



المحتويات

				التطبيقات	العقبات و التحديات	تاريخ NLP	ما هو NLP	المحتويات	1) مقدمة
					البحث في النصوص	ملفات pdf	الملفات النصية	المكتبات	2) أساسيات NLP
T.Visualization	Syntactic Struc.	Matchers	Stopwords	NER	Stem & Lemm	POS	Sent. Segm.	Tokenization	3) أدوات NLP
	Dist. Similarity	Text Similarity	TF-IDF	BOW	Word2Vec	T. Vectors	Word embed	Word Meaning	4) المعالجة البسيطة
T. Generation	NGrams	Lexicons	GloVe	L. Modeling	NMF	LDA	T. Clustering	T. Classification	5) المعالجة المتقدمة
	Summarization & Snippets		Ans. Questions		Auto Correct	Vader	Naïve Bayes	Sent. Analysis	
Search Engine	Relative Extraction		Information Retrieval		Information Extraction		Data Scraping	Tweet Collecting	6) تجميع البيانات
					Rec NN\TNN	GRU	LSTM	Seq to Seq	7) RNN
Chat Bot	Gensim	FastText	Bert	Transformer	Attention Model	T. Forcing	CNN	Word Cloud	8) تكنيكات حديثة

القسم الثاني : أساسيات NLP

الجزء الثاني : التعامل مع الملفات النصية

نشرح هنا كيفية التعامل مع الملفات ذات النصوص , للقراءة منها و الكتابة فيها

لعمل ملف نصي و الكتابة فيه , يمكن استخدام هذا الامر

```
%%writefile text1.txt  
bla bla bla bla bla bla
```

فحينما نقوم بعمل اسم الملف يمكن بعدها كتابة اي سطور , وسيقوم بإنشاء الملف

و اذا اردنا قراءة ملف معين يتم استخدام الامر

```
my_file = open('test.txt')
```

الان الكائن my_file اصبح يشير الي الملف , فيمكن قراءته هنا

```
my_file.read()
```

و الذي سيظهر محتواه

لكن لاحظ انه اذا تكرر استخدام نفس الامر `my_file.read` فسيظهر المحتوى فارغ , لان السهم الذي كان يقرا في الملف صار في النهاية , ولارجاعه في البداية نستخدم الامر `seek` سيبك

```
my_file.seek(0)
```

هنا لو تم استخدام الامر `my_file.read` سيظهر الكلام مرة اخرى

ولو تم استخدام الامر

```
my_file.readlines()
```

فسيقوم بياثون بقراءة الملف و جعل كل سطر في جزء خاص به , وعمل ليست فيها عناصر كل عنصر هو سطر منفصل

و في النهاية يتم اغلاق الملف

```
my_file.close()
```

* * * * *

و لعمل كتابة علي الملف يتم استخدام

```
my_file = open('test.txt','w+')
```

و هنا تكون الكتابة

```
my_file.write('This is a new first line')
```

مع ملحوظة ان الملف سيتم عمل مسح القديم و كتابة الجديد فقط

اما اذا اردنا الاضافة علي الملف يكون هكذا :

```
my_file = open('test.txt','a+')
```

و تكون الاضافة هكذا :

```
my_file.write('\nThis line is being appended to test.txt')
```

مع ملاحظة ان عمل \n في البداية حتي يقوم بعمل سطر جديد

و يمكن كذلك فتح الملف للتعديل هكذا :

```
%%writefile -a test.txt
This is more text being appended to test.txt
And another line here.
```

مع ملاحظة عمل حرف a- ليعرف انه سيتم عمل append

كذلك ممكن قراءة كل سطر علي حدة هكذا :

```
with open('test.txt','r') as txt:
    first_line = txt.readlines()[0]

print(first_line)
```

او

```
with open('test.txt','r') as txt:
    for line in txt:
        print(line, end="")
```

* * * * *

ملفات CSV

وهي اختصار Comma-separated values اي القيم التي يتم فصلها بكوما

و يمكن قراءة الملف هكذا :

```
import pandas as pd  
data = pd.read_csv('01.csv')# , skiprows = 2)  
data.head()
```

مع التأكيد علي تحديد عدد الصفوف التي يمكن تجاهلها
ايضا يمكن فتح ملفات اكسل هكذا :

```
data = pd.read_excel('02.xlsx')# , skiprows = 2)  
data.head()
```

و يمكن انشاء ملف CSV و الكتابة فيه هكذا

```
outfile = open('03.csv', 'w')  
outfile.write('a')  
outfile.close()
```

و انشاء ملف اكسل و الكتابة فيه

```
outfile = open('04.xls', 'w')  
outfile.write('a')  
outfile.close()
```

و انشاء ملف نص و الكتابة فيه

```
f= open('5.txt','w') #write  
f.write('write this line in the file')  
f.close()
```

و قراءته

```
f= open('5.txt','r') #read  
for a in f:  
    print(a)
```


او الاضافة عليه

```
f= open('5.txt','a') #append
f.write('\nmore lines')
f.close()
```

كذلك يمكن حفظ اي داتا بصيغة dataframe الي ملف csv بسهولة

```
import numpy as np
data = pd.DataFrame(pd.Series(np.random.rand(10000)))
data.head(20)
```

```
data.to_csv('6.csv')
```

* * * * *

كما يمكن التعامل مع اللغة العربية , بشرط اضافة

```
encoding="utf8"
```

فيكون الكود هكذا

```
f= open('7.txt','w', encoding="utf8")  
f.write('سطور باللغة العربية')  
f.close()
```

```
f= open('7.txt','a', encoding="utf8")  
f.write('\nسطر ثاني')  
f.write('\nسطر ثالث')  
f.write('\nسطر رابع')  
f.write('\nسطر أخير')  
f.close()
```

```
f= open('7.txt','r', encoding="utf8")  
for a in f:  
    print(a)
```