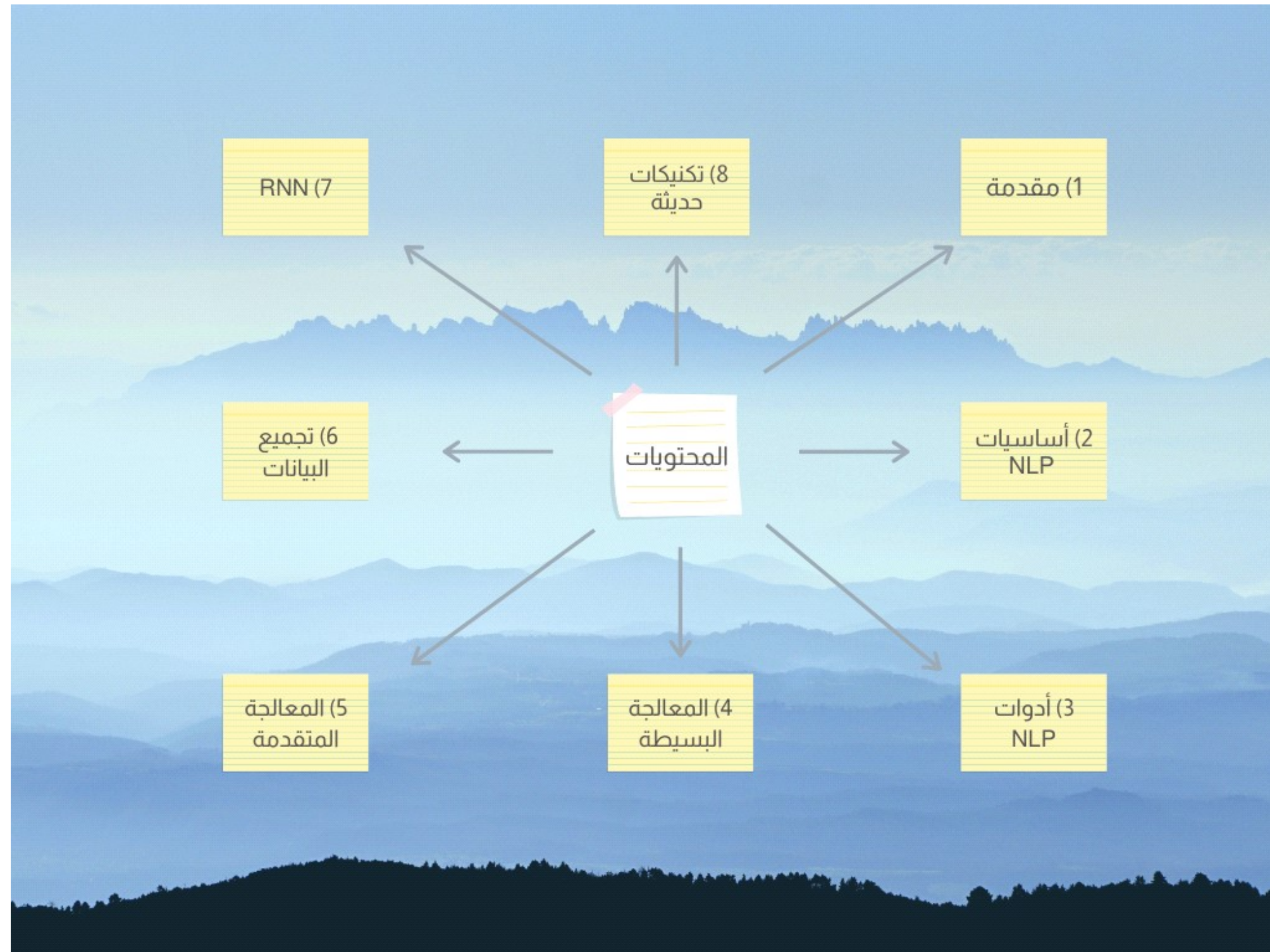


NATURAL LANGUAGE PROCESSING

المعالجة اللغوية الطبيعية



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	L. Modeling	NGrams	Lexicons	GloVe	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	RNN	(7) RNN
Chat Bot	Gensim	FastText	Bert	Hug. Face	Attention Model	T. Forcing	CNN	Word Cloud	(8) تكتيكات حديثة

