

心理與神經資訊學

(Psychoinformatics & Neuroinformatics)

課號: Psy5261

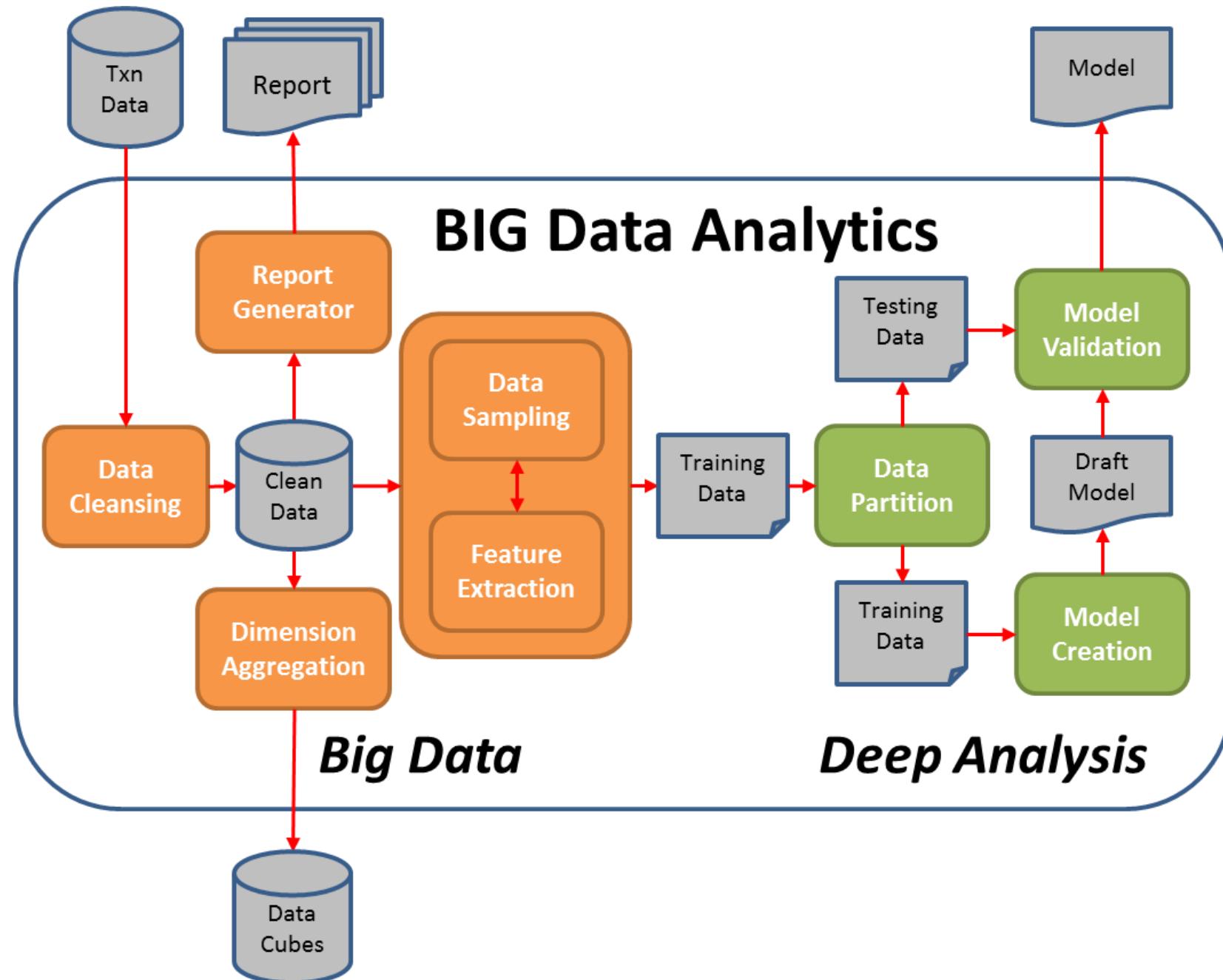
識別碼: 227U9340

教室:彷彿在雲端

時間: —789



資料的清理是使用的第一步



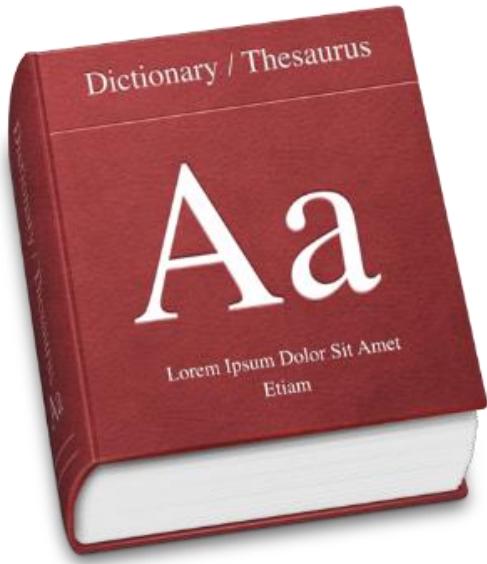


請先pip install nltk jieba gensim

!!

文字資料的解析 (RE)

心理學案例研究



小美研究心理語言學。她想要：

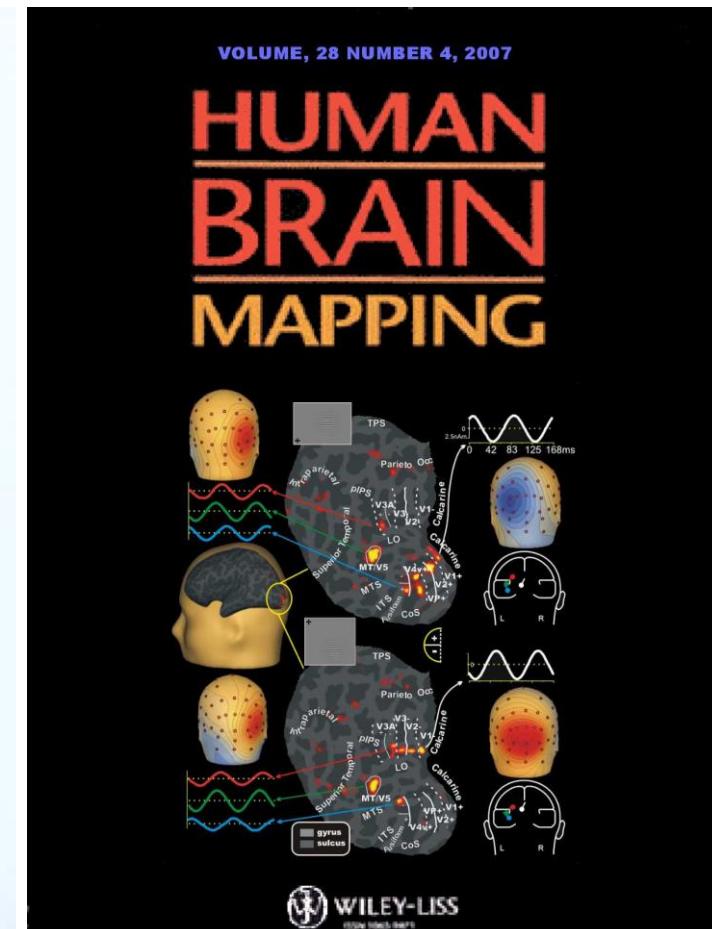
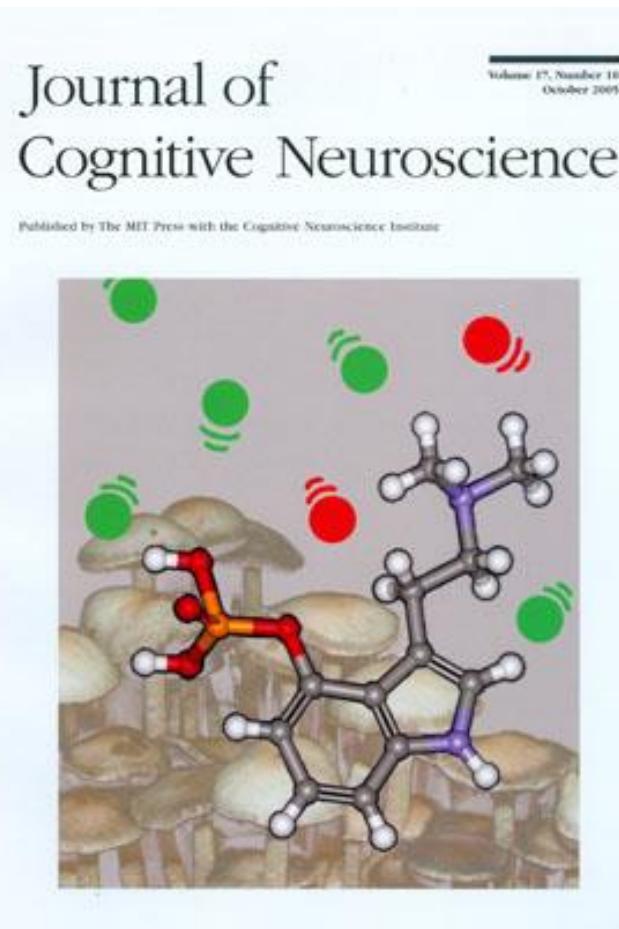
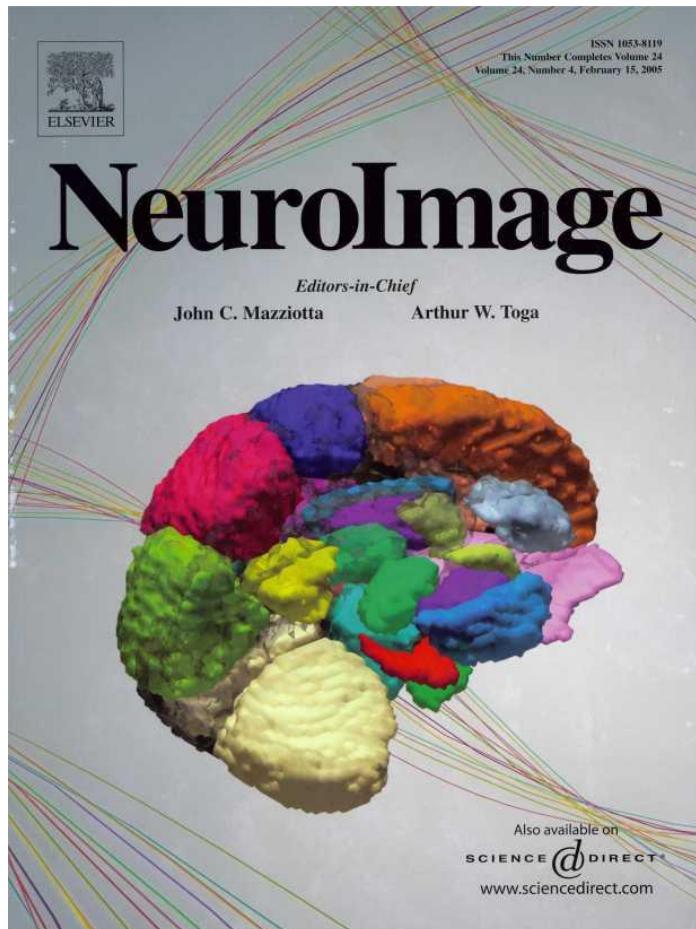
使用3個字元組成的英文字(如dog、cat等)
及3個字元組成的非英文字(如frw、ukq等)

