MVP：
The app is able to detect people’s sentiment related to keyword in Twitter. It allows the users to determine the keyword and how many tweets they want to analyze. It is able to see the tweet text analyzed and the sentiment analysis result showing sentiment score and sentiment magnitude. It is also able to compare two keywords to see people’s sentiment to them in Twitter.

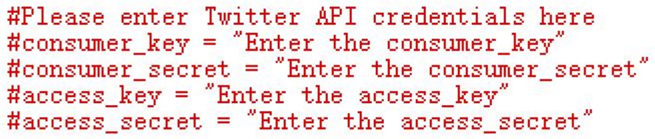
User Stories:
As a marketer, I want to analyze the feedback to the product from people.
As a marketer, I want to adjust the keyword of the product to derive more details and determine the number of tweets that requires analyzing.
As a marketer, I want to compare people’s sentiment to two competing products.
As a reporter, I want to analyze people’s sentiment to certain object or certain well-known person.
As a reporter, I want to compare people’s sentiment to two famous persons.

Users:
Marketer, reporter, analyst, anyone who wants to analyze the sentiment to something in Twitter

Translate user stories to modular design:
(1)Input module:
I want to adjust the keyword to derive more details and determine the number of tweets that requires analyzing.
(2)TwitterAPI module:
I want to derive the text from Twitter according to the input keyword and number
(3)NLPAPI module:
I want to analyze the sentiment to the text.
(4)output module:
I want to see the result of sentiment analysis to one keyword or two keywords compared.

TwitterAPI module is implemented in TwitterAPI.py, NLPAPI module is implemented in NLPAPI.py, input module and output module are implemented in main.py and compare.py(outputting different analysis)

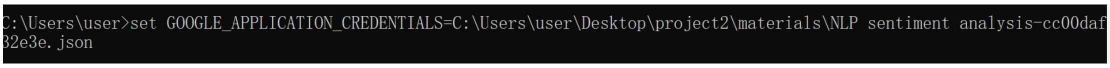
Implemention:
(1) Enter your keys for Twitter API.
Under TwitterAPI.py where



```
#Please enter Twitter API credentials here
#consumer_key = "Enter the consumer_key"
#consumer_secret = "Enter the consumer_secret"
#access_key = "Enter the access_key"
#access_secret = "Enter the access_secret"
```

Please enter your keys for Twitter API.
Due to the secret requirement from Twitter API, I could not provide my keys in the github.

(2) Use the key file for Google NLP API
Google Natural Language API requires adding .json file to the path and hold the same cmd shell when running python files.
Use the google nlp key file in the environment, run under cmd:
set GOOGLE_APPLICATION_CREDENTIALS=C:\Users\user\Desktop\project2\materials\NLP sentiment analysis-cc00daf32e3e.json



```
:\Users\user>set GOOGLE_APPLICATION_CREDENTIALS=C:\Users\user\Desktop\project2\materials\NLP sentiment analysis-cc00daf
2e3e.json
```

Attention: hold the same cmd shell when running python
(3)Use main.py to analyze the sentiment to one keyword. Examples are shown below
(4)Use compare.py to compare the sentiment to two keywords. Examples are shown below

The code are shown below

TwitterAPI.py

```
import tweepy
import sys

def twitter_search(keyword,num):
    non_bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xfffd)

    #Please enter Twitter API credentials here
    #consumer_key = "Enter the consumer_key"
    #consumer_secret = "Enter the consumer_secret"
    #access_key = "Enter the access_key"
    #access_secret = "Enter the access_secret"

    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_key, access_secret)

    api = tweepy.API(auth)

    text = ""

    for tweet in tweepy.Cursor(api.search,q=keyword).items(num):
        text += tweet.text.translate(non_bmp_map)

    return text

if __name__ == '__main__':

    keyword = input("Please enter the keyword:\n")
    num = int(input("Please enter how many tweets you want to analyze:\n"))
    twitter_search(keyword,num)
```

NLPAPI.py

```
# Imports the Google Cloud client library
from google.cloud import language
from google.cloud.language import enums
from google.cloud.language import types

def NLP_analyze(text):
    # Instantiates a client
    client = language.LanguageServiceClient()

    # The text to analyze
    #with open('Tweets.txt', 'r') as review_file:
    #       text = review_file.read()

    #text = u'Hello, world!'
    document = types.Document(
        content=text,
        type=enums.Document.Type.PLAIN_TEXT)

    # Detects the sentiment of the text
    sentiment = client.analyze_sentiment(document=document).document_sentiment

    #print('Text: {}'.format(text))
    #print('Sentiment: {}, {}'.format(sentiment.score, sentiment.magnitude))

    return sentiment
```

main.py:

