Word2Vec-以 gensim 訓練中文詞向量

參考及引用資料來源

- [1] zake7749-使用 gensim 訓練中文詞向量
- [2] gensim/corpora/wikicorpus
- Word2Vec的簡易教學與參數調整指南
- zhconv
- jieba

In [1]:
```
%load_ext memory_profiler
# !pip install -q zhconv
```

確認相關 Packages

In [2]:
```
import os

# Packages
import gensim
import jieba
import zhconv
from gensim.corpora import WikiCorpus
from datetime import datetime as dt
from typing import List


if not os.path.isfile('dict.txt.big'):
    !wget https://github.com/fxsjy/jieba/raw/master/extra_dict/dict.txt.big
jieba.set_dictionary('dict.txt.big')

print("gensim", gensim.__version__)
print("jieba", jieba.__version__)
```

```
gensim 4.3.1
jieba 0.42.1
```

準備中文訓練文本

訓練文本來源: 維基百科資料庫

> 要訓練詞向量，第一步當然是取得資料集。由於 word2vec 是基於非監督式學習，**訓練集一定一定要越大越好，語料涵蓋的越全面，訓練出來的結果也會越漂亮**。[1]

- zhwiki-20210101-pages-articles.xml.bz2 (1.9 GB)

> ```
> wget "https://dumps.wikimedia.org/zhwiki/20210101/zhwiki-20210101-pages-articles.xml.bz2"
> ```

目前已經使用另一份 Notebook (維基百科中文語料庫 zhWiki_20210101) 下載好中文維基百科語料，並可以直接引用

In [3]:
```
ZhWiki = r"C:\Users\user\Downloads\zhwiki-20230501-pages-articles-multistream.xml.bz

# !dir -sh $ZhWiki
# !CertUtil $ZhWiki
# !FileType $ZhWiki
```

中文文本前處理

在正式訓練 `Word2Vec` 之前，其實涉及了文本的前處理，本篇的處理包括如下三點 (而實務上對應的不同使用情境，可能會有不同的前處理流程):

- 簡轉繁: zhconv
- 中文斷詞: jieba
- 停用詞

簡繁轉換

wiki 文本其實摻雜了簡體與繁體中文，比如「数学」與「數學」，這會被 word2vec 當成兩個不同的詞。[1]
所以我們在斷詞前，需要加上簡繁轉換的手續

以下範例使用了較輕量的 Package zhconv，
若需要更高的精準度，則可以參考 OpenCC

In [4]:
```
zhconv.convert("这原本是一段简体中文", "zh-tw")
```

Out[4]:
```
'這原本是一段簡體中文'
```

中文斷詞

使用 jieba `jieba.cut` 來進行中文斷詞，
並簡單介紹 jieba 的兩種分詞模式:

- `cut_all=False` **精確模式**，試圖將句子最精確地切開，適合文本分析；
- `cut_all=True` **全模式**，把句子中所有的可以成詞的詞語都掃描出來, 速度非常快，但是不能解決歧義；

而本篇文本訓練採用**精確模式** `cut_all=False`

In [5]:
```
seg_list = jieba.cut("我来到北京清华大学", cut_all=True)
print("Full Mode: " + "/ ".join(seg_list))  # 全模式

seg_list = jieba.cut("我来到北京清华大学", cut_all=False)
print("Default Mode: " + "/ ".join(seg_list))  # 精確模式
```

```
Building prefix dict from C:\Users\user\Downloads\dict.txt.big ...
Loading model from cache C:\Users\user\AppData\Local\Temp\jieba.u7157396f9872b4f171d
0922602b24c50.cache
```

```
Loading model cost 1.138 seconds.
Prefix dict has been built successfully.
Full Mode: 我/ 来到/ 北京/ 清华/ 清华大学/ 华大/ 大学
Default Mode: 我/ 来到/ 北京/ 清华大学
```

In [6]:
```
print(list(jieba.cut("中英夾雜的example，Word2Vec應該很interesting吧?")))
```

```
['中', '英', '夾雜', '的', 'example', '，', 'Word2Vec', '應該', '很', 'interesting',
'吧', '?']
```

引入停用詞表

停用詞就是像英文中的 **the,a,this**，中文的**你我他**，與其他詞相比顯得不怎麼重要，對文章主題也無關緊要的，
是否要使用停用詞表，其實還是要看你的應用，也有可能保留這些停用詞更能達到你的目標。[1]

- Is it compulsory to remove stop words with word2vec?
- The Effect of Stopword Filtering prior to Word Embedding Training

---

以下範例還是示範引入停用詞表，而停用詞表網路上有各種各樣的資源
剛好 `kaggle` ，環境預設有裝 spacy，
就順道引用 `spacy` 提供的停用詞表吧 (實務上stopwords 應為另外準備好且檢視過的靜態文檔)

