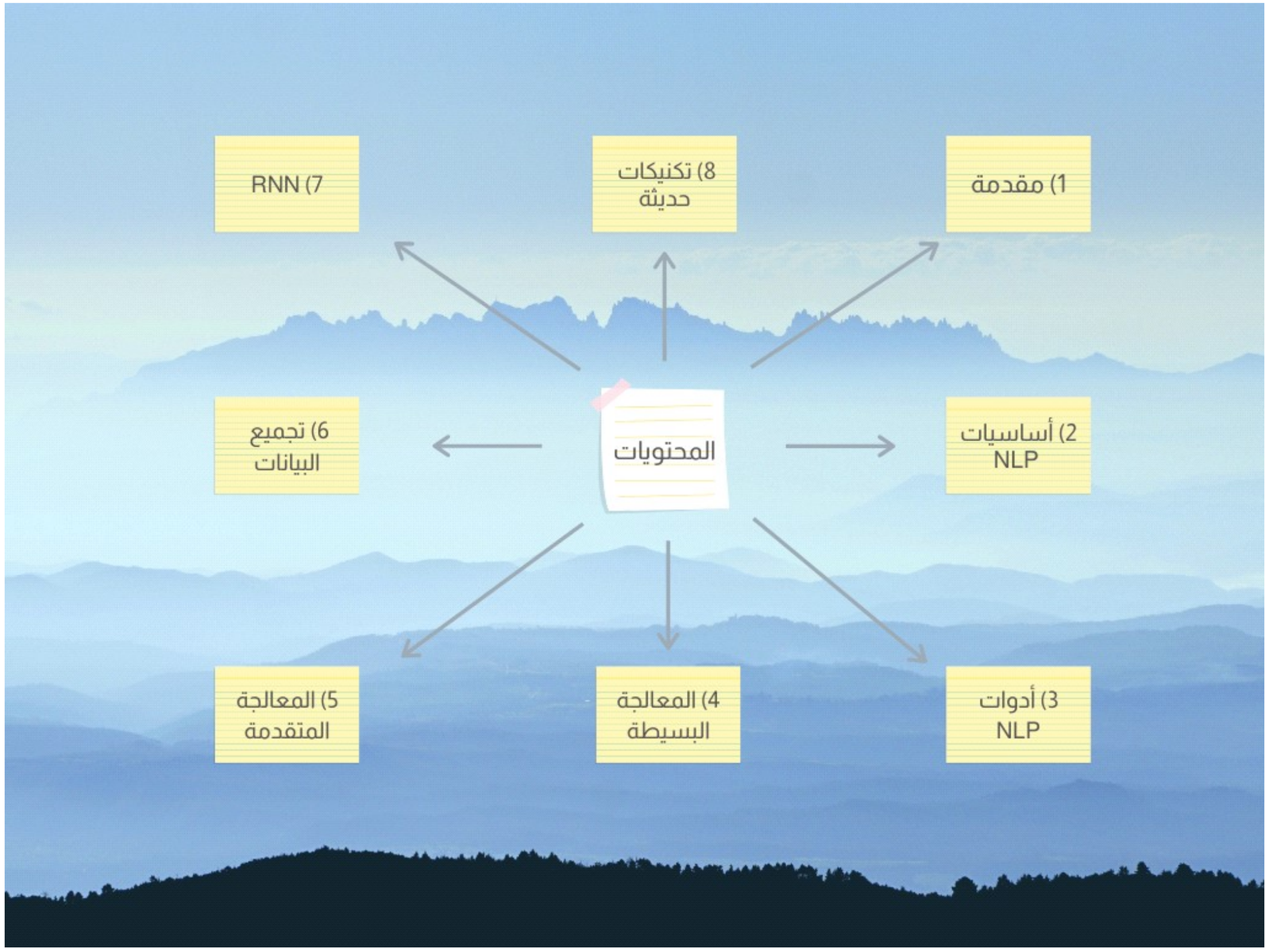
Natural Language Processing المعالجة اللغوية الطبيعية



المحتويات

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

القسم الرابع : المعالجة البسيطة للنصوص

الجزء الثامن : Distributional Similarity

==========

التشابه التوزيعي :

نتناول هنا ما يسمي التشابه التوزيعي , وهذا لحل مشاكل متعلقة بالبحث عن تشابه الكلمات مع بعضها البعض

فالقواميس ليست موجودة لكل اللغات , كما ان اي قاموس لن يحتوي علي جميع الكلمات , لأن هناك مشكلة في ال recall اذ ان الاف الكلمات غير متواجدة فيه (خاصة الكلمات الجديدة)

ففكرة التشابه التوزيعي يمكن اعتمادها دون الاستعانة بقاموس ما

و تقوم الفكرة علي أن أي **كلمة يمكن فهمها عبر تواجدها جوار عدد آخر من الكلمات التي نعرفها بالفعل**

و هنا مثال واضح . .

