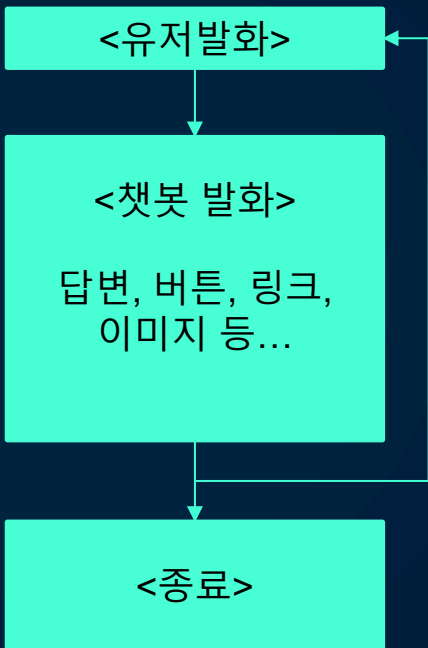
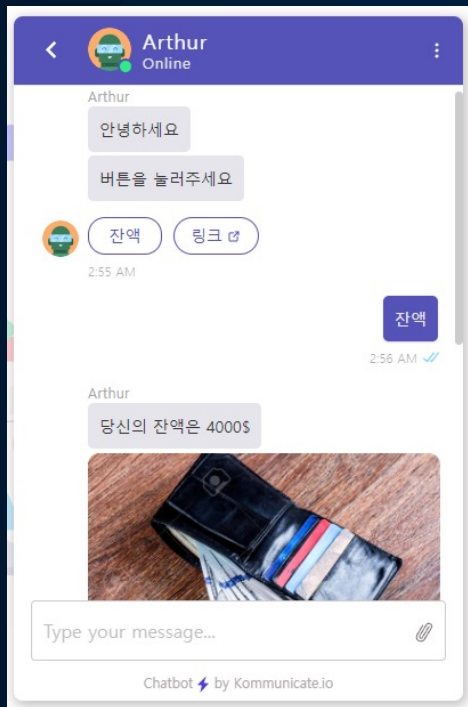




챗봇 빌더



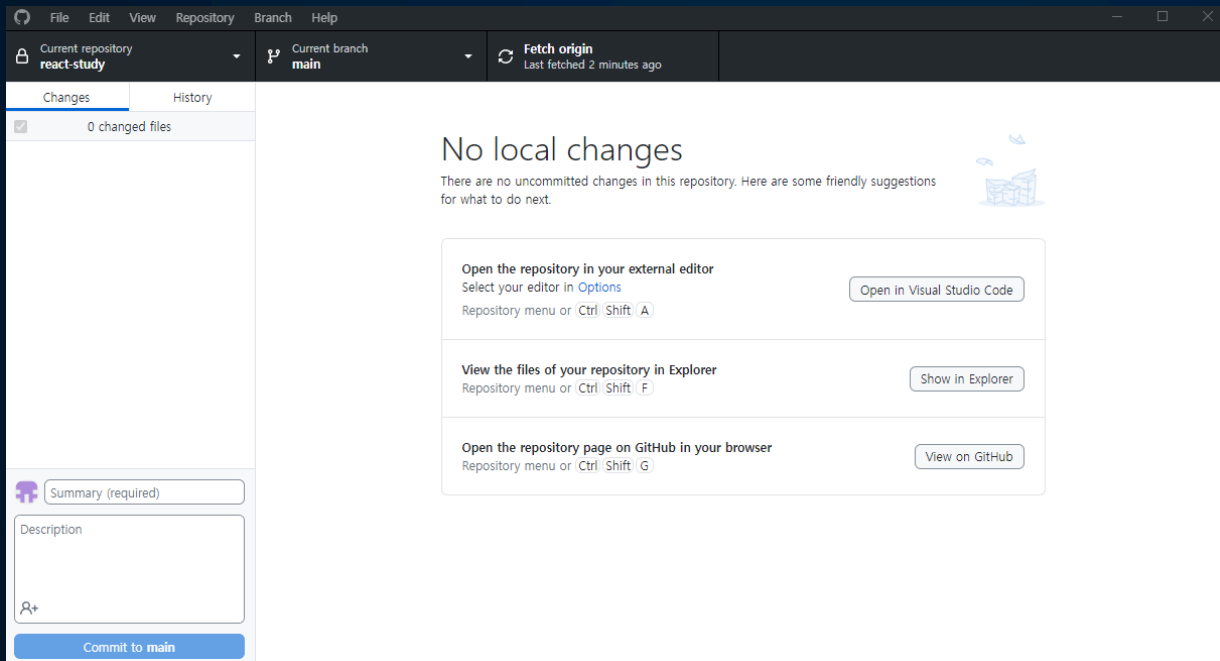
태스크 기반 챗봇

플로우 차트

챗봇 발화(Action)

