Day 65 Word2Vec

gensim 使用簡介

本日知識點目標

目標
知識點

理解非監督學習的 “word2vec”的原理

獲得
知識點

完成今日課程後你應該可以了解

- gensim 的使用模式

什麼是Word2vec?

- NLP 裡面，最低階組成的是詞語，詞語組成句子，句子再組成段落、篇章、文檔, 其關係就如同是二元一次方程式:

  For example: f(x)->y

- 在NLP 中，把x 看做一個句子裡的一個詞語，y 是這個詞語的上下文詞語，那麼這裡的f，便是NLP 中經常出現的『語言模型』（language model），這個模型的目的，就是判斷(x,y) 這個樣本，是否符合自然語言的法則
- Word2vec 正是來源於這個思想
- word2vec 是在做一項**翻譯**的工作，把詞（word）轉換成電腦可以了解的模式（vector）.
- 利用vector 之間的距離關係來決定兩這的相依程度

什麼是Word2vec?

- 為什麼 word2vec 可以讀取出詞與詞的概念呢?

- 用一句話解釋就是他會把這個詞附近的相鄰詞考慮進來
- Word2Vec 是尋找單詞連續 embedding 的技術。通過閱讀大量的文本學習，並記憶哪些單詞傾向於相似的語境。訓練足夠多的資料後，詞彙表中的每個單詞會生成一個 300 維的向量，由意思相近的單詞構成。

把相關、片斷的字聯想一起

Ice cream（冰淇淋） ➔ dessert（甜點）
robot（機器人） ➔ autometically（自動的）
food（食物） ➔ upset（難過）➔ allergy（過敏）
no rainfall（沒有雨） ➔ farmers can’t（農夫不能）➔ good harvest（好的收成）

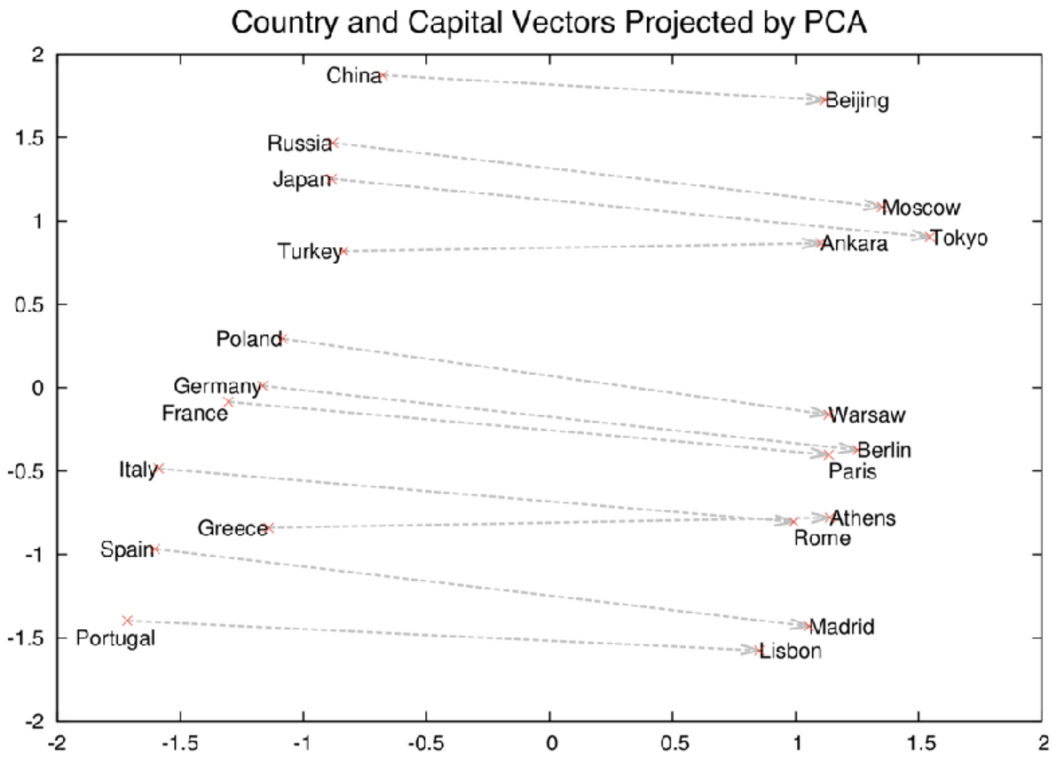
什麼是Word2vec?

- 使用技術：
- 第一種稱作 Continuous Bag Of Words（CBOW），此方法會利用上下文的詞來當作神經網路的輸入，最後預測這個目標的詞是什麼。
- 第二種則是 Skip-Gram 演算法，剛好跟第一種演算法相反，他所輸入的是當前的詞來預測這一段的文章上下文的詞

	MARY	IS	HUNGRY	HAPPY	FOR	APPLES	NOT	JOHN	HE	
“ Mary is hungry for apples.” →	1	1	1	0	1	1	0	0	0	→ [1,1,1,0,1,1,0,0,0]
“ John is happy he is not hungry for apples.” →	0	2	1	1	1	1	1	1	1	→ [0,2,1,1,1,1,1,1,1,]

將句子表示為詞袋。左邊為句子，右邊為對應的表示，向量中的每個數位（索引）代表一個特定的單詞

什麼是Word2vec?