並找出單字辨認的反應時間和影響反應時間的因素

除了手動從字典找字外有沒有其他的方法?

腦科學案例研究

小明要寫程式讓電腦幫我們讀papers



每家期刊格式都不同，怎麼辦？

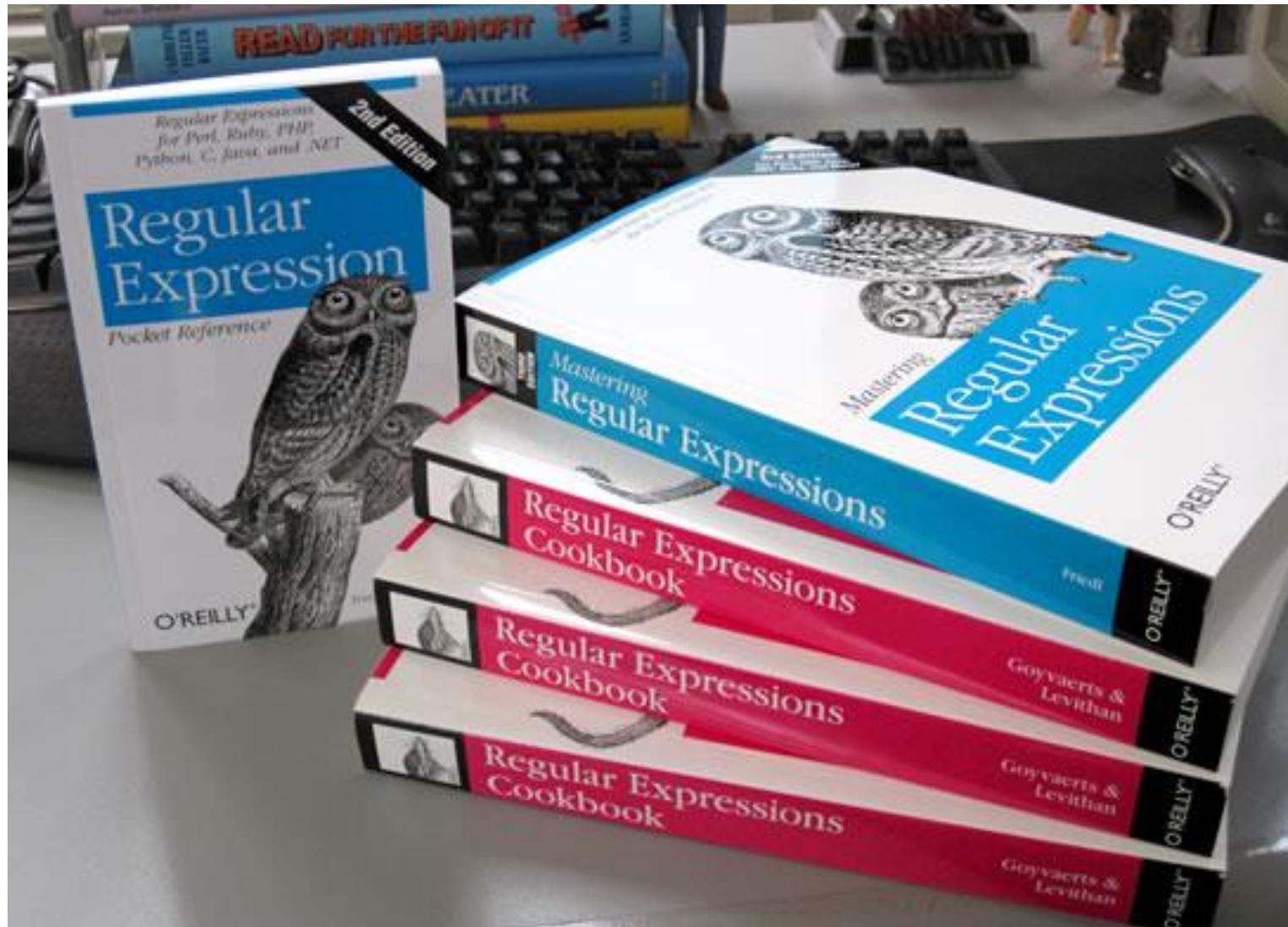
Python String Module

Python內建就有一些文字資料處理能力

```
a='This is a built-in module'  
print(a.lower(),a.upper())  
print(a.split(' '),a.split('-'))  
print(a.find('built'),a[10:15])  
print(a.replace('module','library'))  
print(a.count('is'))  
print(a.center(50,'='))
```

正規表示式(Regular Expression)

RegEx博大精深，今天只能簡介



正規表示式(Regular Expression)

RegEx的文字處理能力非常強大



Oh, no – the rumors were true. Our entire division's been replaced
by a regular expression.

