

رایانش ابری

گزارش کار تمرین اول

در این تمرین از ۴ فایل پایتون یک فایل sql استفاده شده است که در ادامه به شرح کد میپردازیم.

قسمت اول languageTranslator.py

این قسمت از کد وظیفه ترجمه را بر عهده دارد

```
"""
Created on Tue Feb 22 19:23:29 2022

@author: mcpain79
"""
from ibm_watson import LanguageTranslatorV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

# Set some variables
key = 'NjjZDLxqk01LKKkGtJHr40pZ0huqb6HOS33SUN3RyAp'
url = 'https://api.us-south.language-translator.watson.cloud.ibm.com/instances/e61dd886-e748-445d-b243-99be310ea020'
def translate(model,text):
    #connecting to server with url and key
    authenticator = IAMAuthenticator(key)
    language_translator = LanguageTranslatorV3(version='2018-05-01',authenticator=authenticator)
    language_translator.set_service_url(url )
    #translating language
    translation = language_translator.translate(text=text,model_id=model).get_result()
    return translation['translations'][0]['translation']
```

در ابتدا کتابخانه های مورد نیاز را import میکنیم و همچنین مقادیر key و url را در متغیر های مربوطه خودشان ذخیره میکنیم.وظیفه تابع translate ترجمه یک متن با استفاده از مدلی است که ما برای آن مشخص میکنیم.

و هر مدل قابلیت ترجمه از یک زبان به یک زبان دیگر را فراهم میکند.حال باید با استفاده از key و url کارهای مربوط به authentication را انجام دهیم در نهایت ترجمه متن مورد نظر را برمیگردانیم.

قسمت دوم nlu.py

در اینجا ما با استفاده از پردازش زبان طبیعی تشخیص میدهیم که آیا کامنت شامل خشونت(anger) میباشد یا خیر

```

"""
Created on Wed Feb 23 16:57:38 2022

@author: mcaptain79
"""
from ibm_watson import NaturalLanguageUnderstandingV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibm_watson.natural_language_understanding_v1 import Features, EmotionOptions
key = 'ZdRc53L_4nsV1LOFMW3_Gkb4ASGYCx6nbLk6-4yUcPd'
url = 'https://api.us-south.natural-language-understanding.watson.cloud.ibm.com/instances/5e446778-d618-4e90-9fd5-8762e37e9ccf'
def angerDetector(text):
    #authentication using your id and url
    authenticator = IAMAuthenticator(key)
    natural_language_understanding = NaturalLanguageUnderstandingV1(version='2021-03-25',authenticator=authenticator)
    natural_language_understanding.set_service_url(url)
    #recieving response
    #keywords return important keywords in text
    response = natural_language_understanding.analyze(text=text,
        features=Features(emotion = EmotionOptions())).get_result()
    emotionDict = response['emotion']['document']['emotion']
    if emotionDict['anger'] == max(emotionDict.values()):
        return True
    return False

```

در صورتی که کامنت شامل anger باشد مقدار True را برمیگردانیم.

قسمت سوم speechToText.py

در این بخش دو تابع voice_recorder و speech2text را تعریف میکنیم که وظیفه تابع اول ذخیره یک وویس در قالب wav در داخل فایل voice.wav میباشد و تابع speech2text وظیفه تبدیل وویس به متن را برعهده دارد.

```

"""
Created on Thu Feb 24 19:45:29 2022

@author: mcaptain79
"""
import sounddevice as sd
from scipy.io.wavfile import write
from ibm_watson import SpeechToTextV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
key = 'iNexc7JRWowt1rJKk4JrBYYo5HUeTo3z11_Qe5szbJSp'
url = 'https://api.us-south.speech-to-text.watson.cloud.ibm.com/instances/96098b92-b92a-4e1c-98b7-31a3366d57c8'
#function below is for recording the voice of the user and user just have
def voice_recorder():
    # Sampling frequency
    freq = 44100
    # Recording duration
    duration = 10
    # Start recorder with the given values of
    # duration and sample frequency
    recording = sd.rec(int(duration * freq), samplerate=freq, channels=2)
    #Record audio for the given number of seconds
    sd.wait()
    # This will convert the NumPy array to an audio
    # file with the given sampling frequency
    write("voice.wav", freq, recording)
def speech2text():
    authenticator = IAMAuthenticator(key)
    s2t = SpeechToTextV1(authenticator = authenticator)
    s2t.set_service_url(url)
    voice = open('voice.wav', 'rb')
    result = s2t.recognize(audio = voice, content_type = 'audio/wav').get_result()
    #for any silence it saves text in new array form
    fullText = ''
    for i in range(len(result['results'])):
        if result['results'][i]['alternatives'][0]['transcript'] != '%HESITATION':
            fullText += result['results'][i]['alternatives'][0]['transcript']
    return fullText

```

در تابع speech2text به ازای هر وقفه و سکوتی که داریم هر قسمت را در یک آرایه جداگانه قرار میدهد برای همین به یک حلقه for برای ساخت کامل کامنت نیاز داریم. در نهایت نیز متن کامل را برمیگرداند.

