

دنیای ماورای

۱۴۰۰

تشخیص احساسات و لحن نوشته



مؤلفین:

محمد مهدی واحد

سید طاها حسینی

استاد راهنما:

امیررضا کرباسچیان

کلاس جوانه هوش مصنوعی

دوره ۴۶ (ترم اول)

دبیرستان نیکان

۱۴۰۰/۹/۲۸

فهرست مطالب

۱	فهرست مطالب.....
۲	مقدمه.....
۲	کارهای موجود.....
۲	مسائل.....
۲,۳	راه حل ها.....
۳	روش شناسی.....
۳	یادگیری.....
۳	پژوهش.....
۳,۴,۵	پیاده سازی.....
۶	خروجی های طرح و نتایج.....
۶	کارهای آینده.....

مقدمه

تجزیه و تحلیل احساسات یک ابزار بازاریابی قدرتمند است که مدیران محصول را قادر می سازد تا احساسات مشتری را در کمپین های بازاریابی خود درک کنند. وقتی صحبت از شناخت محصول و برند ، وفاداری مشتری، رضایت مشتری ، موفقیت تبلیغات و تبلیغ و پذیرش محصول می شود، این عامل مهمی است. درک روانشناسی مصرف کنندگان می تواند به مدیران محصول و مدیران موفقیت مشتری کمک کند تا نقشه راه محصول خود را با دقت بیشتری تغییر دهند. اصطلاح بازاریابی مبتنی بر احساسات یک راه کار گسترده ای است که شامل پاسخ های عاطفی مشتری، مانند "مثبت"، "منفی"، "خنثی"، "تنفر"، "ناامیدی"، "خشم" و غیره است. درک روانشناسی پاسخ های مشتری همچنین می تواند تعریف و صحبت درباره ی آن محصول و برند را افزایش دهد.

دلیل خودمون برای انتخاب این موضوع این بود که دیدیم که خیلی از مسائل و مشکلات سر سوء تفاهم هایی است که به خاطر درست نخواندن متن پیش می آید پس ما تصمیم گرفتیم کاری را انجام دهیم که از این عواقب جلوگیری کند و به مردم در درک بهتر احساسات پیام ها کمک کند و دیگر به مشکل نخورند .

کارهای موجود

مسائل

مسئله ی ما که بعد از تحقیق و جست و جو بهش رسیدیم که بعد از کلی ایده پردازی و توجه به نیاز مردم سنجیدن موقعیت این موضوع که به نظرمون بهترین بود انتخاب کردیم که تشخیص درست احساسات و لحن از نظرات مردم بود تا با درصد خطای کم و در ۵ درجه علاقه مردم به سایت یا کالا مشخص بشه تا با استفاده از آنها بتوان آن محصول یا سایت را بهبود یافت.

همچنین برای این انتخاب سایت هایی مثل <https://github.com/ashalogic/Persian-Sentiment-Analyzer> و <https://github.com/Text-Mining/Persian-Sentiment-Resources> بررسی کردیم.

راه حل ها

راهه حل هایی که ما به ذهنمون رسید این بود که متن فارسی را با استفاده از لایبری های پایتون سنتیمنتش رو پیدا کنیم که برای فارسی نوشته شده بودن چون اگه خودمون می خواستیم از صفر این کار را انجام بدیم بیشتر از ۳ ماه با آموزش وقت می گرفت ولی بعد از تحقیق و بررسی متوجه شدیم که لایبری خوب و مناسب فارسی برای کار ما وجود نداره و این موضوع به نظرمون اومد که اول متن فارسی را به انگلیسی تبدیل کنیم

و بعد توسط یکی از لایبری های انگلیسی تشخیص احساسات و لحن، احساساتش رو در بیاریم و بقیه ی زمانمون رو روی این موضوع گذاشتیم و بعد از جست و جو به دو لینک <https://www.thepythoncode.com/article/translate-text-in-python> و <https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair> رسیدیم که به ترتیب ترجمه متن و تشخیص لحن بودن و بر اساس آنها کارمون رو تموم کردیم و نتیجه ی خوبی دریافت کردیم.