القسم الثامن : تكنيكات حديثة

الجزء السابع : FastText

=====

هي من المكتبات القوية و السريعة في التعامل مع النصوص , وهي من إنتاج فيسبوك

يمكن التعامل معها عبر تحميل المكتبة بشكل مباشر , او من مكتبة gensim , لكن تحميلها بشكل مباشر يعطي ادوات اكثر

الموقع الرسمي لها :

<https://fasttext.cc/>

كما ان صفحتهم علي جيتهاب

<https://github.com/facebookresearch/fastText>

و تحميلها علي Jupyter فيه بعض الخطوات المعقدة , وهي هنا :

<https://medium.com/@oleg.tarasov/building-fasttext-python-wrapper-from-source-under-windows-68e693a68cbb>

لذا فالأسهل التعامل معها علي كاجل حيث انها تم تنصيبها بالفعل

و يفضل تحميل هذه الملفات التي سنستخدمها من هنا

<https://github.com/sumanp/text-classification-fastText>

ومن هنا يمكن تحميل العديد من النماذج التي تم تدريبها علي ويكيبيديا

<https://fasttext.cc/docs/en/english-vectors.html>

و هي ملفات تحتوي علي مئات الملايين من الكلمات التي تم تدريبها علي ويكيبيديا , وتشتمل علي اكثر من ملف للغة الانجليزية , كما ان منها العديد من الملفات الاخر لعدد 150 لغة , منها اللغة العربية

و هناك نوعين من الملفات في كل لغة , امتداد bin و vec

امتداد vec يكون فيه الكلمات التي تم التدريب عليها , وقيم ال map الخاصة بها , بينما امتداد bin هي التي يمكن فتحها بالخوارزم للتعامل معها و اكتشاف كلمات جديدة

ملف اللغة الانجليزية :

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz>

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.vec.gz>

و ملف اللغة العربية هنا :

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ar.300.bin.gz>

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ar.300.vec.gz>

كما ان ملف اللهجة المصرية هنا :

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.arz.300.bin.gz>

<https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.arz.300.vec.gz>

و اذا كنت ستعمل علي كاجل , فيمكن البحث عن قواعد البيانات بسهولة , حتي يتم العمل عليها دون تحميلها من البداية

و الانجليزية هنا :

<https://www.kaggle.com/kingarthur7/fasttext-common-crawl-bin-model>

<https://www.kaggle.com/facebook/fasttext-wikinews>

و العربية :

<https://www.kaggle.com/javadhelali/fasttext-pretrained-arabic-word-vectors>

كما يمكن تحميل اي لغة علي كاجل بالأمر :

```
import fasttext.util
fasttext.util.download_model('en', if_exists='ignore')
```

و كتابة حروف اللغة بدلا من en

ثم قراءة الملفات عبر وضع المسار الخاص بها

```
import fasttext  
ft = fasttext.load_model('../input/fasttext-common-crawl-bin-model/cc.en.300.bin')
```

و بمجرد تحميل الملف , يمكن البدء في قراءته عبر الدالة :

```
import io
```

```
def load_vectors(fname):
```

```
    fin = io.open(fname, 'r', encoding='utf-8', newline='\n', errors='ignore')
```

```
    n, d = map(int, fin.readline().split())
```

```
    data = {}
```

```
for line in fin:
    tokens = line.rstrip().split(' ')
    data[tokens[0]] = map(float, tokens[1:])
return data
```

و لكن يفضل عمل تعديل فيها بحيث تقرأ عدد محدد من الصفوف لعدم ضياع الوقت هكذا :

```
import io
```

```
def load_vectors(fname,length = 100000):
    fin = io.open(fname, 'r', encoding='utf-8', newline='\n', errors='ignore')
    n, d = map(int, fin.readline().split())
    data = {}
    i=0
    for line in fin:
        i+=1
```



```
tokens = line.rstrip().split(' ')
data[tokens[0]] = map(float, tokens[1:])
if i == length :
    break
return data
```

كذلك يمكن فتح أحد الموديولات التي تم التدريب عليها بالفعل , و استخدامه لإيجاد كلمات قريبة من كلمات معينة

و نري التطبيق في ملفات fast1,fast2,fast3,fast4

كما يمكن التعامل مع الموديلات سابقة التدريب ، و الموديلات التي قامت فيسبوك بتدريبها علي ويكيبيديا موجودة هنا

<https://fasttext.cc/docs/en/pretrained-vectors.html>

كما أننا يمكن استخدام fast text من مكتبة genism , حيث انها تتسم بالسرعة و السهولة, ولكن بإمكانيات أقل

* * * * *