正規表示式(Regular Expression)

看起來很像火星文



臺灣身份證號碼簡單檢查：

```
import re
print(bool(re.match('[A-Za-z]\d{9}', 'a123456789')))
```

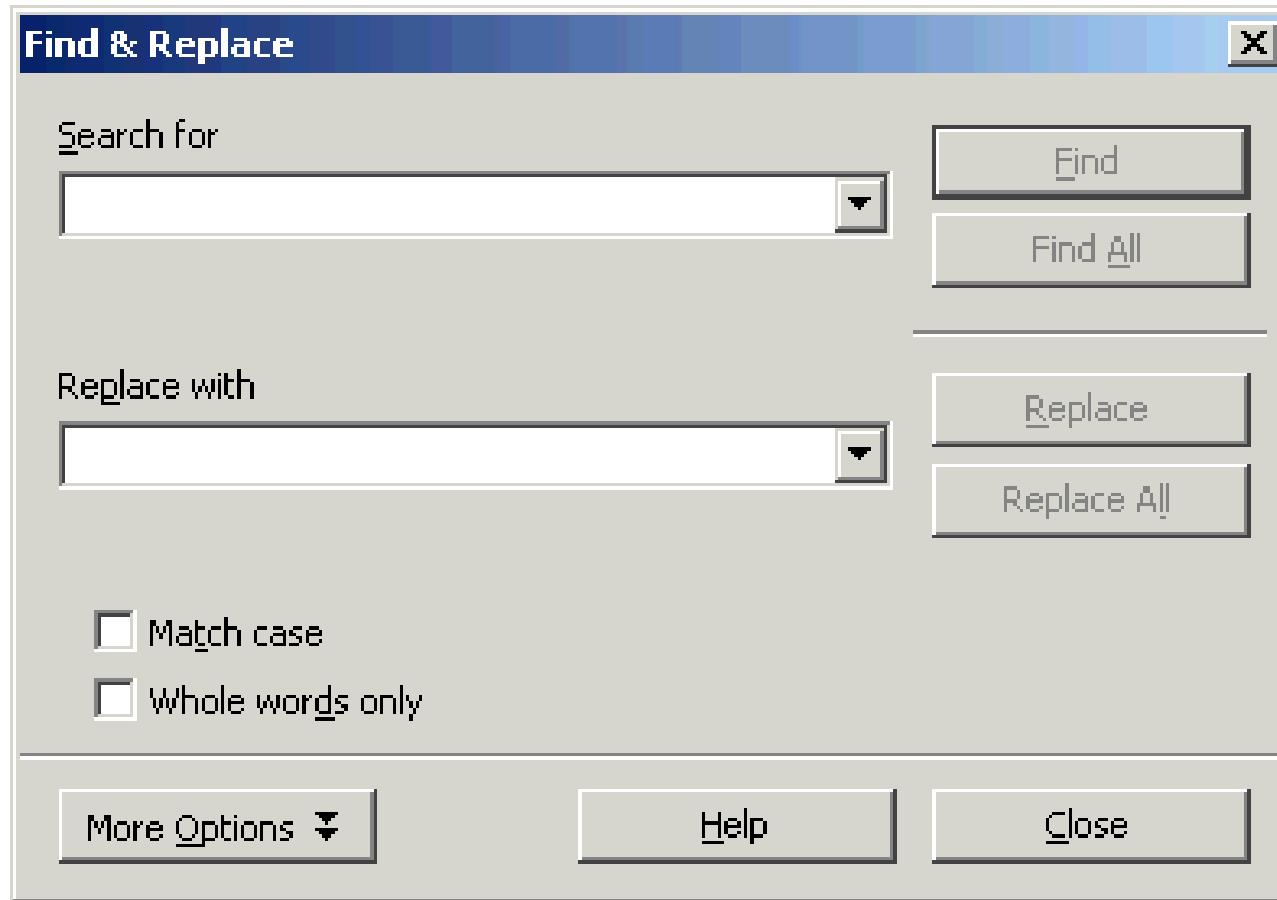
電子郵件地址的簡單檢查：

```
import re
print(bool(re.match('[@]+@[^.][@]+','a@b.c')))
# [@]+ 代表一個以上非@的任何字元
```

Python的re module文件看這裡

正規表示式(Regular Expression)

功能可概略分成"搜尋"與"取代"



搜尋：

`re.compile()`
`re.search()`
`re.match()`
`re.findall()`
`re.split()`

.....

取代：

`re.sub()`

.....

強大處在於搜尋的是樣式(pattern)而非一特定字

用RegEx來搜尋

```
import re  
regex=re.compile('abc',re.IGNORECASE)  
for txt in ['abc','hello abc','hi AbC aBc']:  
    print('-'*50)  
    out=regex.search(txt) #inexact match  
    if(out): print('search() found:',out.string)  
    out=regex.match(txt) #exact match  
    if(out): print('match() found:',out.string)  
    out=regex.findall(txt) #search into a list  
    if(out): print('findall() found:',out)
```

應用:搜尋retweets (回推)

```
import re

tweets=['RT @spiketren No class tomorrow',
'No class tomorrow (via @spiketren)']

rt=re.compile('^(RT|via) (@\w+)')
#rt=re.compile('^(RT|via) (@\w+\s*)')

for t in tweets:
    m=rt.search(t)
    print(m.group(1),m.group(2));
    #print(m[0],m[1],m[2]);
```



用RegEx來取代

修完本課失業後至少可以當spammer?

```
import re
```

```
html='<body><b>test</b><img src=test.jpg></body>'
```

```
print(re.sub('<[^<]*>', '', html))
```

```
e='@mail.ncku.edu.tw'
```

```
t='pichun_huang,chendy'
```

```
print(re.sub('\w+', '\g<0>' + e, t))
```



職稱	助理教授
姓名	黃碧群
e-mail	pichun_huang
專長	視知覺，弱視，空間視覺的

職稱	助理教授
姓名	陳德祐
e-mail	chendy
專長	人類及動物的功能性腦造影

**傳統NLP: 字/詞→類別/風格→主題
(NLTK → LIWC → LDA)**

自然語言處理(NLP)

自然語言處理 (Natural Language Processing)

是人工智慧和語言學領域的分支學科。

在這此領域中探討如何處理及運用自然語言；
自然語言認知則是指讓電腦「懂」人類的語言。

句子「我們把香蕉給猴子，因為它餓了」和

「我們把香蕉給猴子，因為它熟透了」

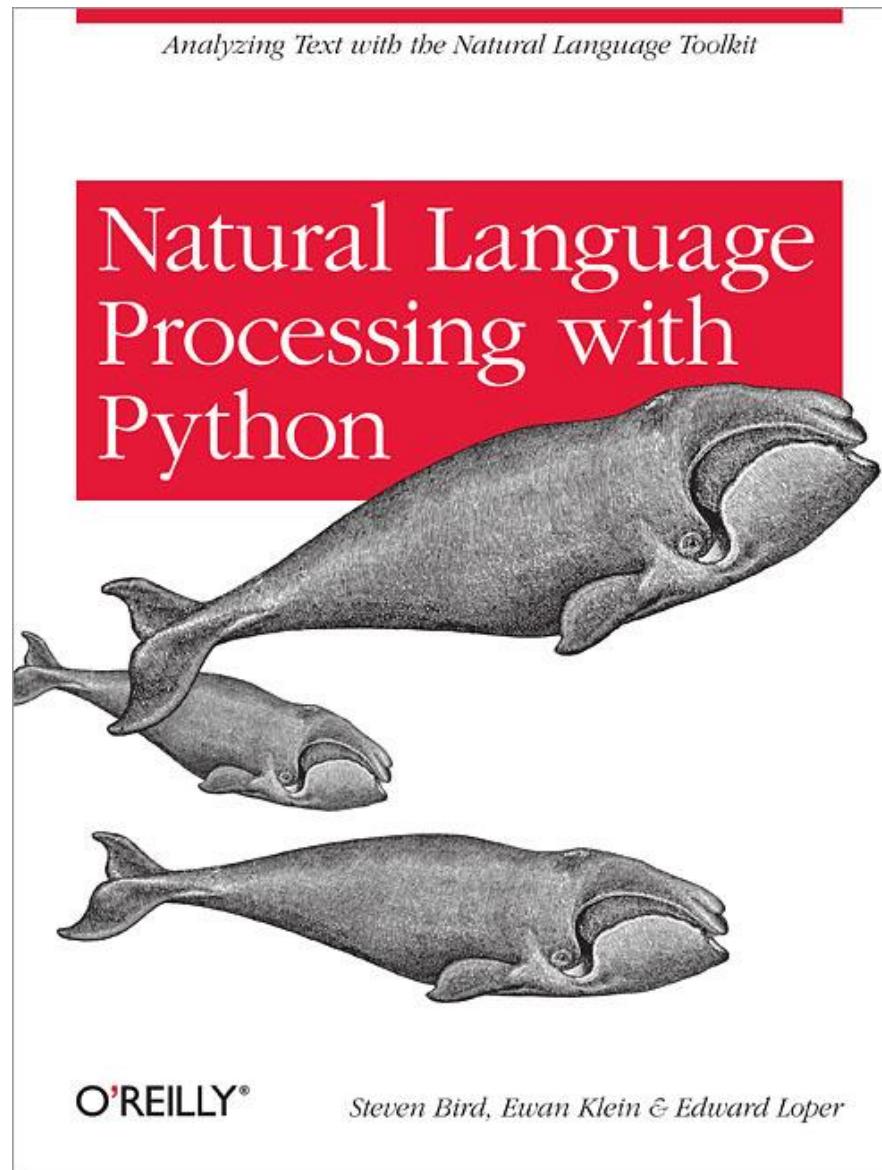
有同樣的結構。但是代詞「它」在第一句中指的是
「猴子」，在第二句中指的是「香蕉」。