Github repository 정리

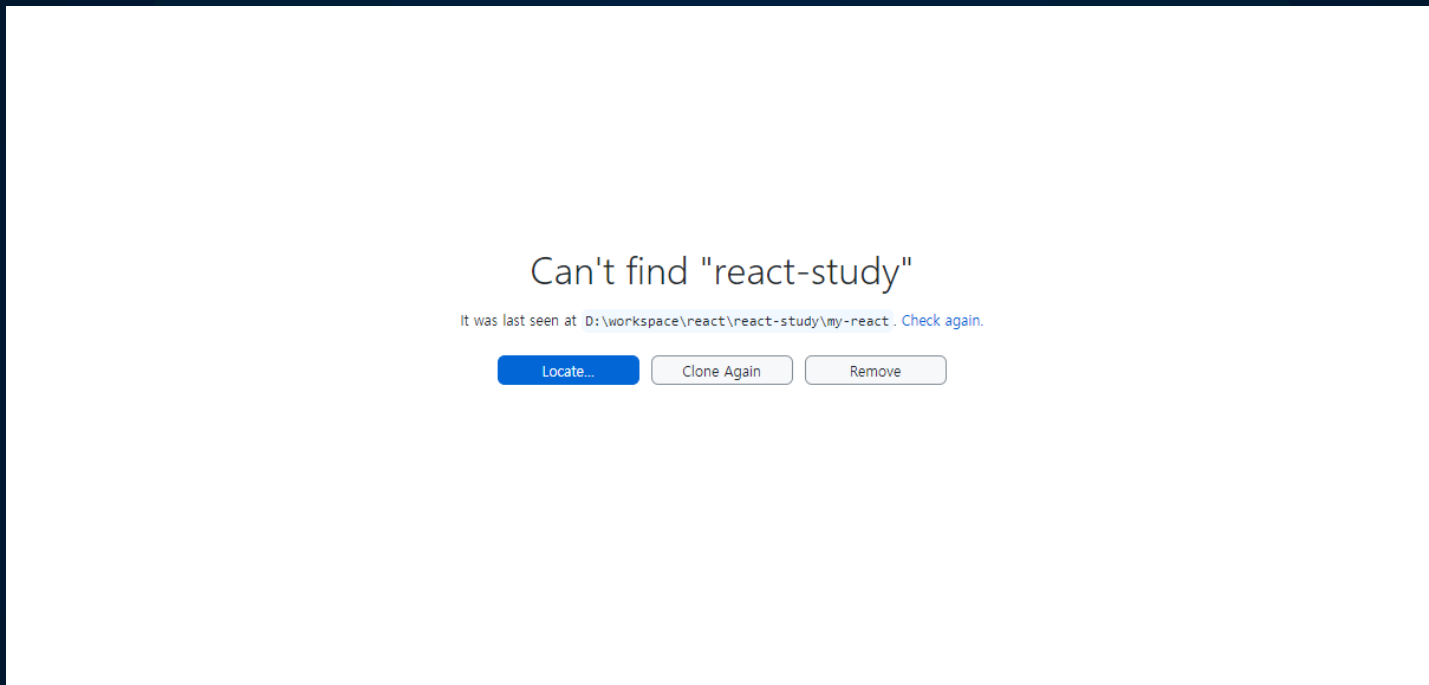
프로젝트 진행하면서 생성한 dataset, py, ipynb 등..
본인이 편한 경로에 새폴더 생성하여 정리하기



기존 repository폴더
.Git폴더 잘라내기

새롭게 정리한 폴더로
.Git폴더 붙여넣기

Github



.Git 폴더 이동 시, Can't find "repository이름" 오류 발생
Locate.. 클릭 후, 경로 .Git 폴더가 있는 폴더로 경로 재설정.
이후 활성화 될 시, Commit & push

Python 명령어 복습

1. jupyter notebook 실행
2. jupyter notebook에서 wellness_dataset 전처리 과정 후, csv파일로 내보내기
3. 작성했던 chatbot.py (streamlit 챗봇 프로그램) 실행 시켜보기

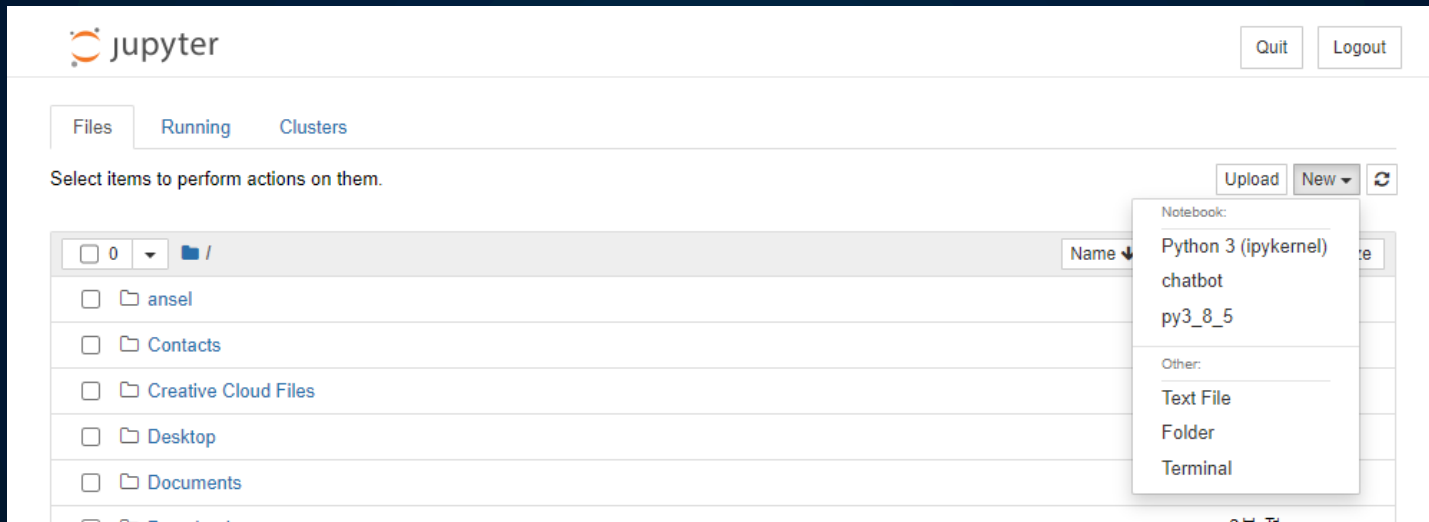
Python

anaconda prompt 실행

`conda activate chatbot` // chatbot 가상환경 활성화

`jupyter notebook` // jupyter notebook 실행

Python



dataset.ipynb

```
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
```


SentenceBERT 모델 로드

```
model = SentenceTransformer('jhgan/ko-sroberta-multitask')  
  
sentences = ["안녕하세요?", "한국어 문장 임베딩을 위한 버트 모델입니다."]   
embeddings = model.encode(sentences)  
  
print(embeddings)
```

```
[[-0.37510434 -0.7733841  0.5927711  ...  0.5792351  0.32683483  
  -0.6508966 ]  
 [-0.09361703 -0.18191542 -0.19230805 ... -0.03165817  0.30412528  
  -0.26793632]]
```

