詞向量文本分類模型與機器學習

台師大通識教育課程

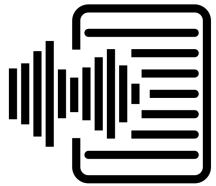
文本分析與程式設計

授課：卓騰語言科技 _ PeterWolf

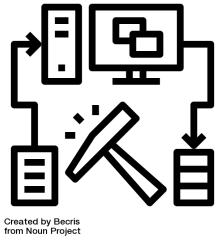
機器學習 (Machine Learning) 在 NLP 領域裡是一個「逐步失真」的過程

Language

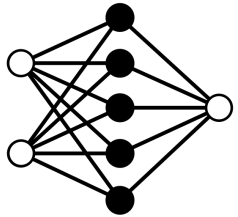
Text



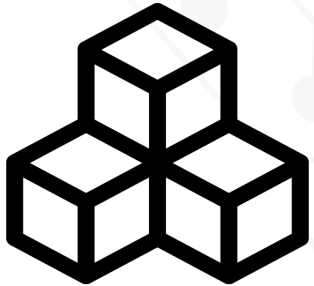
Preprocessing



Machine Learning



Language Model



ASR 喪失：
語氣、語調、語速

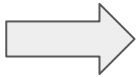
斷句喪失：
前後文、語境

NN/Vec 喪失：
文法、句構

LM 喪失：
邏輯、因果、知識

從「頻率」到「維度」

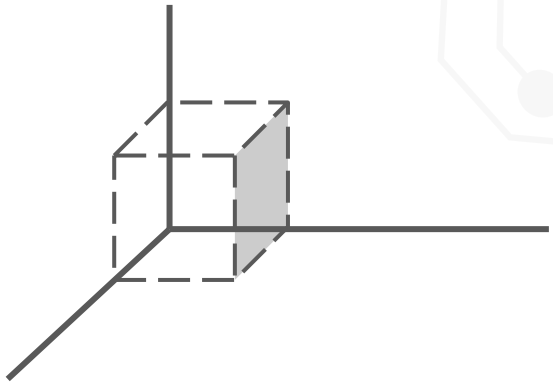
a. 李男養了三隻土狗...土狗咬傷婦人
b. 李男養了三位婦人...婦人咬傷土狗



a. 李男:1, 養了: 1, 三隻: 1, 土狗: 2, 婦人: 1
b. 李男:1, 養了: 1, 三位: 1, 土狗:1, 婦人: 2

	李男	養了	三隻	三位	土狗	婦人	...
a	1	1	1	0	1	0	...
b	1	1	0	1	0	1	...
c	...	...	...	...	...	...	...

a. [1, 1, 1, 0, 1, 0, ...]
b. [1, 1, 0, 1, 0, 1, ...]



維度！維度！更多維度！

bi-gram:

	李男	養了	三隻	三位	土狗	婦人
李男	0	0	0	0	0	0
養了	2	0	0	0	0	0
三隻	0	1	0	0	0	0
三位	0	1	0	0	0	0
土狗	0	0	1	0	0	0
婦人	0	0	0	1	0	0

另一個角度來思考維度！

bi-gram: 前後文

	李男	養了	三 _C1_	三 _C2_	土狗	婦人
李男	0	0	0	0	0	0
養了	2	0	0	0	0	0
三隻	0	1	0	0	0	0
三位	0	1	0	0	0	0
土狗	0	0	1	0	0	0
婦人	0	0	0	1	0	0

維度好像可以抓住「前後文」的關係呢！(喲嘿！)

https://sa.ylib.com/MagArticle.aspx?id=2773


John R. Firth

You should know a word by the company it keeps.