فلو كان لدينا عبارات هنا هي :

- ذبح أبي **قنعرا**
- لم يحب أخي لحم **القنعر** البارحة
- هرول **القنعر** هربا من الليث
- إنه سمين **كالقنعر**

فيمكن استنباط أن القنعر هو من أنواع الحيوانات السمينة . .

وهذه هي فكرة التشابه التوزيعي , ان نتعرف علي معني كلمة مجهولة , عبر الاستعانة بكلمات أخري معلومة

_

و اذا تذكرنا فكرة مصفوفة الكلمات , فيمكن عبر البحث عن الارقام المتشابهة , ان نعرف ان كلمتي battle , soldier قريبتان من بعضهما , بينما كلمتي clown , fool قريبتان من بعضهما

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

كما اننا يمكن أن نعرف ان روايتي قيصر و هنري الخامس قريبتان من بعضهما كذلك

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

و بدلا من استخدام ملفات كاملة , يمكن فقط البحث في عدد محدود من الكلمات حول كل كلمة مطلوبة , وليكن عشرون كلمة

واذا كان لدينا اربع جمل هكذا , بحيث تحتوي كلا منها علي كلمة معينة نريد التعامل معها

- equal amount of sugar, a sliced lemon, a tablespoonful of **apricot** preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened to that of
- of a recursive type well suited to programming on the **digital** computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and **information** necessary for the

60 study authorized in the first section of this

و يمكن عمل المصفوفات هكذا , وادراك ان كلمتي pineapple , apricot قريبتان من بعضهما , بينما , digital information قريبتان من بعضهما , وذلك لتشابه عدد مرات تكرارهما في الملفات

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

و هنا يمكن استخدام احد الخوارزميات التي تسمي PPMI, و هذا قانونها

Dan Jurafsky

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w,context)

	computer	data	pinch	result	sugar	**p(w)**
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

65

و هنا نقوم بعمل الخطوات التالية :

- ففي البداية نقوم بحساب مجموع الارقام كلها و هي 19
- ثم تحديد احتمالية تقاطع كل كلمة مع الملف , عبر قسمة الرقم علي الرقم الكلي
- ثم تحديد احتمالية الكلمة بالكامل , و قسمتها علي الرقم علي الرقم الكلي
- ثم تحديد احتمالية الملف بالكامل , و قسمتها علي الرقم علي الرقم الكلي

Dan Jurafsky

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

	p(w,context)					p(w)
	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

- pmi(information,data) = $\log_2$ (.32 / (.37*.58)) = .58

(.57 using full precision)

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

66

اخيرا حساب PMI لكل كلمة عبر حساب لوغاريتم الرقم المتقاطع , علي حاصل ضرب قيمة الصف في العمود , هكذا , مع التأكيد علي ان الرقم السالب يتم حسابه بصفر و الموجب كما هو

لكن المشكلة ان الارقام غير منطقية , فنري ان صفي pineapple , apricot قريبان جدا من بعضها , بشكل كبير , بينما digital , information ليسا بهذا القرب , علي الرغم ان العكس هو الصحيح

و السبب ان هناك bias في هذه المعادلة , لذا نستخدم فكرة لابلاس

Add-2 Smoothed Count(w,context)

	computer	data	pinch	result	sugar
apricot	2	2	3	2	3
pineapple	2	2	3	2	3
digital	4	3	2	3	2
information	3	8	2	6	2

	p(w,context) [add-2]					**p(w)**
	computer	data	pinch	result	sugar	
apricot	0.03	0.03	0.05	0.03	0.05	0.20
pineapple	0.03	0.03	0.05	0.03	0.05	0.20
digital	0.07	0.05	0.03	0.05	0.03	0.24
information	0.05	0.14	0.03	0.10	0.03	0.36
p(context)	0.19	0.25	0.17	0.22	0.17	

68

و سنستخدم هنا اضافة رقم 2 علي الجميع و اعادة حساب القيم

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

PPMI(w,context) [add-2]

	computer	data	pinch	result	sugar
apricot	0.00	0.00	0.56	0.00	0.56
pineapple	0.00	0.00	0.56	0.00	0.56
digital	0.62	0.00	0.00	0.00	0.00
information	0.00	0.58	0.00	0.37	0.00

نري هنا ان الارقام اختلفت و صارت اقرب للمنطق

_