데이터셋 로드

웰니스 대화 스크립트 데이터셋

<https://aihub.or.kr/opendata/keti-data/recognition-laguage/KETI-02-006>

```
df = pd.read_csv('wellness_dataset_original.csv')  
df.head()
```

	구분	유저	챗봇	Unnamed: 3
0	감정/감정조절이상	제 감정이 이상해진 것 같아요. 남편만 보면 화가 치밀어 오르고 감정 조절이 안되요.	감정이 조절이 안 될 때만큼 힘들 때는 없는 거 같아요.	NaN
1	감정/감정조절이상	더 이상 내 감정을 내가 컨트롤 못 하겠어.	저도 그 기분 이해해요. 많이 힘드시죠?	NaN
2	감정/감정조절이상	하루종일 오르락내리락 롤러코스터 타는 기분이에요. 그럴 때는 밥은 잘 먹었는지, 잠은 잘 잤는지 체크해보는 것도 좋아요.		NaN
3	감정/감정조절이상	꼭 롤러코스터 타는 것 같아요.	NaN	NaN
4	감정/감정조절이상	롤러코스터 타는 것처럼 기분이 왔다 갔다 해요.	NaN	NaN

전처리

```
df = df.drop(columns=['Unnamed: 3'])  
df.head()
```

	구분	유저	챗봇
0	감정/감정조절이상	제 감정이 이상해진 것 같아요. 남편만 보면 화가 치밀어 오르고 감정 조절이 안되요.	감정이 조절이 안 될 때만큼 힘들 때는 없는 거 같아요.
1	감정/감정조절이상	더 이상 내 감정을 내가 컨트롤 못 하겠어.	저도 그 기분 이해해요. 많이 힘드시죠?
2	감정/감정조절이상	하루종일 오르락내리락 롤러코스터 타는 기분이에요. 그럴 때는 밥은 잘 먹었는지, 잠은 잘 잤는지 체크해보는 것도 좋아요.	
3	감정/감정조절이상	꼭 롤러코스터 타는 것 같아요.	NaN
4	감정/감정조절이상	롤러코스터 타는 것처럼 기분이 왔다 갔다 해요.	NaN

```
df = df[~df['챗봇'].isna()]
```

```
df.head()
```

	구분	유저	챗봇
0	감정/감정조절이상	제 감정이 이상해진 것 같아요. 남편만 보면 화가 치밀어 오르고 감정 조절이 안되요.	감정이 조절이 안 될 때만큼 힘들 때는 없는 거 같아요.
1	감정/감정조절이상	더 이상 내 감정을 내가 컨트롤 못 하겠어.	저도 그 기분 이해해요. 많이 힘드시죠?
2	감정/감정조절이상	하루종일 오르락내리락 롤러코스터 타는 기분이에요.	그럴 때는 밥은 잘 먹었는지, 잠은 잘 잤는지 체크해보는 것도 좋아요.
15	감정/감정조절이상/화	평소 다른 일을 할 때도 비슷해요. 생각한대로 안되면 화가 나고...그런 상황이 지속되...	화가 폭발할 것 같을 때는 그 자리를 피하는 것도 좋은 방법이라고 생각해요.
16	감정/감정조절이상/화	예전보다 화내는 게 과격해진 거 같아.	정말 힘드시겠어요. 화는 남에게도 스스로에게도 상처를 주잖아요.

유저 대화내용 인코딩

```
df['embedding'] = pd.Series([[]] * len(df)) # dummy
df['embedding'] = df['유저'].map(lambda x: list(model.encode(x)))
df.head()
```

	구분	유저	챗봇	embedding
0	감정/감정조절이 상	제 감정이 이상해진 것 같아요. 남편만 보면 화가 치밀어 오르고 감 정 조절이 안되요.	감정이 조절이 안 될 때만큼 힘을 때는 없는 거 같아요.	[-0.4806065, -0.29486918, 0.43789992, -0.64013...
1	감정/감정조절이 상	더 이상 내 감정을 내가 컨트롤 못 하겠어.	저도 그 기분 이해해요. 많이 힘드시죠?	[-1.1561579, -0.14506263, 0.29490355, -0.67394...
2	감정/감정조절이 상	하루종일 오르락내리락 롤러코스터 타는 기분이에요.	그럴 때는 밥은 잘 먹었는지, 잠은 잘 잤는지 체크해보는 것 도 좋아요.	[-0.6652002, -0.081268094, 1.0945568, 0.105792...
15	감정/감정조절이 상/화	평소 다른 일을 할 때도 비슷해요. 생각한대로 안되면 화가 나고...그 런 상황이 지속되...	화가 폭발할 것 같을 때는 그 자리를 피하는 것도 좋은 방법 이라고 생각해요.	[-0.76790583, 0.4652072, 0.52850723, -0.507604...
16	감정/감정조절이 상/화	예전보다 화내는 게 과격해진 거 같아.	정말 힘드시겠어요. 화는 남에게도 스스로에게도 상처를 주 잖아요.	[-0.20277739, -0.37413904, 0.04053206, -0.8624...