如果不了解猴子和香蕉的屬性，無法區分。



使用Python做自然語言處理

NLTK=Natural Language Toolkit



教育程度與意識形態



聽說教育程度愈高意識形態愈嚴重

可是教育程度如何衡量呢？

```
from nltk.book import text4  
print(len(set(text4))/len(text4)) # lexical diversity  
long_words=[w for w in set(text4) if len(w)>15]  
print(len(long_words)) # number of big words
```

斷句/詞(Tokenization)與統計

```
import nltk  
mytxt='This is a cat. That is a dog.'  
print(nltk.sent_tokenize(mytxt))  
text=nltk.Text(nltk.word_tokenize(mytxt))  
text.plot()  
from nltk.book import text4  
text4.dispersion_plot(['democracy','freedom','duties'])  
dist=nltk.FreqDist(text4)  
print(dist['freedom'])  
print([w for w in dist.keys() if dist[w]>1000])
```

有加權的詞頻:TF-IDF

term frequency–inverse document frequency

$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

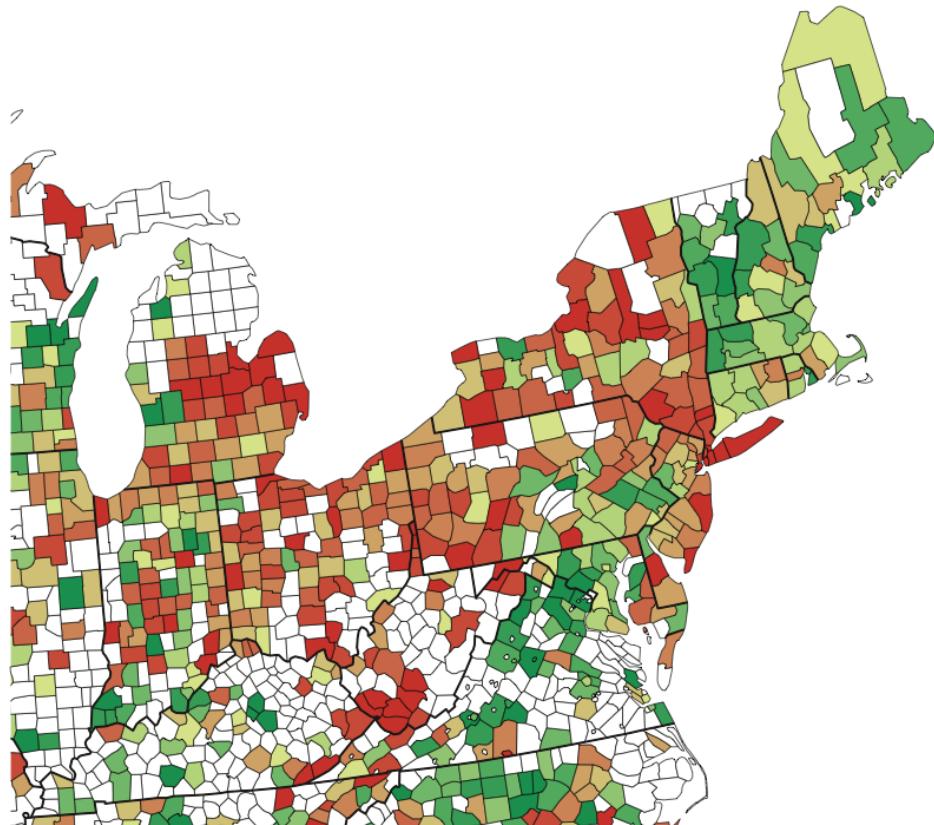
N = total number of documents

frequency	highest idf	tf * idf
32 the	1.00 tdt000077	3.20 orr
16 were	1.00 picknickers	2.81 charlotte
14 said	0.93 screaming	2.65 payne
12 and	0.93 timmy	2.48 dc
12 to	0.86 6thld	2.24 usair
11 a	0.80 orr	2.00 plane
10 of	0.78 1016	1.93 crash
9 at	0.76 bergen	1.74 bones
9 was	0.75 dripping	1.63 survivors
7 in	0.73 abrams	1.50 dripping
6 on	0.72 0419	1.49 wreckage
6 they	0.69 fuselage	1.35 dead
6 people	0.66 nc	1.29 hospitals
6 had	0.66 thunderstorm	1.27 airport
6 plane	0.66 payne	1.23 55

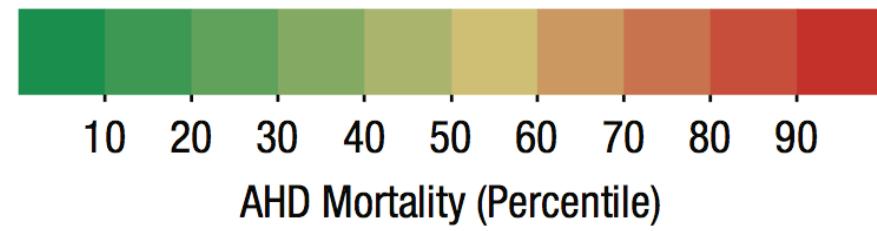
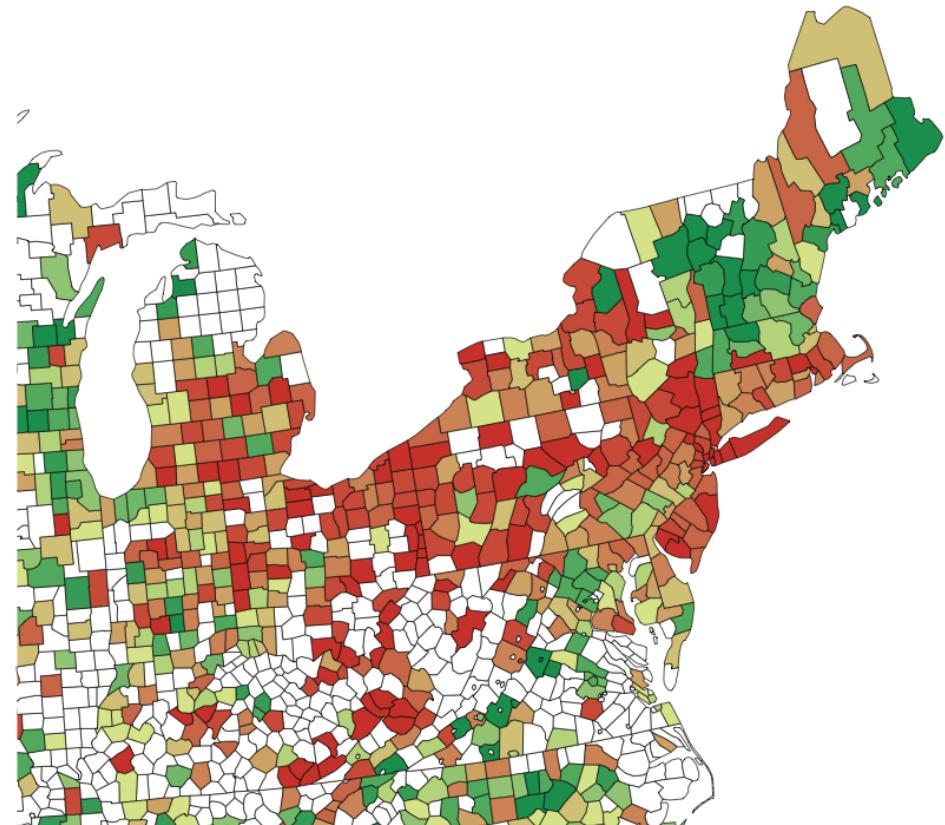
可用來建構文件向量=(詞1, 詞2,...)以計算文件相似度

