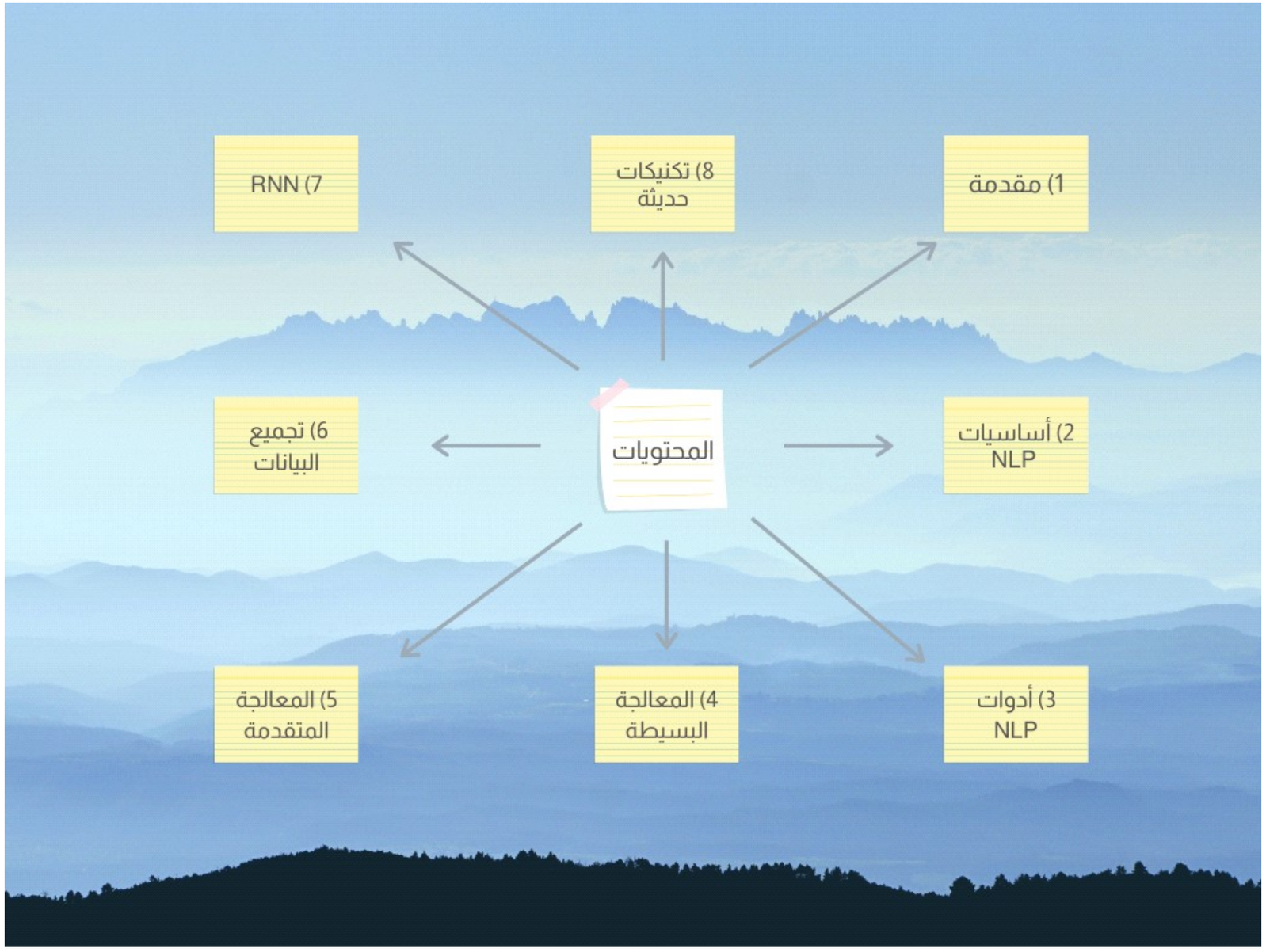
Natural Language Processing المعالجة اللغوية الطبيعية



المحتويات

القسم الخامس : المعالجة المتقدمة للنصوص

الجزء الثاني عشر : Vader

==========

في الطريقة السابقة تعلمنا كيف نقوم بتحليل الانطباع , بناء علي تدريب مسبق , عبر supervised data اي بيانات لها قيم features (الجمل نفسها) , وقيمة y (كلام سلبي ام ايجابي) . فيقوم الخوارزم بالتدرب عليه و تحليل اي جملة جديدة