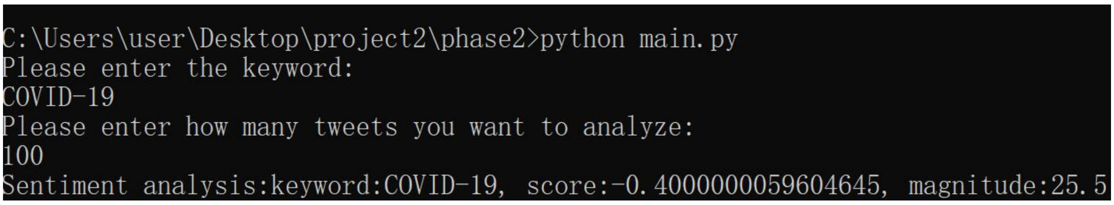
```
from TwitterAPI import twitter_search
from NLPAPI import NLP_analyze

#input module
keyword = input("Please enter the keyword:\n")
num = int(input("Please enter how many tweets you want to analyze:\n"))

#Twitter_API module
text = twitter_search(keyword,num)

#NLPAPI module
sentiment = NLP_analyze(text)

#output module
print('Sentiment analysis:keyword:{}, score:{},
magnitude:{}'.format(keyword,sentiment.score, sentiment.magnitude))

with open('sentiment.txt','a') as output:
    output.write('keyword:{}, score:{}, magnitude:{}\n'.format(keyword,sentiment.score,
sentiment.magnitude))

with open('tweets.txt','w',encoding='utf-8') as tweets:
    tweets.write(text)   #record analyzed text
```

compare.py:

```
from TwitterAPI import twitter_search
from NLPAPI import NLP_analyze

#input module
print("Compare the sentiment related keyword1 and keyword2 in twitter")
keyword1 = input("Please enter the keyword1:\n")
keyword2 = input("Please enter the keyword2:\n")
num = int(input("Please enter how many tweets you want to analyze:\n"))

#Twitter_API module
text1 = twitter_search(keyword1,num)
text2 = twitter_search(keyword2,num)

#NLPAPI module
sentiment1 = NLP_analyze(text1)
sentiment2 = NLP_analyze(text2)

#output module
print('Sentiment analysis:keyword:{}, score:{},
magnitude:{}'.format(keyword1,sentiment1.score, sentiment1.magnitude))
print('Sentiment analysis:keyword:{}, score:{},
magnitude:{}'.format(keyword2,sentiment2.score, sentiment2.magnitude))
if sentiment1.score < sentiment2.score:
    print("Sentiment to {} is more negative than sentiment to {} in
Twitter".format(keyword1,keyword2))
elif sentiment1.score == sentiment2.score:
    print("Sentiment to {} and sentiment to {} are similiar in
Twitter".format(keyword1,keyword2))
else:
    print("Sentiment to {} is more positive than sentiment to {} in
Twitter".format(keyword1,keyword2))
```

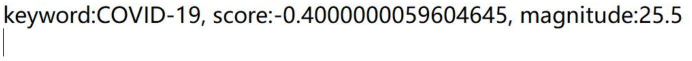
Test examples:

(1) Use main.py to analyze the sentiment to “COVID-19”

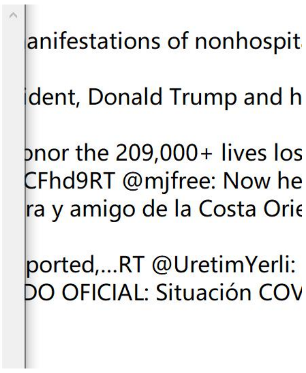


```
C:\Users\user\Desktop\project2\phase2>python main.py
Please enter the keyword:
COVID-19
Please enter how many tweets you want to analyze:
100
Sentiment analysis:keyword:COVID-19, score:-0.4000000059604645, magnitude:25.5
```

The text would be saved in tweets.txt and the sentiment reulst would be saved in sentiment.txt



```
keyword:COVID-19, score:-0.4000000059604645, magnitude:25.5
```

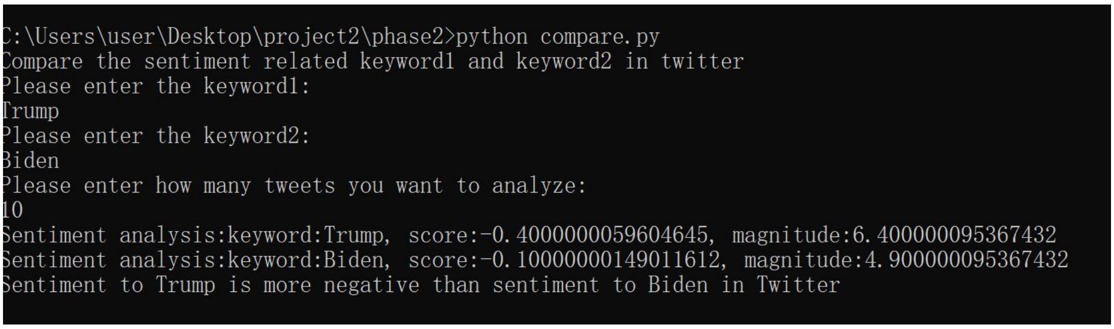


```
anifestations of nonhospit

ident, Donald Trump and h

onor the 209,000+ lives los
CFhd9RT @mjfree: Now he
ra y amigo de la Costa Orie

ported,...RT @UretimYerli:
DO OFICIAL: Situación COV
```