心理學案例研究：負向情緒 & 心臟病

CDC-Reported AHD Mortality



Twitter-Predicted AHD Mortality



Eichstaedt et al.,
2015, *Psychol. Sci.*

情緒分析(Sentiment Analysis)



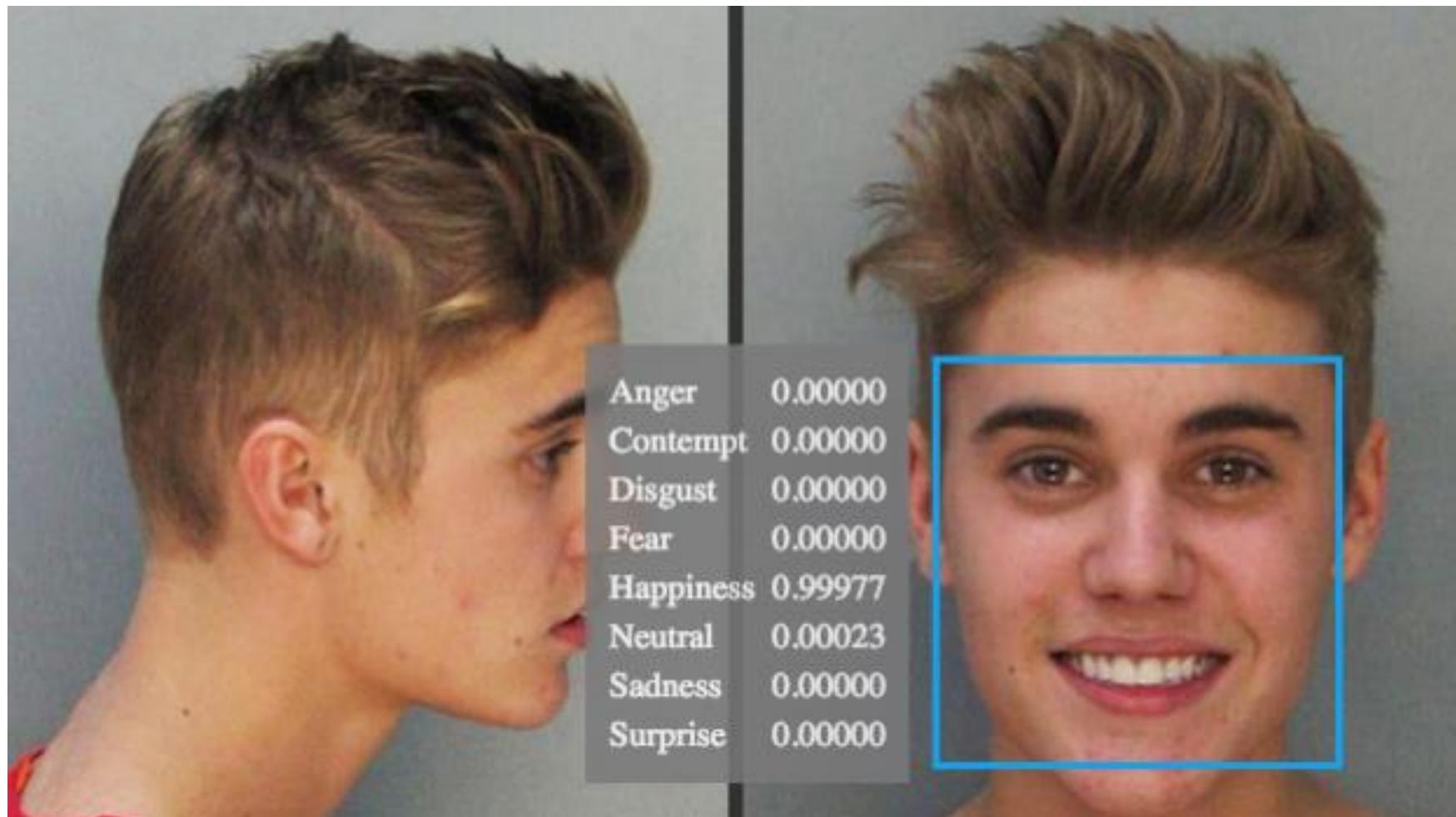
解法2：網路上先下載正面和負面情緒的字詞列表。對一篇文章算出正面字和負面字各有多少，若正面字數遠大於負面字數文章即為正面。

解法3：利用nltk中的機器學習(nltk.classify)來自動分類句子或文章為正面或負面，例如這個範例。

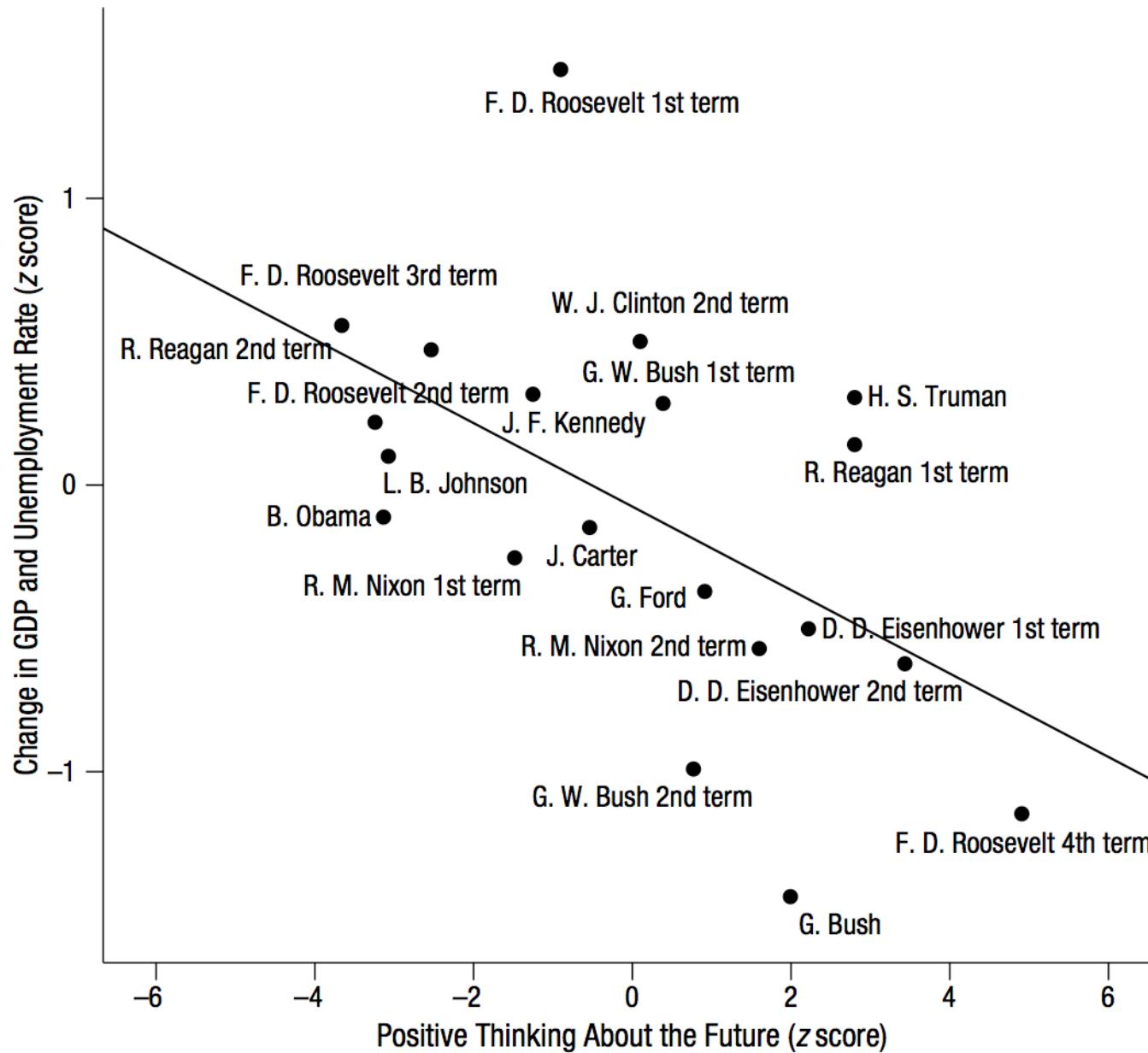
不只是正向或負向情緒

NRCWE將14,182個字分成Plutchik's 8個情緒類別

Stanford U也有做多情緒分析器



心理學案例研究:未來正向思考& GDP



Sevincer et al., 2015, *Psychol. Sci.*

Linguistic Inquiry & Word Count

LIWC有線上試用版



DISCOVER LIWC2015

LIWC2015 is the gold standard in computerized text analysis. Learn how the words we use in everyday language reveal our thoughts, feelings, personality, and motivations. Based on years of scientific research, LIWC2015 is more accurate, easier to use, and provides a broader range of social and psychological insights compared to earlier LIWC versions. Check it out.