데이터 전체를 인코딩하기 때문에 시간이 오래 소요됨

```
df.to_csv('wellness_dataset.csv', index=False)
```

streamlit 실행

anaconda prompt(window) / terminal(mac)
cd (python 파일과 csv 데이터 파일이 있는 폴더로 경로 이동)

예시) (chatbot) C:\Users\Nextop(사용자이름)\chatbot

streamlit run chatbot.py 입력

최초 실행시 email 입력 필요 (이메일 인증 절차는 없음)

JSON 데이터 전처리

aihub 접속

도서자료 기계독해 샘플데이터 다운로드



#기계 독해 #질의응답 #AI 챗봇 #자연어 #지식정보

도서자료 기계독해

분야 한국어 유형 텍스트

갱신년월: 2023-04 구축년도: 2020 조회수: 2,646 다운로드: 798 용량: 104.82 MB

다운로드 ↓ 샘플 데이터 ?

JSON 데이터 전처리

```
In [1]: import json
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
```

```
In [2]: model = SentenceTransformer('jhgan/ko-sroberta-multitask')
```

```
In [6]: df = pd.read_json('tqa_sample.json')
df.head()
```

Out [6]:

	version	data
0	v0.1	{'time': '20201029154920', 'title': '생물다양성과 합성...
1	v0.1	{'time': '20201029154602', 'title': '해외 환경정책 인...
2	v0.1	{'time': '20201029155823', 'title': '범죄 빅데이터를 ...
3	v0.1	{'time': '20201029154348', 'title': '한국관광정책Kor...
4	v0.1	{'time': '20201029155011', 'title': '미국 중학교 제도...

data 안의 하위 데이터들이
전부 합쳐진 형태로 생성됨


```
{
  "version": "v0.1",
  "data": [
    {
      "time": "20201029154920",
      "title": "생물다양성과 합성생물학",
      "agency": "환경부 국립생물자원관, 세종대학교",
      "year": "2019",
      "content_id": "PCY_202003100356096481",
      "KDC": "00",
      "paragraphs": [
        {
          "context": "국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증거의 수집·검토 및 국가별 실질적 :
          "qas": [
            {
              "question": "국제사회는 무엇을 거쳐서 국제적 거버넌스를 구성하였지",
              "id": 427521,
              "is_impossible": false,
              "answers": [
                {
                  "text": "구조적 변화",
                  "answer_start": 257
                }
              ]
            },
            {
              "question": "무엇을 거쳐서 국제사회는 국제적 거버넌스를 조직하였어",
              "id": 427522,
              "is_impossible": false,
              "answers": [
                {
                  "text": "구조적 변화",
                  "answer_start": 257
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

sample.json

tqa_sample.json 구조

tqa_sample.json

data

time
title
agency
year
content_id
KDC

version

paragraphs

context

qas

question
id
is_impossible

answers

text
answer_start

json-parsing.ipynb

데이터가 너무 크기 때문에 시작하기 전에 데이터 일부 삭제하는 과정 필요

```
import json
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
```

```
model = SentenceTransformer('jhgan/ko-sroberta-multitask')
```

```
with open('tqa_sample.json', 'r', encoding='utf-8') as f:
    data = json.load(f)
```

json-parsing.ipynb

```
rows = []
for d in data['data']:
    for p in d['paragraphs']:
        for q in p['qas']:
            row = {
                'question': q['question'],
                'answer': q['answers'][0]['text'],
                'context': p['context']
            }
            rows.append(row)

df = pd.DataFrame(rows, columns=['question', 'answer', 'context'])

df.head()
```

`pd.DataFrame()`
Dataframe을 생성하는 함수

`rows`
각 행에 저장할 데이터
(question, answer, context)를
순서대로 저장한 리스트

`columns`
각 열의 이름을 지정

json-parsing.ipynb

	question	answer	context
0	국제사회는 무엇을 거쳐서 국제적 거버넌스를 구성하였지	구조적 변화	국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증...
1	무엇을 거쳐서 국제사회는 국제적 거버넌스를 조직하였어	구조적 변화	국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증...
2	국제사회가 국제적 거버넌스를 만들기 위해 거친 게 뭘까	구조적 변화	국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증...
3	대학교가 학생회를 만들기 위해 거친 게 뭘까	구조적 변화	국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증...
4	국제사회는 구조적 변화를 거쳐서 무엇을 구성했지	국제적 거버넌스	국제연합을 비롯하여 전문성을 갖춘 국제기구들이 설립되기 시작하면서 분야별 과학적 증...

해당 데이터 활용 예시

유저가 특정 단어에 대한 질문(question)을 말하면,
챗봇은 유저가 설명하는 단어의 정답(answer)을 말하고,
그 단어에 대한 의미(context)를 출력한다.

json-parsing.ipynb

앞서 진행했던 데이터셋과 동일하게
데이터프레임에 embedding을 추가한 후
json파일로 내보내기

```
df.to_json('tqa_dataset.json')
```

quiz-chatbot.py

```
import streamlit as st
from streamlit_chat import message
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
import json

@st.cache(allow_output_mutation=True)
def cached_model():
    model = SentenceTransformer('jhgan/ko-sroberta-multitask')
    return model

@st.cache(allow_output_mutation=True)
def get_dataset():
    df = pd.read_json('tqa_dataset.json')
    return df
```

quiz-chatbot.py

```
model = cached_model()
df = get_dataset()

st.header('도서 퀴즈 챗봇')

if 'generated' not in st.session_state:
    st.session_state['generated'] = []

if 'past' not in st.session_state:
    st.session_state['past'] = []

if 'context' not in st.session_state:
    st.session_state['context'] = []
```


quiz-chatbot.py