روش شناسی

یادگیری

یادگیری مباحث پایتون ، یادگیری ماشین و توسعه ی مدل و یادگیری بکارگیری مدل ، یادگیری گیت و گیت هاب و کار با آن و یادگیری استفاده از کتابخانه ها و api های مختلف .

فاز یادگیری حدود ۳ هفته طول کشید که بعد از ۲ ی مقدمات پایتون سراغ دیدن محتوای مربوط به مبحث اصلی توسعه ی مدل و بکارگیری آن مطالب . د نیاز دیگر (کار با api مطالعه ی کتابخانه های پایتون و ...) رفتیم .

پژوهش

طوفان فکری و دادن ایده ، پیدا کردن پروژه های مشابه در گیت هاب و تحقیق و پیدا کردن کتابخانه های کمکی در پایتون ، پیدا کردن ساختمان درست مدل ، پیدا کردن ایده ی مناسب برای استفاده از تحلیلگر احساسات در متون فارسی .

در این مرحله که حدود ۳ هفته تا پیاده سازی طول کشید :

۱. سراغ تحلیلگر های مشابه در گیت هاب رفتیم .
۲. روی کتابخانه های پایتون تحقیق و بهترینشان برای کارمان را انتخاب کردیم
۳. در آخر و پس از کسب اطلاعات کافی از بین راه حل ها بهترین و کارآمد ترینشان را انتخاب کردیم .

پیاده سازی

تقسیم کارهای موجود ، توسعه ی مدل و یا تغییر مدل های مشابه متناسب با زبان فارسی ، بکارگیری و استفاده از آن .

در این مرحله که با توجه به اینکه اطلاعات لازم را داشتیم حدود ۱ هفته طول کشید :

ساختار مناسب کد های مربوط به کتابخانه ی vader را پیاده سازی کردیم و با متن های انگلیسی تستش کردیم .

```
# import SentimentIntensityAnalyzer class
# from vaderSentiment.vaderSentiment module.
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Function to print sentiments
# of the sentence.
def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary,
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    print("Overall sentiment dictionary is : ", sentiment_dict)
    print("sentence was rated as ", sentiment_dict['neg']*100, "% Negative")
    print("sentence was rated as ", sentiment_dict['neu']*100, "% Neutral")
    print("sentence was rated as ", sentiment_dict['pos']*100, "% Positive")

    print("Sentence Overall Rated As", end = " ")

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        print("Positive")

    elif sentiment_dict['compound'] <= - 0.05 :
        print("Negative")

    else :
        print("Neutral")

# Driver code
if __name__ == "__main__" :

    print("\n1st statement :")
    sentence = "Geeks For Geeks is the best portal for \
the computer science engineering students."

    # function calling
    sentiment_scores(sentence)
```

سپس این ساختار را با توجه به نیازمان (۵ درجه احساسات) تغییر دادیم به صورتی که اگر:

درصد احساس منفی جمله صفر و درصد احساس مثبت جمله بزرگ تر از درصد احساس خنثی آن بود
درجه ی ۵ را خروجی دهد و اگر این گونه نبود درصد احساس مثبت را با نصف درصد احساس خنثی جمع
و در ۲۰ تقسیم کند و جواب را به صورت یک عدد اعشاری بین ۱ تا ۵ بدهد.

```
final sentiment analysis base on positive.py
File Edit Format Run Options Window Help
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
#vader library
from googletrans import Translator
#googletrans library
from pprint import pprint
#pprint library

def sentiment_scores(sentence):
    translator = Translator()
    translation = translator.translate(sentence, src="fa")
    translated = (f"{translation.text}")

    sid_obj = SentimentIntensityAnalyzer()

    sentiment_dict = sid_obj.polarity_scores(translated)

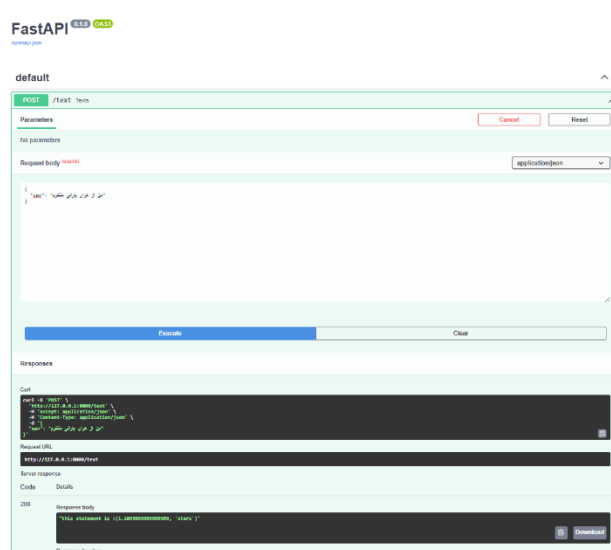
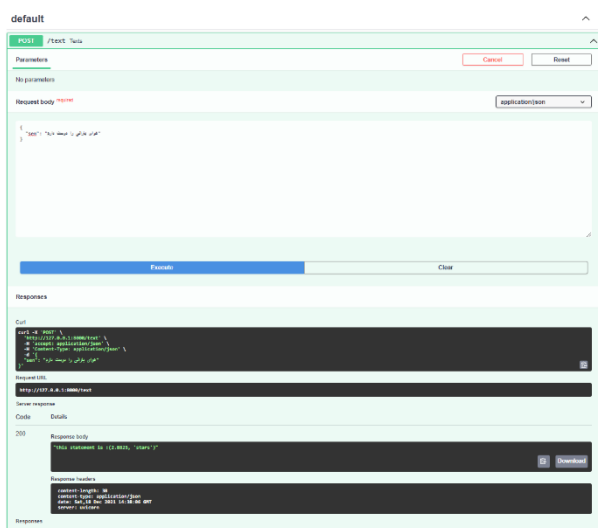
    if sentiment_dict['neg']*100==0 and sentiment_dict['pos']*100>sentiment_
        print("5 star")
    else :
        x = (sentiment_dict['pos']*100+((sentiment_dict['neu']*100)/2))/
        print(x,"star")

x=str(input("what is your statement ? :"))
print("\nthis statement is :")
sentiment_scores(x)
```

بعد از api گوگل ترنسلیت در کد استفاده کردیم تا متون فارسی را به انگلیسی ترجمه و سپس احساسات را تحلیل کنیم .