[BUY NOW](#)

中文資料的處理:第一步要斷詞



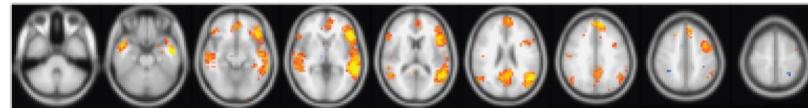
中研院中文斷詞系統
結巴中文分詞套件
中文版LIWC詞典 (demo)

Topic Modeling (LDA)

Topic 121 (68 docs)
regret, surprise, reasoning
arousal, learning



Topic 71 (115 docs):
narrative, discourse, comprehension,
memory, discourse_processing



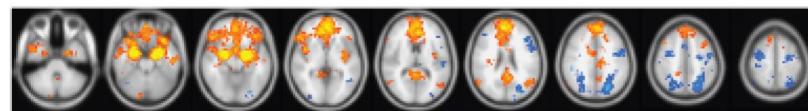
Topic 108 (128 docs):
empathy, pain, theory_of_mind,
awareness, facial_expression



Topic 61 (441 docs):
memory, working_memory, maintenance,
visual_working_memory,
spatial_working_memory



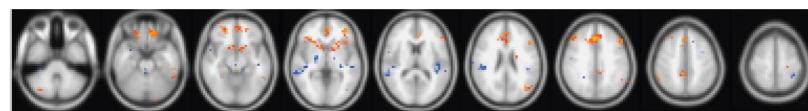
Topic 93 (495 docs):
emotion, valence, arousal, attention,
focus



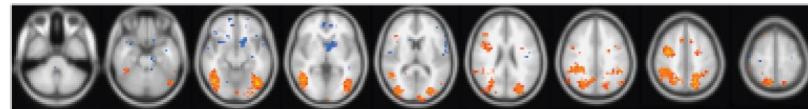
Topic 20 (497 docs):
action, movement, goal, context,
perception



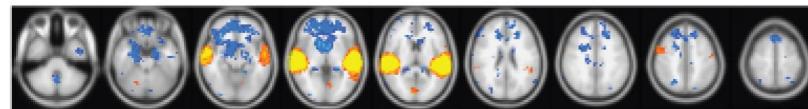
Topic 86 (519 docs):
decision, decision_making, choice,
fixation, uncertainty



Topic 43 (566 docs):
attention, focus, visual_attention,
fixation, attentional_resources



Topic 74 (769 docs):
auditory, perception, hearing,
attention, listening

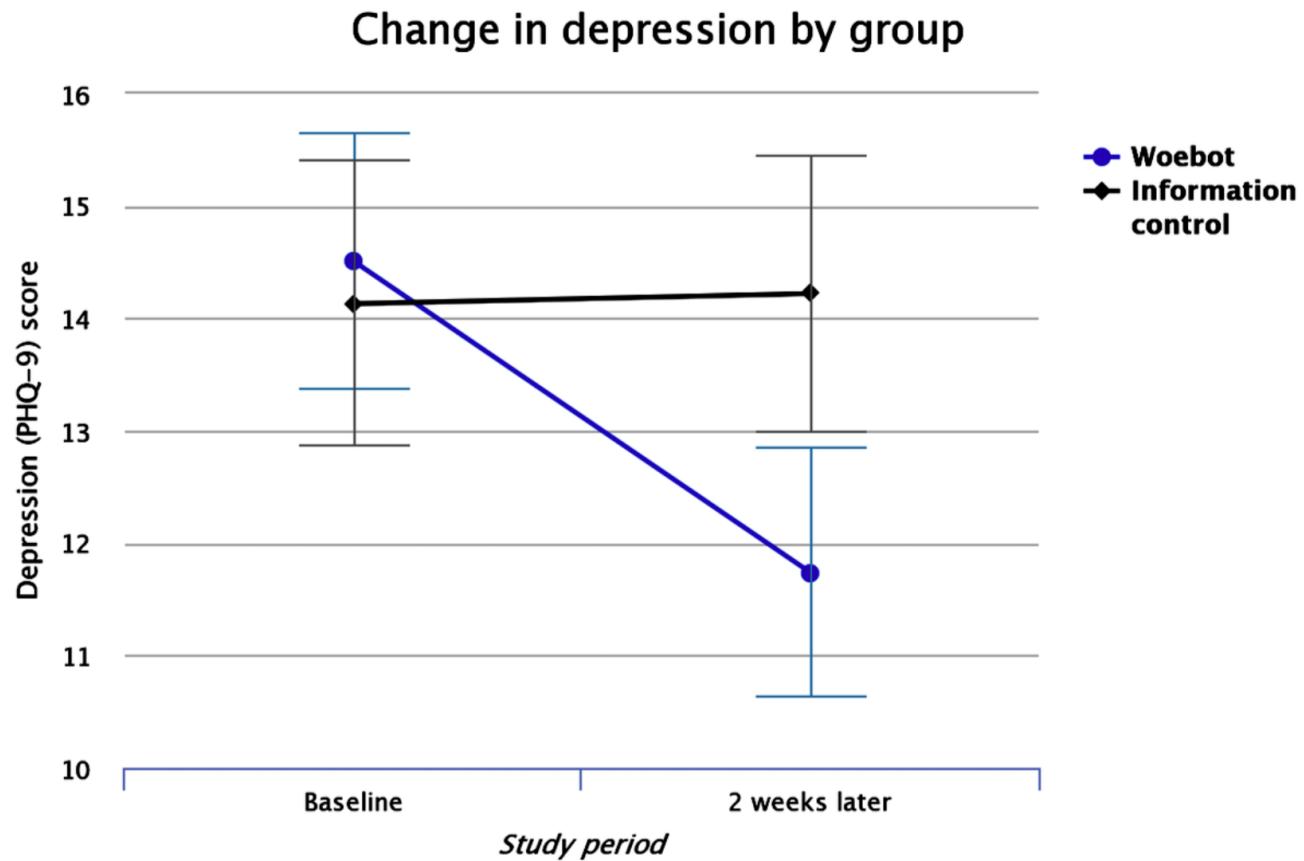
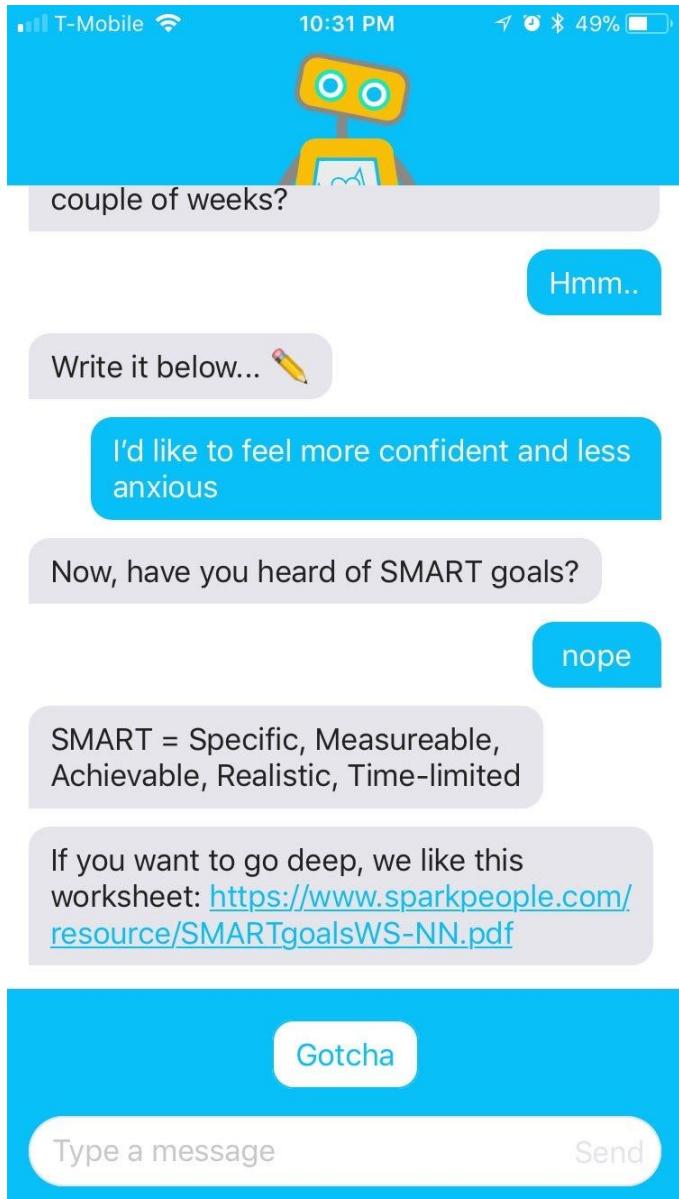