```
with st.form('form', clear_on_submit=True):  
    user_input = st.text_input('당신: ', '')  
    submitted = st.form_submit_button('전송')  
  
if submitted and user_input:  
    embedding = model.encode(user_input)  
  
    df['distance'] = df['embedding'].map(lambda x:  
cosine_similarity([embedding], [x]).squeeze())  
    answer = df.loc[df['distance'].idxmax()]  
  
    st.session_state.past.append(user_input)  
    st.session_state.generated.append(answer['answer'])  
    st.session_state.context.append(answer['context'])
```

quiz-chatbot.py

```
for i in range(len(st.session_state['past'])):
    message(st.session_state['past'][i], is_user=True, key=str(i)
+ '_user')
    if len(st.session_state['generated']) > i:
        message(st.session_state['generated'][i], key=str(i) +
'_bot')
        message(st.session_state['context'][i], key=str(i) +
'_botContext')
```

작성 후, streamlit run quiz-chatbot.py 실행

데이터를 많이 잘라냈기 때문에
답변의 정확도가 높지 않음.

도서 퀴즈 챗봇

당신:

전송

안녕하세요



유아교육·보육기관 대상 평가자



뉴질랜드에서는 유아교육·보육기관 대상 평가자가 유아교육·보육 전공자일 필요가 없으며 평가자는 모든 학교와 유아교육기관의 질을 평가하고 있다. 같은 사람이 여러 수준의 학교를 평가하기 때문에 개인마다 특화된 학교수준을 결정하는 것도 필요하다고 본다. 평가자에게 요구되는 행동방침은 정당함(fair), 공평함(impartial), 책임감(responsible), 신뢰로움(trustworthy)이다. 또한, 평가자의 역량으로 분석과 판단, 의사소통, 관계성, 평가팀과의 조정, 윤리적 행동, 결과지향성, 전문성개발 및 지식, 평가팀에의 참여를 제시하고 있으며 이러한 역량의 중앙에는 윤리적인 실천이 있으며 리더쉽과 He Toa Takitini(여러 사람의 공헌)가 바탕에 깔려있다.

python으로 구현한 챗봇 텔레그램 연동

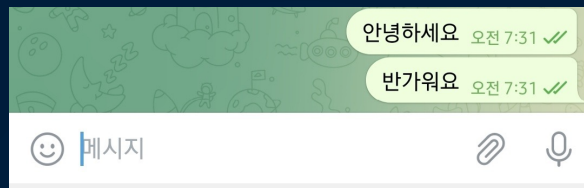
기존에 만들었던 텔레그램 봇 재사용(없을 경우 새로 생성)

bot 토큰 필요

<https://api.telegram.org/bot{봇토큰}/getUpdates>

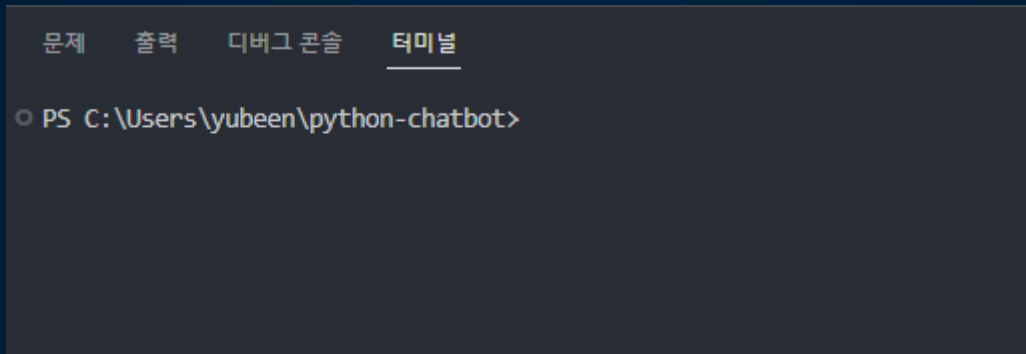
ex) <https://api.telegram.org/bot6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI/getUpdates>

```
{"ok":true,"result":[]}
```

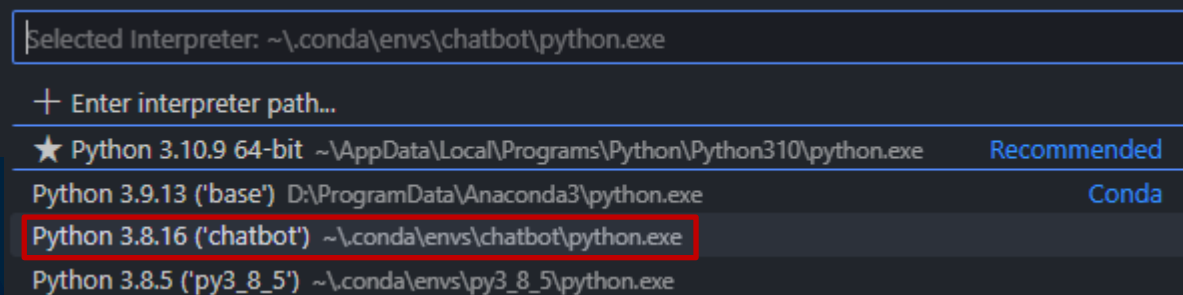
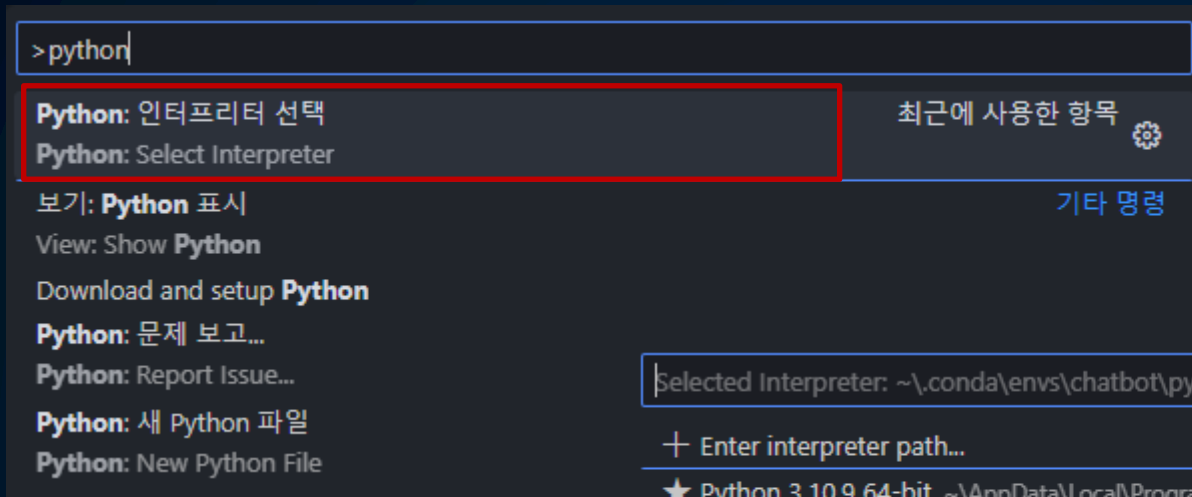