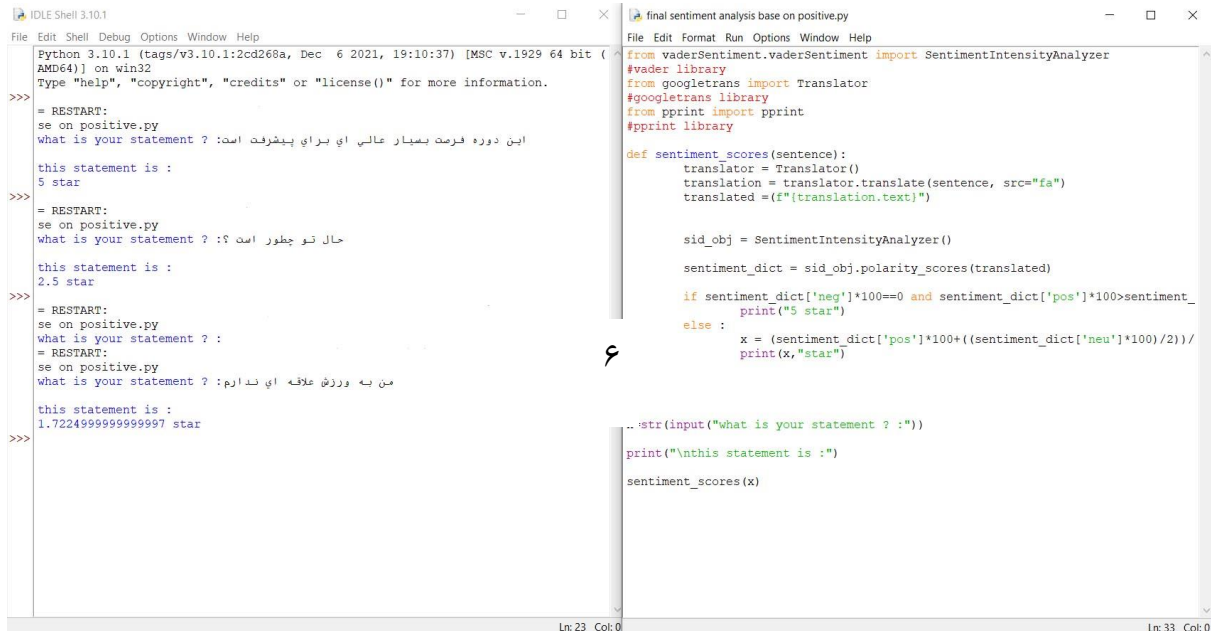
```
translator = Translator()
translation = translator.translate(sentence, src="fa")
translated =(f"{translation.text}")
```

برای بخش API بعد از تموم شدن قسمت توسعه باید اول اطلاعات مورد نظر از قسمت های مختلف و فیلم های آموزشی دریافت می شد و بد از درک کد ها نوشتن کد API کار مسئله پیچیده ای نبود و به سرعت با کار کردن با API می شد آن را یادداشت و کار را به اتمام رساند تا وارد بخش های بعدی شود.



خروجی های طرح و نتایج

و در نهایت ما با استفاده از یک کتابخانه ی مخصوص (vader) و api گوگل ترنسلیت برای استفاده از متون فارسی و قرار دادن بخش های فوق کنار هم کار نهایی را ارائه دادیم .



```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART:
se on positive.py
what is your statement ? این دوره فرصت بسیار عالی ای برای پیخرفت است:
this statement is :
5 star

>>>
= RESTART:
se on positive.py
what is your statement ? حال تو چطور است ؟
this statement is :
2.5 star

>>>
= RESTART:
se on positive.py
what is your statement ?
= RESTART:
se on positive.py
what is your statement ? من به ورزش علاقه ای ندارم:
this statement is :
1.7224999999999997 star

>>>
```

```
final sentiment analysis base on positive.py
File Edit Format Run Options Window Help
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
#vader library
from googletrans import Translator
#googletrans library
from pprint import pprint
#pprint library

def sentiment_scores(sentence):
    translator = Translator()
    translation = translator.translate(sentence, src="fa")
    translated = f"{translation.text}"

    sid_obj = SentimentIntensityAnalyzer()
    sentiment_dict = sid_obj.polarity_scores(translated)

    if sentiment_dict['neg']*100==0 and sentiment_dict['pos']*100>sentiment_
        print("5 star")
    else :
        x = (sentiment_dict['pos']*100+((sentiment_dict['neu']*100)/2))/
        print(x,"star")

..:str(input("what is your statement ? :"))
print("\nthis statement is :")
sentiment_scores(x)
```

کارهای آینده

ارزیابی محصول با توجه به نیاز مخاطب و پایین آوردن درصد خطای تشخیص لحن آن و به کمک آن تغییر پروژه و سایت و بهتر کردن آن با توجه به نیاز مشتری و توسعه ی آن برای احساسات بیشتر (خشم ، نگرانی و ...)