晚近NLP：詞/句向量→文向量

(Gensim/spaCy)

心理學案例研究: 聊天機器人

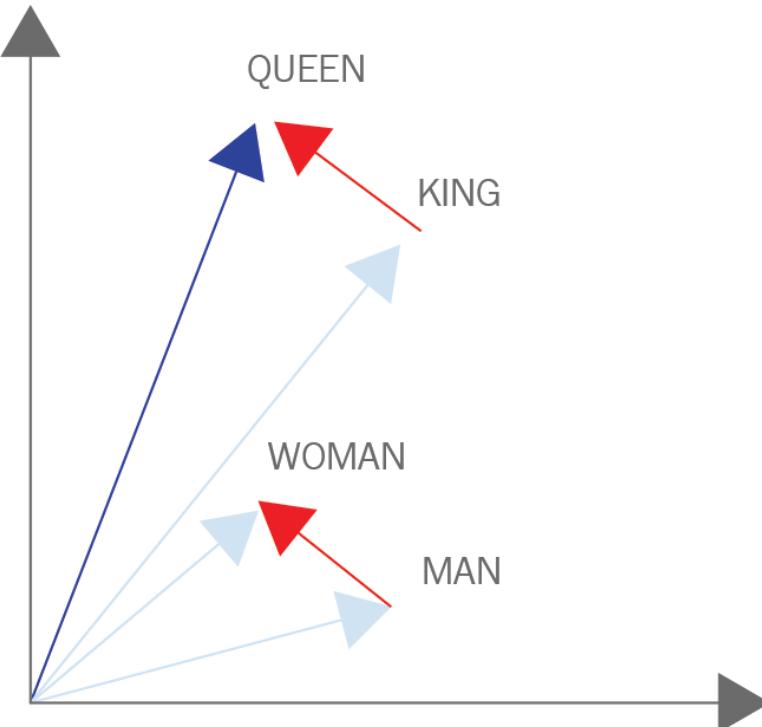
Woebot應用認知行為治療



Fuzzy/Semantic Match

Word2Vec輸出字在語意空間的向量表徵

So king + man - woman = queen!

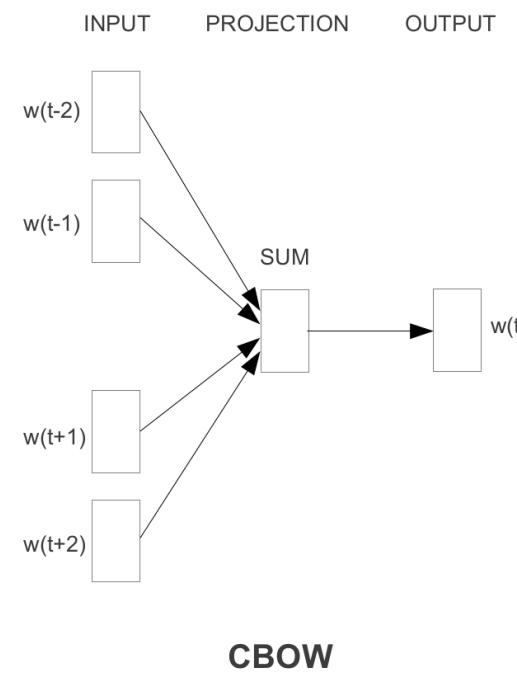
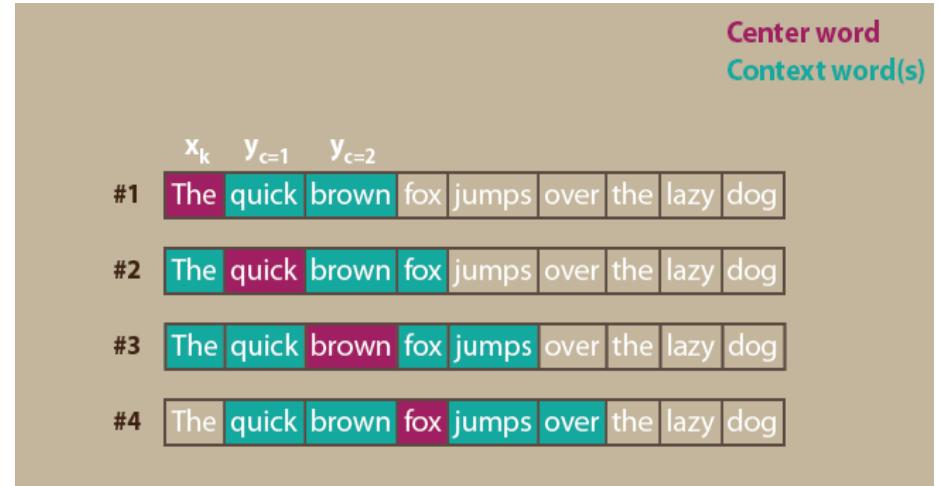
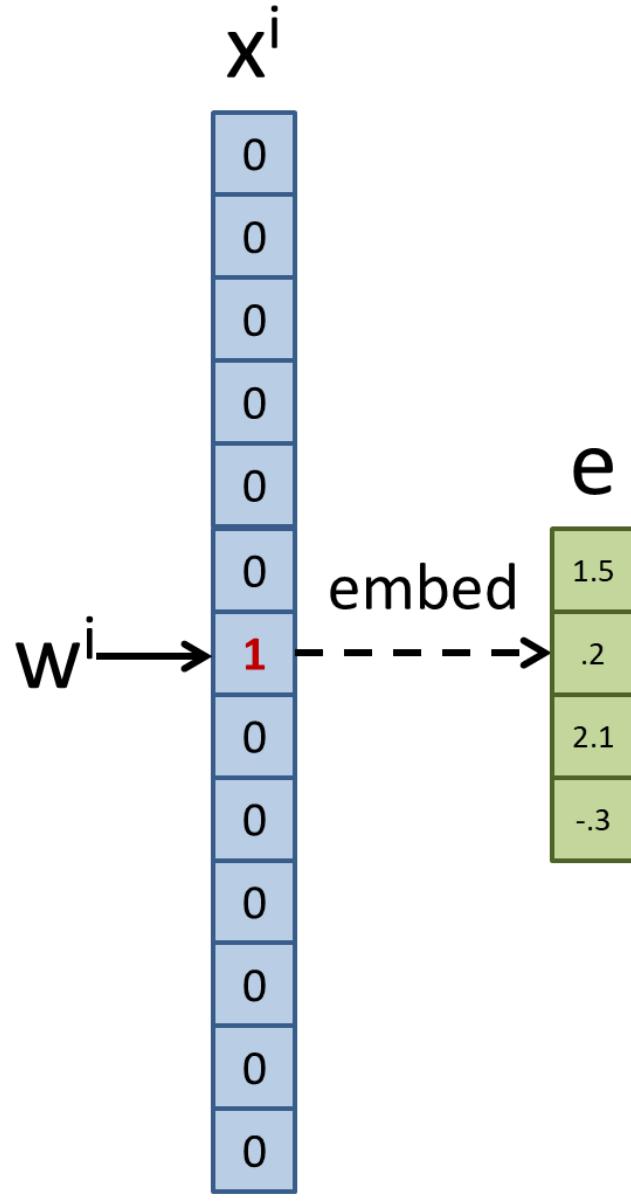


$$\text{"king"} + (\text{"woman"} - \text{"man"}) = \text{"queen"}$$

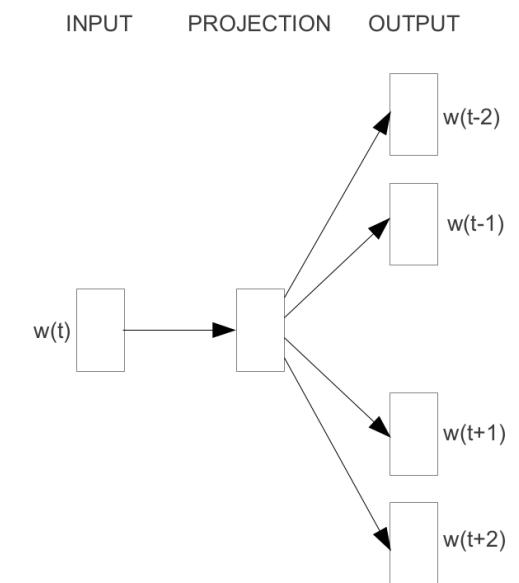
$$(\text{"good"} + \text{"best"}) / 2 = \text{"better"}$$