فایل backend.py

```

"""
Created on Sun Feb 20 16:44:52 2022

@author: mcaptain79
"""
import mysql.connector
from languageTranslator import translate
from nlu import angerDetector
from speechToText import voice_recorder, speech2text
myDB = mysql.connector.connect(user = 'root', password = None, host = 'localhost', database = 'mySql')

```

در ابتدا از بقیه فایل ها توابع مورد نظر را import میکنیم و همچنین از تابع mysql.connector برای اتصال به پایگاه داده استفاده میکنیم.

بعد از آن به پایگاه داده متصل میشویم و عملیات های مورد نظر را انجام میدهیم

```

myDB = mysql.connector.connect(user = 'root',password = None,host = 'localhost',database = 'mySql')
cursor = myDB.cursor()
#showin user pannel in a loop
while True:
    #choosing to see or post a comment
    choice = input('1)post a comment\n2)see comments\nchoose:')

```

```

if choice == '1':
    choice2 = input('enter movie id: ')
    choice3 = input('1)write comment\n2)record a voice\nchoose: ')
    if choice3 == '1':
        text = input('enter comment: ')
        if angerDetector(text):
            print('sorry your comment violates our terms:')
        else:
            query1 = 'select count(*) from comment'
            cursor.execute(query1)
            commentId = 0
            for i in cursor:
                commentId = i[0]
            query2 = 'insert into comment (comment_id,movie_id,comment_text) values (%s,%s,%s)'
            cursor.execute(query2,(int(commentId)+1,int(choice2),text))
            myDB.commit()
            print('comment posted successfully.thanks for your comment')
    elif choice3 == '2':
        print('you just have 10 seconds:')
        print('recording voice...')
        voice_recorder()
        print('voice recorded')
        print('converting voice to text please be patient')
        text = speech2text()
        print('your comment:',text)
        if angerDetector(text):
            print('sorry your comment violates our terms:')
        else:
            query1 = 'select count(*) from comment'
            cursor.execute(query1)
            commentId = 0
            for i in cursor:
                commentId = i[0]
            query2 = 'insert into comment (comment_id,movie_id,comment_text) values (%s,%s,%s)'
            cursor.execute(query2,(int(commentId)+1,int(choice2),text))
            myDB.commit()
            print('comment posted successfully.thanks for your comment')
    else:
        print('bad input')

```

```

else:
    comments = []
    choice2 = input('enter your movie id: ')
    query = 'select comment_text from comment where movie_id = %s'
    cursor.execute(query,(choice2,))
    for i in cursor:
        comments.append(i[0])
    choice3 = input('1)english\n2)japanese\n3)arabic\nchoose: ')
    if choice3 == '1':
        for i in comments:
            print(i)
    elif choice3 == '2':
        for i in comments:
            print(translate('en-ja', i))
    elif choice3 == '3':
        for i in comments:
            print(translate('en-ar', i))
    else:
        print('bad input')

```

در ابتدا از کاربر خواسته میشود که بین کامنت دادن و دیدن کامنت ها انتخاب کند. اگر کاربر کامنت دادن را انتخاب کند با وارد کردن آیدی فیلم میتواند انتخاب کند که کامنت را متنی بنویسد یا به صورت وویس که بعد از وارد کردن کامنت از تابع تشخیص خشونت استفاده میکنیم. همچنین کاربر میتواند کامنت ها را به ۳ زبان انگلیسی، ژاپنی و عربی مشاهده کند.

قسمت چهارم myDataBase.sql

شامل دو جدول فیلم ها و کامنت ها میباشد

```
create table movie(  
    movie_id int,  
    movie_name varchar(100),  
    movie_poster_link text,  
    movie_director varchar(100),  
    primary key (movie_id)  
);  
  
create table comment(  
    comment_id int,  
    movie_id int,  
    comment_text text,  
    primary key (comment_id),  
    foreign key (movie_id) references movie(movie_id)  
);
```

۵ تا فیلم به جدول movie اضافه کردم

```
-- inserting movies  
insert into movie values (1,'iron man 3','https://drive.google.com/file/d/1eXHUGKmlzqgBk3_tK88yJ-7UxxQocPy0/view?usp=sharing',  
, 'shane black');  
insert into movie values (2,'fifty shades of grey','https://drive.google.com/file/d/1ngPBEYtNuIuDDJ7ZjIWUykdHGrpRkErh/view?usp=sharing',  
, 'james foley');  
insert into movie values (3,'the avengers','https://drive.google.com/file/d/1Ezw6RjXE0bLy4Tt00_DCN8c63Y6CPYU5/view?usp=sharing',  
, 'Joss Whedon');  
insert into movie values (4,'tenet','https://drive.google.com/file/d/1FmCNSFFdRiCT4Gu4zUkClCFMUiQgdhS5/view?usp=sharing',  
, 'Christopher Nolan');  
insert into movie values (5,'joker','https://drive.google.com/file/d/1TBQNLmhk3f6YcYzvHGsYz5q6CciE_M9/view?usp=sharing',  
, 'Todd Phillips');
```