```
{
  "ok": true,
  "result": [
    {
      "update_id": 28530052,
      "message": {
        "message_id": 103,
        "from": {
          "id": 6095113280,
          "is_bot": false,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "language_code": "ko"
        },
        "chat": {
          "id": 6095113280,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "type": "private"
        },
        "date": 1682202658,
        "text": "\uc57c\uc548\ub155"
      },
      "update_id": 28530053,
      "message": {
        "message_id": 104,
        "from": {
          "id": 6095113280,
          "is_bot": false,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "language_code": "ko"
        },
        "chat": {
          "id": 6095113280,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "type": "private"
        },
        "date": 1682202659,
        "text": "\uc548\ub155"
      },
      "update_id": 28530054,
      "message": {
        "message_id": 105,
        "from": {
          "id": 6095113280,
          "is_bot": false,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "language_code": "ko"
        },
        "chat": {
          "id": 6095113280,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "type": "private"
        },
        "date": 1682202660,
        "text": "\ud558\uc774"
      },
      "update_id": 28530055,
      "message": {
        "message_id": 106,
        "from": {
          "id": 6095113280,
          "is_bot": false,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "language_code": "ko"
        },
        "chat": {
          "id": 6095113280,
          "first_name": "\uc720\uube48",
          "last_name": "\uc774",
          "type": "private"
        },
        "date": 1682202661,
        "text": "\ubb50\ud568"
      }
    ]
  ]
}
```

visual studio code anaconda연결



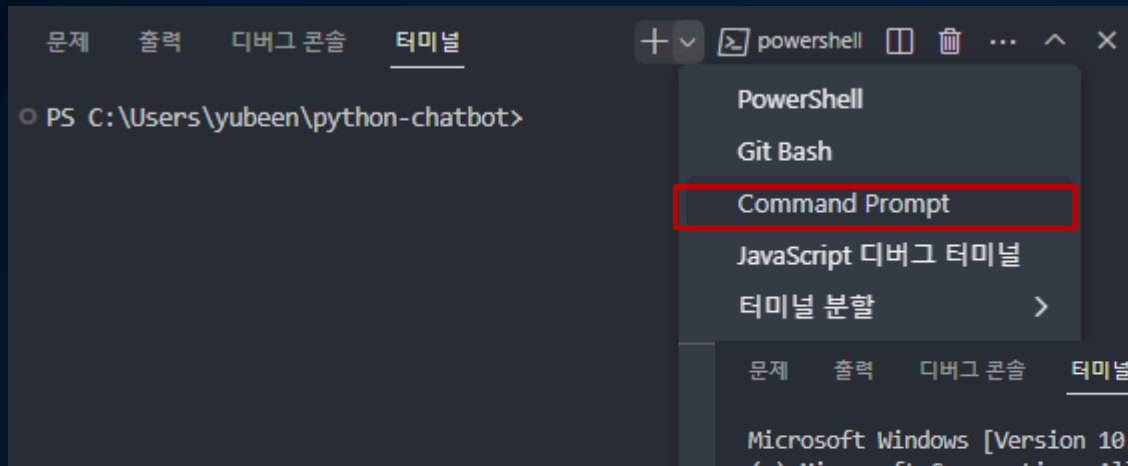
vscode 기본 터미널
아나콘다 가상환경과 연결이 되어있지 않음.



ctrl + shift + p 입력,

python: select interpreter 선택

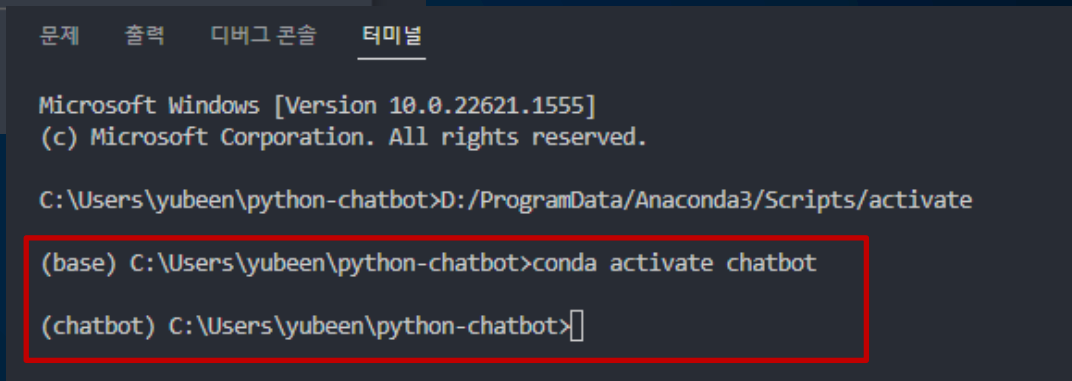
그 후 chatbot 가상환경 선택



하단의 터미널 창

+ 버튼 옆 아래 화살표 클릭,

Command Prompt 선택



클릭 후, 터미널을

conda prompt처럼 이용가능