Word2Vec的原理

得到word vectors/embeddings有兩種方式

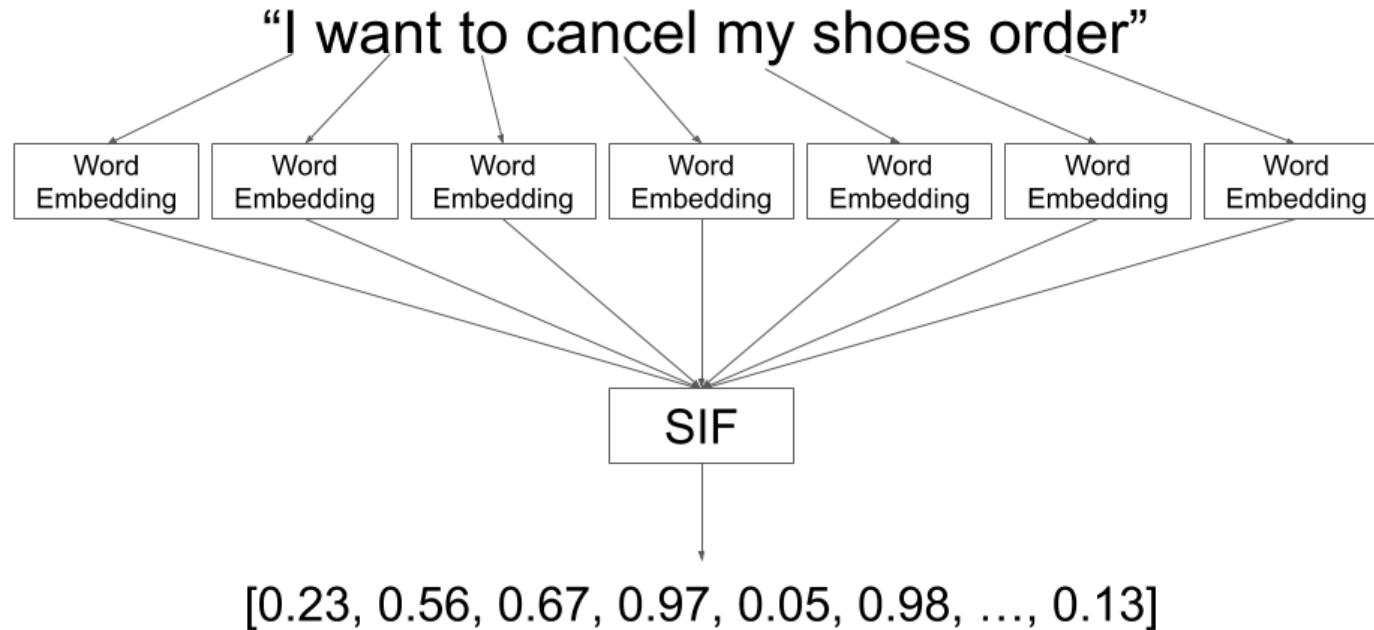


CBOW



Skip-gram

Sentence Embedding



Algorithm 1 Sentence Embedding

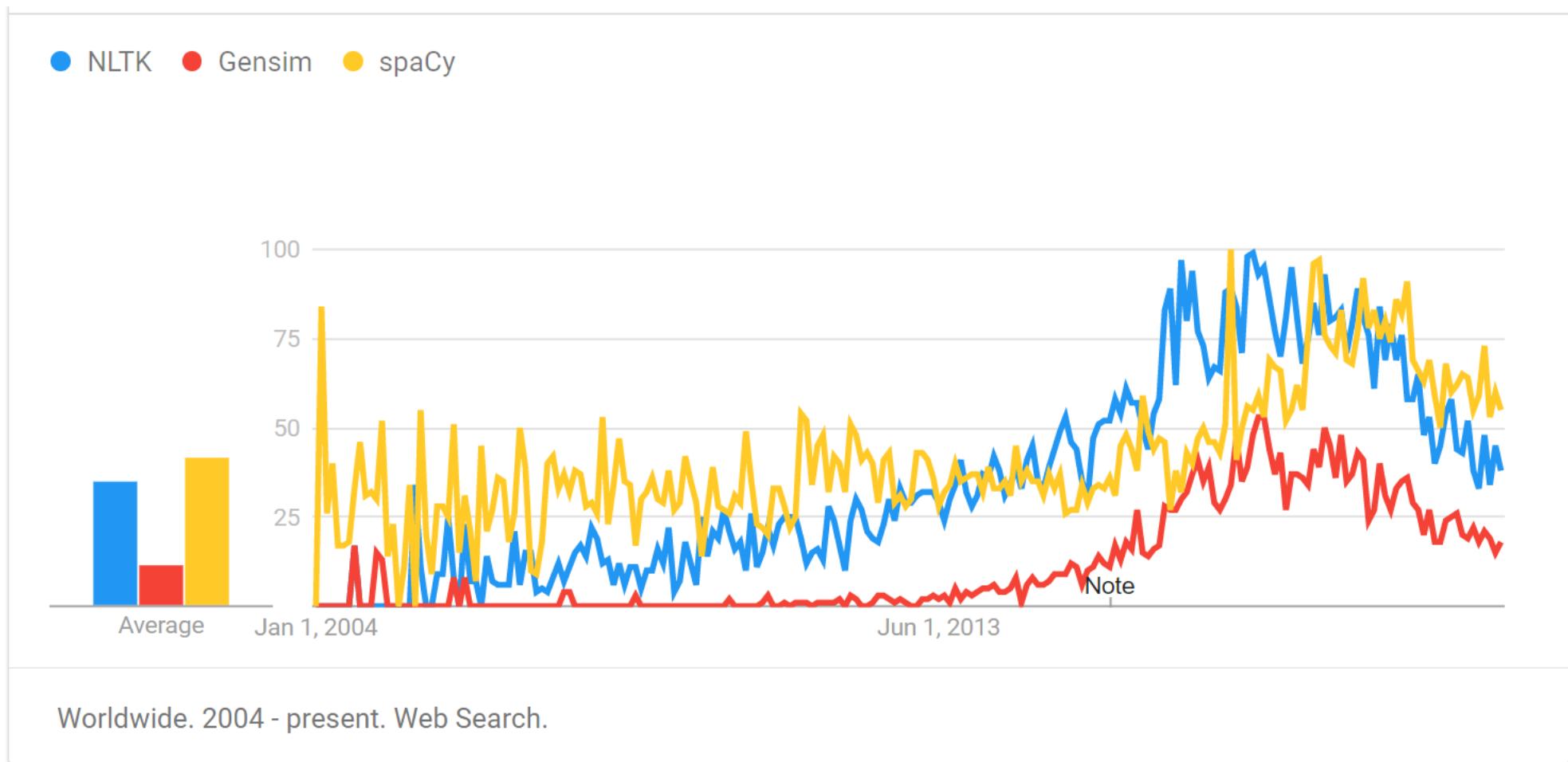
Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences \mathcal{S} , parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.

Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$

- 1: **for all** sentence s in \mathcal{S} **do**
- 2: $v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$
- 3: **end for**
- 4: Form a matrix X whose columns are $\{v_s : s \in \mathcal{S}\}$, and let u be its first singular vector
- 5: **for all** sentence s in \mathcal{S} **do**
- 6: $v_s \leftarrow v_s - uu^\top v_s$
- 7: **end for**

近代NLP套件

spaCy在NLP的功能上比較gensim完整



本週作業

進一步研究regex與topic modeling



1. 用urllib+re在臺大資工系網頁搜集教師 emails (4分)
2. 對2.5 Topic Modeling 的範例程式文本先過濾掉 stop words, 使5個主題能夠更加明確 (4分)

Game Over