- J. R. Firth

(從一個字的上下文固定出現的元素就能學會這個字)

一個詞的是被它鄰近且一起出現的詞 /元素所定義的

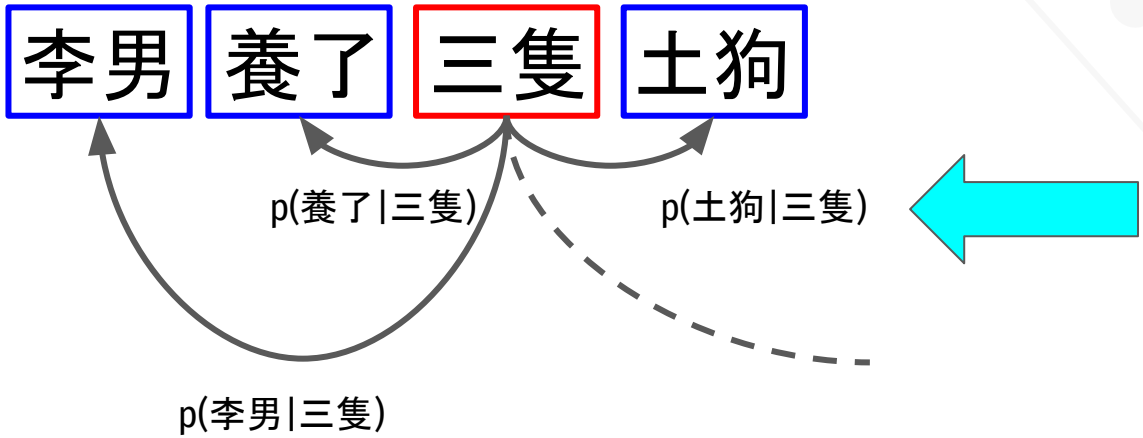
https://en.wikipedia.org/wiki/John_Rupert_Firth

https://zh.wikipedia.org/zh-tw/约翰·鲁伯特·弗斯

https://zh.wikipedia.org/wiki/泰卢固语

維度好像可以抓住「前後文」的關係呢！(喲嘿！)

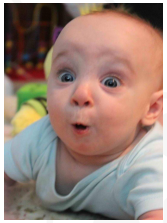
1. 假如我們擁有大量文體相似的語料(光是幾篇狗咬人的文章是不夠的)
2. 每個詞都能被它前後文算出向量，並用這個向量來表示它在維度空間的位置

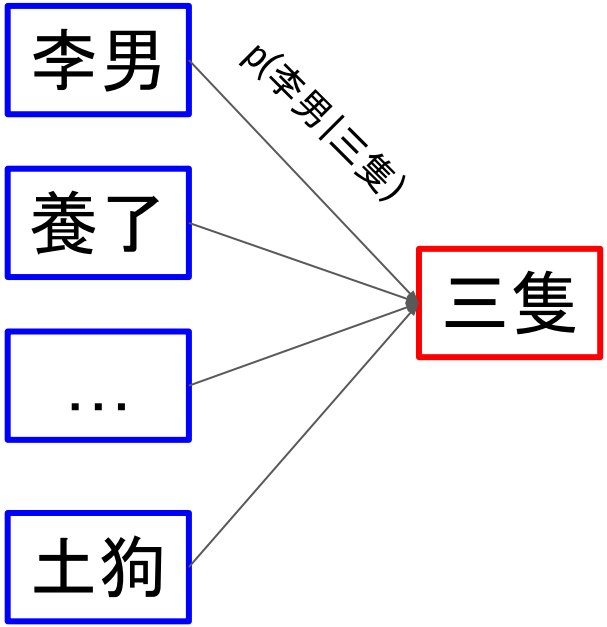


p(Word_Y|Word_X)：在 Word_X 的前提下，出現 Word_Y 的機率

兩種視角之一：CBOW (連續詞袋模型)

