe_speech_to_text(input_lang_code)
    if input_text:
        chat_messages.append({"type": "user", "content": input_text})
        translated_text = translate_text_with_openai(input_text, target_language=target_lang_code)
        if translated_text:
            chat_messages.append({"type": "bot", "content": translated_text})
            text_to_speech(translated_text, lang=target_lang_code)

# 채팅 메시지 렌더링
for msg in chat_messages:
    if msg["type"] == "user":
        st.markdown(f"""
            <div class="chat-container">
                <div class="chat-message user-message">
                    {msg["content"]}
                </div>
            </div>
            """, unsafe_allow_html=True)
    elif msg["type"] == "bot":
        st.markdown(f"""
            <div class="chat-container">
                <div class="chat-message bot-message">
                    {msg["content"]}
                </div>
            </div>
            """, unsafe_allow_html=True)

if __name__ == "__main__":
    main()
```

# 최종 결과

## 언어 설정

입력 언어를 선택하세요:

한국어

번역 언어를 선택하세요:

영어



## 동시통역 프로그램

말하기

**O  
P  
E  
N  
A  
I**

# Thank you

---