لكن هل هناك طريقة لعمل تحليل انطباع دون تدريب مسبق , اي ان يكون هناك مكتبة تقوم بتحليل الجملة هل هي بانطباع ايجابي ام سلبي دون تدريب ؟ ؟

بالفعل يوجد باستخدام اداة Vader الموجودة في NLTK و هي اختصار جملة :

Valence Aware Dictionary Sentimental Reasoning

والتي تم تدريبها مسبقا علي ملايين الجمل بالفعل , ولديها القدرة علي تحليل اي جملة جديدة دون تدريب مسبق , لمعرفة شيئين هامين :

- أولا هل الإنطباع سلبي ام ايجابي
- ثانيا : قوة هذه الانطباع , هل هو قوي او ضعيف

فلو كان تحليلها ان المطعم سيئ جدا جدا , فسيئ هي النقطة الاولي و جدا هي الثانية

و أداة Vader يستطيع قياس ان كلمة like اقل من كلمة love , وان وضع علامة التعجب في نهاية الكلام معناها سخرية LOVE IT !!!! , وان كلمة not او never تعكس المعني تماما , كذلك يحاول فهم الكلام الساخر الذي يستخدم كلمات ايجابية بمعني سلبي

_

و لعمل تطبيق عملي , نبدأ بتحميل ملف vader_lexicon من مكتبة NLTK ثم استدعاء الاوبجكت هكذا :

```
import nltk
nltk.download('vader_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
```

وهو ما يقوم باظهار اربع قيم لكل جملة معطاة , هي neg وهي مقدار الانطباع السلبي , neu مقدار الانطباع المتعادل , pos مقدار الانطباع الايجابي , compound و هي المحصلة النهائية للارقام , وجميع هذه الارقام تتراوح بين 0 و 1 عدا الاخير من سالب 1 الي 1

فحينما نقوم بتقييم الجمل التالية :

```
a = 'This was a good movie.'
sid.polarity_scores(a)
```

```
a = 'This was the best, most awesome movie EVER MADE!!!'
sid.polarity_scores(a)

a = 'This was the worst film to ever disgrace the screen.'
sid.polarity_scores(a)

a = 'i dont hate you.'
sid.polarity_scores(a)
```

مع ملاحظة ان الجملة الاخيرة فهم ان don’t تعني عكس hate فكان لها رقم ايجابي

و ممكن ان نطبقها علي ملف تقييم الافلام

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('../TextFiles/amazonreviews.tsv', sep='\t')
df.head()

df['label'].value_counts()

df.dropna(inplace=True)

blanks = []  # start with an empty list

for i,lb,rv in df.itertuples():  # iterate over the DataFrame
    if type(rv)==str:            # avoid NaN values
        if rv.isspace():         # test 'review' for whitespace
            blanks.append(i)     # add matching index numbers to the list

df.drop(blanks, inplace=True)

df['label'].value_counts()
```

ثم نقوم باستعراض التعليقات , وعمل توقع لها باستخدام فيدر

```
df.loc[0]['review']
```

'Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your mind so well I would recomend it even to people who hate vid. game music! I have played the game Chrono Cross but out of all of the games I have ever played it has the best music! It backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. It would impress anyone who cares to listen! ^_^'

```
sid.polarity_scores(df.loc[0]['review'])
df.loc[0]['label']
```

و هنا لعدد من التعليقات

```
for i in range (10,25) :
    print(df.loc[i]['review'])
    print()
    print(sid.polarity_scores(df.loc[i]['review']))
    print()
    print(df.loc[i]['label'])
    print('-----------------------------------')
```

ثم نقوم بعمل عمود سكور , وكومباوند , وكومباند سكور , لحساب دقة التوقع بناء علي علي قيمة واي في الداتا نفسها

```
df['scores'] = df['review'].apply(lambda review: sid.polarity_scores(review))
df.head()
```

```
df['compound']  = df['scores'].apply(lambda score_dict: score_dict['compound'])
df.head()
```

```
df['comp_score'] = df['compound'].apply(lambda c: 'pos' if c >=0 else 'neg')
df.head()
```

ثم يتم حساب الدقة

```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix

accuracy_score(df['label'],df['comp_score'])

print(classification_report(df['label'],df['comp_score']))

print(confusion_matrix(df['label'],df['comp_score']))
```

-