這！就是文法嗎！

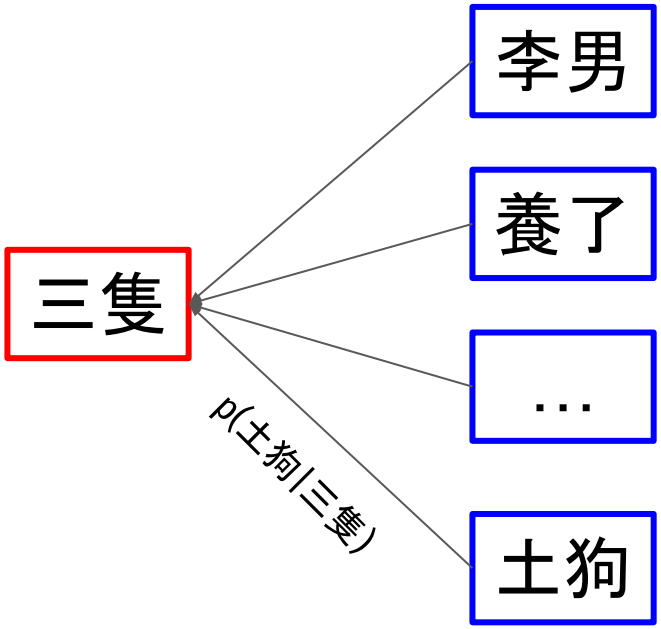




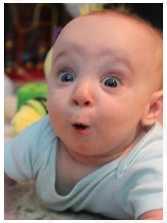
李男養了 _____ 土狗

別人的
三隻
幾條
混血的

兩種視角之二：Skip-gram (跳詞)



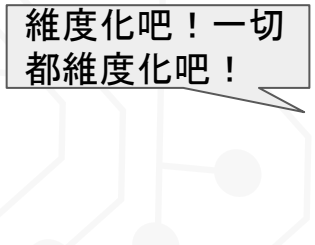
這！就是語意嗎！



三隻

____	____	____ …
李男	看見	小魚
昨天	賣掉	海豚
魚池裡	餵飽	小鳥牌
你家	養了	土狗

不只是 word 可以 2vec 哦！就連 sentence 也能把它 2vec 呢！



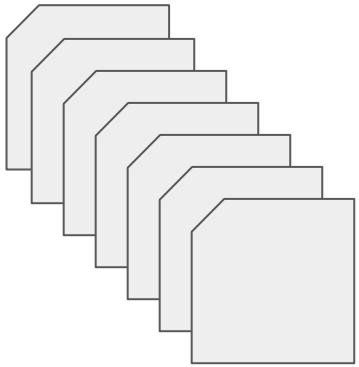
大野狼敲敲外婆家的門，它裝出小女孩的聲音：“外婆，我是小紅帽，我帶東西來看您了！

S1 S2 S3 S4 S5

p(Sent_Y|Sent_X)：在 Sent_X 的前提下，出現 Sent_Y 的機率

https://www.google.com/search?q=%22%E5%AE%83%E8%A3%9D%E5%87%BA%E5%B0%8F%E5%A5%B3%E5%AD%A9%E7%9A%84%E8%81%B2%E9%9F%B3%22&safe=strict&client=firefox-b-d&sxsrf=ALeKk032P8V95G68dFitmrFJxR3AtbklDg:1607334768043&ei=cPvNX5mVAoeB0wTosr7ADA&start=20&sa=N&ved=2ahUKEwiZjYnPzLvtAhWHwJQKHWiZD8g4ChDw0wN6BAgEEEQ&biw=1427&bih=738

把所有的文件都向量化後，就能計算兩篇文章「像不像？有多像？」



doc_x = [0.699847, 0.737547, 0.822377, 0.505353, 0.496978, ...]
doc_y = [0.718883, 0.544967, 0.486920, 0.659358, 0.629189, ...]
...

超棒！word2vec 就解決 NLP 的文法／語意問題了耶！

我們來看看真正的「中文」實驗吧！

https://medium.com/pyladies-taiwan/自然語言處理入門-word2vec小實作-f8832d9677c8

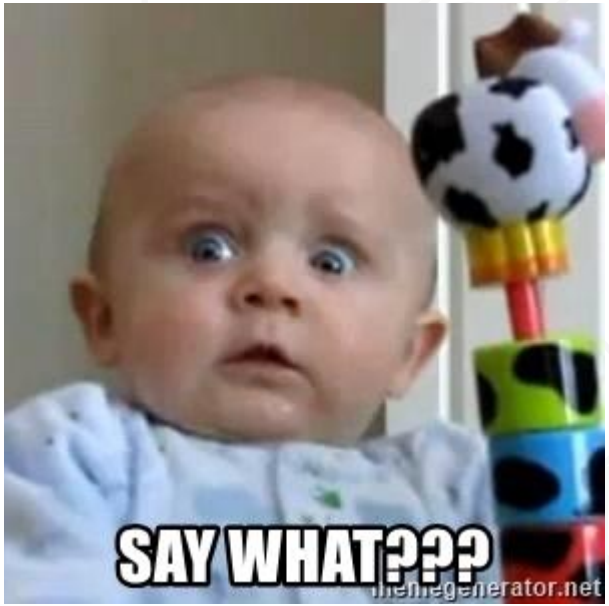
```
# 顯示空間距離相近的詞
model = word2vec.load('corpusWord2Vec.bin')
indexes = model.cosine(u'畢業') # 此字詞有出現在corpusWord2Vec.bin當中
for index in indexes[0]:
    print model.vocab[index]


# Result
畢業生
離校
考入
剛畢業
大四
學畢業
考上
開學
結業
放暑假
```