圖片來源：google blog

- word2vec 使用 Cosine Similarity 來計算兩個詞的相似性，這是一個 -1 到 1 的數值，如果兩個詞完全一樣就是 1，像是「台灣」這個詞和他自身的 Cosine Similarity 就是 1）
- 而除了比較相似性以外， word2vec 還可已有**類推**這個概念，從引用至 google open source blog 的圖來看，我們可以看到國家的距離是彼此相近的，中國、俄國、日本．．．等，而相對應首都的距離也是相近的

How to 實現 “word2vec” 詞向量

- 一個詞的意涵跟他的左右鄰居很有關係，如何找出左右鄰居？
  - 就如同英文克漏字或是中文的填空數獨
- 新的問題是：左右鄰居有誰？
  - 在word2vec有一個概念叫windows
  - Windows size 的設定可以讓你知道找幾個鄰居

把相關、片斷的字聯想一起

小橋－流水－人家，古道－西風－瘦馬－斷腸人在天涯

小橋．流水－人家，古道－西風－瘦馬－斷腸人－在天涯

How to 實現 “word2vec” 詞向量

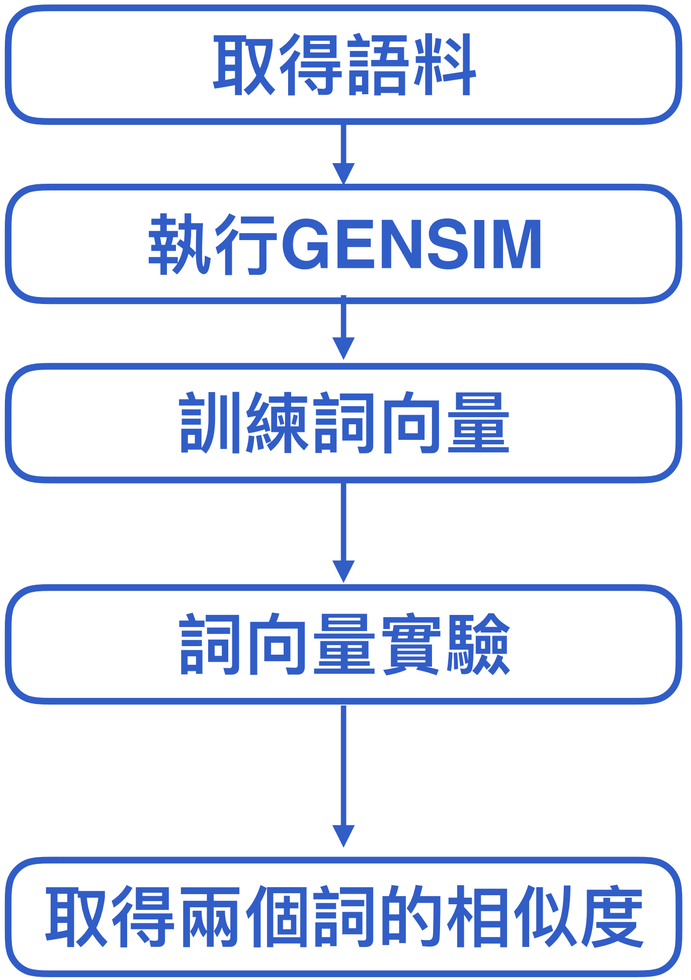
- 為了更好理解， 我們使用 GENSIM 來了解 word2vec

  1. 快速安裝

  easy install -U gensim
  pip install --upgrade gensim

  2. 相關模組

  Python >= 3.x
  NumPy >= 1.3
  SciPy >= 0.7



前述流程 / python程式 對照

```
import gensim, logging
from gensim.models import word2vec
```

```
import gensim, logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)

sentences = [['first', 'sentence'], ['second', 'sentence']]
# train word2vec on the two sentences
model = word2vec.Word2Vec(sentences, size=256, min_count=1, window=5, workers=4, sg=0)

# sg=0 表示COBW, sg=1 表示skip-gram
```

前述流程 / python程式 對照

- 參數說明
- sentences：當然了，這是要訓練的句子集，沒有他就不用跑了
- size：這表示的是訓練出的詞向量會有幾維
- alpha：機器學習中的學習率，這東西會逐漸收斂到 min_alpha
- sg：sg=1表示採用skip-gram，sg=0 表示採用cbow
- window：能往左往右看幾個字的意思
- workers：執行緒數目，
- min_count：若這個詞出現的次數小於min_count，那他就不會被視為訓練對象

- Note: 若是運行過程中碰到記憶體不足的問題, 可以把worker 的值設置在 4 以下

重要知識點複習

- 學習了 word2vec 的基本觀念，了解到把詞變成一個向量以後，可以根據語意做詞向量的加法減法得到新的詞向量

解題時間
It’s Your Turn

請跳出PDF至官網Sample Code＆作業
開始解題