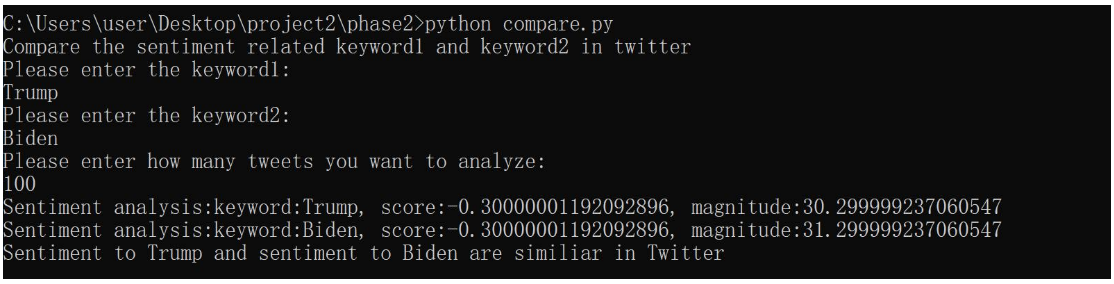
(2) Use compare.py to compare

In the example, we compare the sentiment in Twitter to Trump and Biden

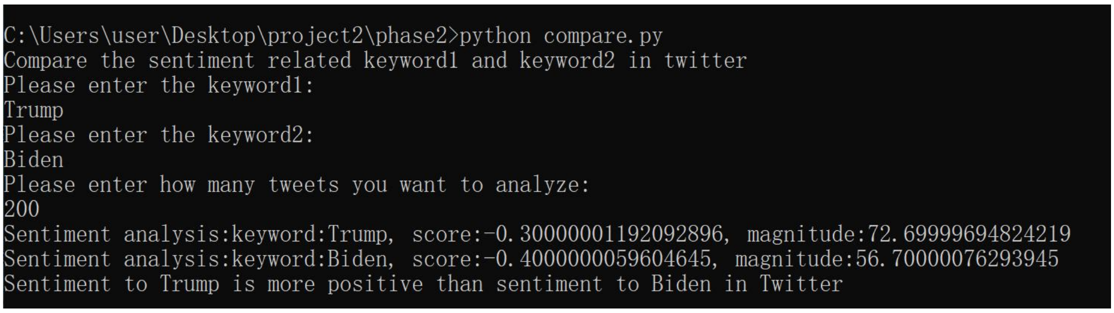
Attention: the result varies due to the change of number



```
C:\Users\user\Desktop\project2\phase2>python compare.py
Compare the sentiment related keyword1 and keyword2 in twitter
Please enter the keyword1:
Trump
Please enter the keyword2:
Biden
Please enter how many tweets you want to analyze:
10
Sentiment analysis:keyword:Trump, score:-0.4000000059604645, magnitude:6.400000095367432
Sentiment analysis:keyword:Biden, score:-0.10000000149011612, magnitude:4.900000095367432
Sentiment to Trump is more negative than sentiment to Biden in Twitter
```



```
C:\Users\user\Desktop\project2\phase2>python compare.py
Compare the sentiment related keyword1 and keyword2 in twitter
Please enter the keyword1:
Trump
Please enter the keyword2:
Biden
Please enter how many tweets you want to analyze:
100
Sentiment analysis:keyword:Trump, score:-0.30000001192092896, magnitude:30.299999237060547
Sentiment analysis:keyword:Biden, score:-0.30000001192092896, magnitude:31.299999237060547
Sentiment to Trump and sentiment to Biden are similiar in Twitter
```



```
C:\Users\user\Desktop\project2\phase2>python compare.py
Compare the sentiment related keyword1 and keyword2 in twitter
Please enter the keyword1:
Trump
Please enter the keyword2:
Biden
Please enter how many tweets you want to analyze:
200
Sentiment analysis:keyword:Trump, score:-0.30000001192092896, magnitude:72.69999694824219
Sentiment analysis:keyword:Biden, score:-0.4000000059604645, magnitude:56.70000076293945
Sentiment to Trump is more positive than sentiment to Biden in Twitter
```