超棒！word2vec 就解決 NLP 的文法／語意問題了耶！

```
# 放入字詞: '寶寶'
indexes = model.cosine(u'寶寶')
for index in indexes[0]:
    print model.vocab[index]
# Result
寶寶的
小寶寶
孩子
準媽媽
寶寶在
胎兒
媽媽們
小孩
小孩子
媽媽
```

[寶寶] 的近義詞之一是 [準媽媽]



重新思考：word2vec 在中文裡，究竟代表了什麼？

I bought the skirt at your store yesterday,
The skirt that I bought at your store yesterday,
...

我昨天買的這件裙子
這件裙子我昨天買的
昨天我買的這件裙子
昨天這件裙子我買的
昨天我買的這件裙子
...

和印歐語系的語言相比之下，中文的詞綴非常貧乏，因此「前後上下文」沒有幾個會固定出現的元素。

#那剛剛的「三隻狗」...我故意拿量詞拐你的 :)

那 word2vec 為什麼有時候有用？有時候怪怪的？

1. 假如我們擁有大量文體相似的語料 (光是幾篇狗咬人的文章是不夠的)
2. 每個詞都能被它前後文算出向量，並用這個向量來表示它在維度空間的位置

1. 文體相似 (例如新聞體裁、朋友對話體裁、正式書信體裁..)，表示句型會類似。句型類似，表示「詞彙順序」變化不大。
2. 大量語料，表示「在變化不大的情況」下，模型幾乎已經涵蓋了所有可能的變化。如果語料的「量不夠大」，則模型沒辦法涵蓋所有可能的變化。如果語料的「文體不相似」，則變化太大，模型一樣無法涵蓋所有可能的變化。

於是 [寶寶] 的近義詞之一就會是 [準媽媽] 了！

Quiz:

課堂中說明了文本處理中「詞向量」、「句向量」的原理，請思考以下問題...。

1. 除了詞、句以外，是否有可能以「文件」做為一個向量？
2. 適用向量來解決文本分析的場景先決條件為何？
3. NLP 的內涵可概分為「語言問題」和「計算問題」。前者討論的是「呈現出語言系統的內部資訊」，後者聚焦的是「如何讓稀疏的語言(文字)系統成為可計算的連續系統」。請討論向量是否解決了語言和計算的問題？
4. 回想前面的「失真階梯」，請思考向量是否能處理「一字多義」、「詞性變化的轉品」或「破音字」的問題？為什麼？

Assignment: 小組作業，每組繳一份至你們的「組名目錄」即可

1. 從課程 github repo 中把課程中提供的week13 的目錄 git pull 下來。
2. 把 week13.py 改名為 **week13_分組隊名.py**
3. 在 **week13_分組隊名.py** 中，設計你的程式，利用gensim 完成以下指定規格：
   a. pip install gensim #安裝 gensim 機器學習模組
   b. 下載 300 維 CBOW 繁中詞向量 | wiki2019tw_word2vec_cbow_d300.zip | 2.3GB | https://drive.google.com/uc?export=download&id=1C085CLs4CtV5SvhOo2etoBppGXtwmqe5
      i. 模型來源：台灣自然語言處理與人工智慧交流社FB 社團
   c. 任意填入 10 個詞彙在 week13_分組隊名.py 的 "simLIST" 中，從上述模型裡取得和這10個詞彙向量最接近的前10 個字，存入 w2v_隊名.json 中。
   d. 儲存格式為 {"第一個詞": ["第一個最接近詞", "第二個最接近詞", ..."第十個最接近詞], ...}