```
(chatbot) C:\Users\yubeen\python-chatbot>pip install python-telegram-bot
```

하단의 터미널 창

+ 버튼 옆 아래 화살표 클릭,

Command Prompt 선택

telegram.py

```
import requests

token = '6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI'
chat_id = 6095113280
bot_message = "hello"

url =
https://api.telegram.org/bot{token}/sendMessage?chat\_id={chat\_id}&text={bot\_message}

requests.get(url)
```

```
(chatbot) C:\Users\yubeen\python-chatbot>python telegram.py
```

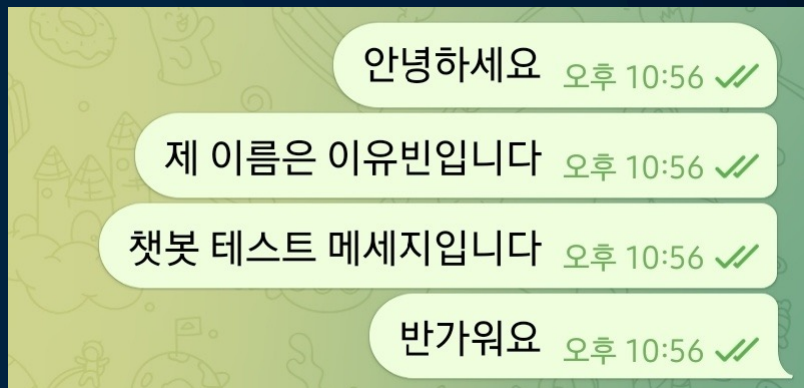
```
https://api.telegram.org/botbot6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI/sendMessage  
?chat_id=6095113280&text=hello
```

1. python telegram.py 입력 (코드 실행)
실행 시 텔레그램 봇 채널에서 “hello” 메시지 전송
2. print(url)로 해당 api 주소 출력
(ctrl + 클릭으로 해당 주소 이동)
주소 접속 시 동일하게 봇 채널에서 “hello” 메시지 전송

텔레그램 api로 json 데이터 추출

getUpdates 요청을 통해
새로운 메시지 정보 받아오기

텔레그램 봇에게 메시지 전송



get-json.py

```
import requests
import json

token = '6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI'
chat_id = 6095113280
bot_message = "hello"

url = f"https://api.telegram.org/bot{token}/getUpdates"
data = requests.get(url).json()

print(url)

with open('chat.json', 'w', encoding='utf-8') as file:
    json.dump(data, file, indent="\t", ensure_ascii=False)
```

```
(chatbot) C:\Users\yubeen\python-chatbot>python get-json.py
```

```
https://api.telegram.org/bot6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI/getUpdates
```

1. python get-json.py 입력 (코드 실행)
유저가 봇 채널에 입력한 메시지의 정보를
해당 폴더에 json파일로 저장
2. print(url)로 해당 api 주소 출력
(ctrl + 클릭으로 해당 주소 이동)
유저가 입력한 메시지에 대한 정보를 json 형식으로
보여주는 페이지로 이동

```

1  {
2      "ok": true,
3      "result": [
4          {
5              "update_id": 28530064,
6              "message": {
7                  "message_id": 133,
8                  "from": {
9                      "id": 6095113280,
10                     "is_bot": false,
11                     "first_name": "유빈",
12                     "last_name": "이",
13                     "language_code": "ko"
14                 },
15                 "chat": {
16                     "id": 6095113280,
17                     "first_name": "유빈",
18                     "last_name": "이",
19                     "type": "private"
20                 },
21                 "date": 1682776593,
22                 "text": "안녕하세요"
23             }
24         },
25         {
26             "update_id": 28530065,
27             "message": {

```

```

{"ok":true,"result":[{"update_id":28530064,
"message":{"message_id":133,"from":
{"id":6095113280,"is_bot":false,"first_name":"\uc720\ucbe48","last_name":"\uc774","lan
guage_code":"ko"},"chat":
{"id":6095113280,"first_name":"\uc720\ucbe48","last_name":"\uc774","type":"private"},"
date":1682776593,"text":"\uc548\ub155\ud558\uc138\uc694"}},{ "update_id":28530065,
"message":{"message_id":134,"from":
{"id":6095113280,"is_bot":false,"first_name":"\uc720\ucbe48","last_name":"\uc774","lan
guage_code":"ko"},"chat":
{"id":6095113280,"first_name":"\uc720\ucbe48","last_name":"\uc774","type":"private"},"
date":1682776597,"text":"\uc81c \uc774\ub984\uc740
\uc774\uc720\ucbe48\uc785\ub2c8\ub2e4"}},{ "update_id":28530066,
"message":{"message_id":135,"from":
{"id":6095113280,"is_bot":false,"first_name":"\uc720\ucbe48","last_name":"\uc774","lan
guage_code":"ko"},"chat":
{"id":6095113280,"first_name":"\uc720\ucbe48","last_name":"\uc774","type":"private"},"
date":1682776602,"text":"\ucc57\ubd07 \ud14c\uc2a4\ud2b8
\uba54\uc138\uc9c0\uc785\ub2c8\ub2e4"}},{ "update_id":28530067,
"message":{"message_id":136,"from":
{"id":6095113280,"is_bot":false,"first_name":"\uc720\ucbe48","last_name":"\uc774","lan
guage_code":"ko"},"chat":
{"id":6095113280,"first_name":"\uc720\ucbe48","last_name":"\uc774","type":"private"},"
date":1682776604,"text":"\ubc18\uac00\uc6cc\uc694"}]}}
```

```

1  {
2    "ok": true,
3    "result": [
4      {
5        "update_id": 28530064,
6        "message": {
7          "message_id": 133,
8          "from": {
9            "id": 6095113280,
10           "is_bot": false,
11           "first_name": "유빈",
12           "last_name": "이",
13           "language_code": "ko"
14         },
15         "chat": {
16           "id": 6095113280,
17           "first_name": "유빈",
18           "last_name": "이",
19           "type": "private"
20         },
21         "date": 1682776593,
22         "text": "안녕하세요"
23       },
24     ],
25     {
26       "update_id": 28530065,
27       "message": {

```

JSON

OK

result

update_id

message

message_id
form {...}
chat {...}
date
text

update_id

message

message_id
form {...}
chat {...}
date
text

chatbot-telegram.py

```
import requests
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
import json

token = "6148368452:AAF-w4V91NLqst87DrBaJmMxK80M4p62HNI"
chat_id = 6095113280
```

필요한 라이브러리 import

```
def cached_model():  
    model = SentenceTransformer('jhgan/ko-sroberta-multitask')  
    return model  
  
def get_dataset():  
    df = pd.read_csv('wellness_dataset.csv')  
    df['embedding'] = df['embedding'].apply(json.loads)  
    return df  
  
model = cached_model()  
df = get_dataset()
```

언어처리 모델과 챗봇 dataset 캐싱

```
last_update_id = None

while True:
    url = f"https://api.telegram.org/bot{token}/getUpdates"
    if last_update_id:
        url += f"?offset={last_update_id + 1}"
    response = requests.get(url)
    if response.status_code != 200:
        continue

    data = response.json()
```

getUpdates로 유저가 입력한 대화 정보를 받아온다.
last_update_id = 마지막 메시지의 id를 저장

while문 안에 이어서 계속 작성.

```
for update in data["result"]:
    text = update["message"]["text"]
    chat_id = update["message"]["chat"]["id"]
    update_id = update["update_id"]

    last_update_id = update_id

    embedding = model.encode(text)

    df['distance'] = df['embedding'].map(lambda x:
cosine_similarity([embedding], [x]).squeeze())
    answer = df.loc[df['distance'].idxmax()]
    bot_message = answer['챗봇']
```

json data에서 text, chat_id, update_id 추출.

하단 코드는 streamlit챗봇 코드에서 사용했던 코드의 재사용

while문 안에 이어서 계속 작성.

```
print(answer['distance'])
print(answer['유저'])
print(answer['챗봇'])

url =
f"https://api.telegram.org/bot{token}/sendMessage?chat_id={chat_id}&text={bot_message}"

requests.get(url)
```

코드 작성 후
python chatbot-telegram.py 실행

```
(chatbot) C:\Users\yubeen\python-chatbot>python chatbot-telegram.py
```

0.4965766004177441

어휴, 그런 말은 나도 하겠네.

기분이 풀릴 때까지 실컷 욕해도 좋아요. 제가 들어드릴게요.

0.8991264861037545

요즘 너무 피곤해요.

피곤하면 정말 아무 것도 하기 어렵죠. 잠깐 휴식을 취하는 건 어떨까요?

0.851989148353177

진짜 어이없고 화가 나서 숨이 턱 막히는 거야.

화를 쏟아내고 나면 기분이 조금 풀릴 거예요.

0.6422633553449788

너무 외로워요.

저는 너무 외로울 때면 말랑한 무언가를 만지고는 해요. 단순한 행동에 집중

□

안녕하세요 오후 11:40 ✓✓

기분이 풀릴 때까지 실컷 욕해도 좋아요. 제가 들어드릴게요.

오후 11:40

오늘도 너무 피곤해요 오후 11:40 ✓✓

피곤하면 정말 아무 것도 하기 어렵죠. 잠깐 휴식을 취하는 건 어떨까요?

오후 11:40

너무 화가나요 오후 11:40 ✓✓

화를 쏟아내고 나면 기분이 조금 풀릴 거예요.

오후 11:40

지금은 슬퍼요 오후 11:40 ✓✓

저는 너무 외로울 때면 말랑한 무언가를 만지고는 해요. 단순한 행동에 집중하면 기분이 약간은 나아질 거예요.

오후 11:40

😊 메시지



번외) 텔레그램 봇 명령어 추가

```
(chatbot) C:\Users\yubeen\python-chatbot>pip install python-telegram-bot
```

pip install python-telegram-bot 설치

```
{'ok': True, 'result': [
  {'update_id': 28530020, 'message': {'message_id': 14, 'from': {'id': 6095113280, 'is_bot': False, 'first_name': '유빈', 'last_name': '이', 'language_code': 'ko'}, 'chat': {'id': 6095113280, 'first_name': '유빈', 'last_name': '이', 'type': 'private'}, 'date': 1682135197, 'text': '안녕하세요'},
  {'update_id': 28530021, 'message': {'message_id': 15, 'from': {'id': 6095113280, 'is_bot': False, 'first_name': '유빈', 'last_name': '이', 'language_code': 'ko'}, 'chat': {'id': 6095113280, 'first_name': '유빈', 'last_name': '이', 'type': 'private'}, 'date': 1682135198, 'text': '반갑습니다'}
],
```

텔레그램 api의 getUpdates를 통해 가져온 JSON 파일에서 'text' 항목 추출 필요

streamlit으로 구현했던 심리상담 봇의 코드를 일부 수정하는 것으로 쉽게 연동이 가능.