In [7]:
```
import spacy

# 下載語言模組
spacy.cli.download("zh_core_web_sm")  # 下載 spacy 中文模組
spacy.cli.download("en_core_web_sm")  # 下載 spacy 英文模組

nlp_zh = spacy.load("zh_core_web_sm") # 載入 spacy 中文模組
nlp_en = spacy.load("en_core_web_sm") # 載入 spacy 英文模組

# 印出前20個停用詞
print('--\n')
print(f"中文停用詞 Total={len(nlp_zh.Defaults.stop_words)}: {list(nlp_zh.Defaults.stc
print("--")
print(f"英文停用詞 Total={len(nlp_en.Defaults.stop_words)}: {list(nlp_en.Defaults.stc
```

```
✓ Download and installation successful
You can now load the package via spacy.load('zh_core_web_sm')
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
--

中文停用詞 Total=1891: ['完成', '您是', '不仅...而且', '长话短说', '将', '将近', 'R ·
L ·', '倒是', '活', '哎', '着呢', '相反', '看看', '如下', '来讲', '饱', '为此', '换言
之', '来不及', '大不了'] ...
--
英文停用詞 Total=326: ['many', 'rather', 'put', 'latterly', 'except', 'go', 'still',
'unless', 'this', 'has', 'of', 'anyhow', 'hundred', 'two', 'call', 'none', 'bottom',
'your', 'please', 'out'] ...
```

In [8]:
```
STOPWORDS =  nlp_zh.Defaults.stop_words | \
             nlp_en.Defaults.stop_words | \
             set(["\n", "\r\n", "\t", " ", ""])
print(len(STOPWORDS))
```

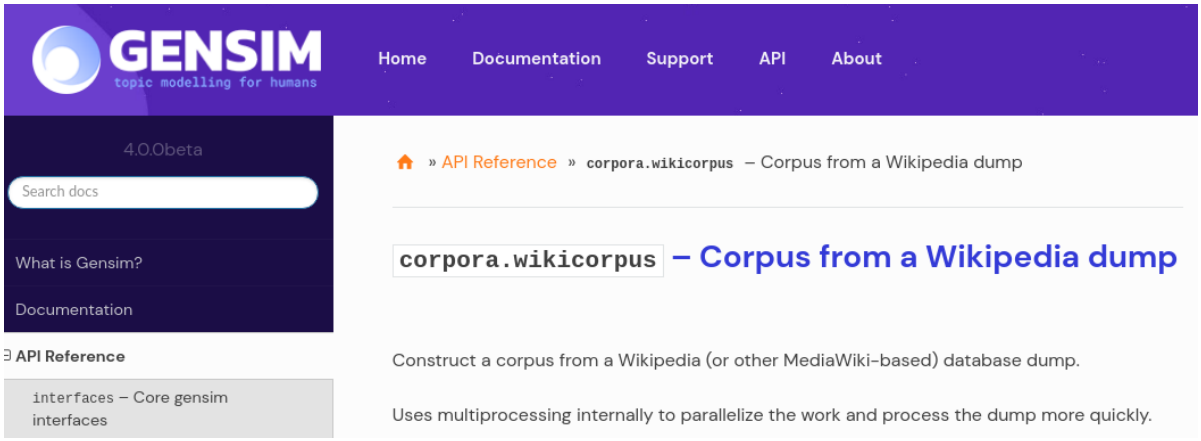
```
# 將簡體停用詞轉成繁體，擴充停用詞表
for word in STOPWORDS.copy():
    STOPWORDS.add(zhconv.convert(word, "zh-tw"))

print(len(STOPWORDS))
```

```
2222
3005
```

讀取 wiki 語料庫，並且進行前處理和斷詞

維基百科 (`wiki.xml.bz2`)下載好後，先別急著解壓縮，因為這是一份 xml 文件，裏頭佈滿了各式各樣的標籤，我們得先想辦法送走這群不速之客，不過也別太擔心， `gensim` 早已看穿了一切，藉由調用 wikiCorpus，我們能很輕鬆的只取出文章的標題和內容。[1]



[2]

Supported dump formats:

- `<LANG>wiki-<YYYYMMDD>-pages-articles.xml.bz2`
- `<LANG>wiki-latest-pages-articles.xml.bz2`

The documents are extracted on-the-fly, so that the whole (massive) dump can stay compressed on disk.

In [9]:
```
def preprocess_and_tokenize(
    text: str, token_min_len: int=1, token_max_len: int=15, lower: bool=True) -> Lis
    if lower:
        text  = text.lower()
    text = zhconv.convert(text, "zh-tw")
    return [
        token for token in jieba.cut(text, cut_all=False)
        if token_min_len <= len(token) <= token_max_len and \
            token not in STOPWORDS
    ]
```

In [10]:
```
print(preprocess_and_tokenize("歐幾里得，西元前三世紀的古希臘數學家，現在被認為是幾何之父
print(preprocess_and_tokenize("我来到北京清华大学"))
print(preprocess_and_tokenize("中英夾雜的example，Word2Vec應該很interesting吧?"))
```

```
['歐幾', '裡得', '西元前', '世紀', '古希臘', '數學家', '幾何', '父', '此畫', '拉斐爾']
['來到', '北京', '清華大學']
['中', '英', '夾雜', 'example', 'word2vec', 'interesting']
```

In [11]:

```
%%time
%%memit
from utils import preprocess_and_tokenize
from typing import List


print(f"Parsing {ZhWiki}...")
wiki_corpus = WikiCorpus(ZhWiki, tokenizer_func=preprocess_and_tokenize, token_min_l
```

```
Parsing C:\Users\user\Downloads\zhwiki-20230501-pages-articles-multistream.xml.bz
2...
C:\Users\user\anaconda3\ANA\lib\site-packages\gensim\utils.py:1333: UserWarning: det
ected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected %s; aliasing chunkize to chunkize_serial" % entity)
peak memory: 2185.99 MiB, increment: 1325.21 MiB
Wall time: 2h 8min 19s
```

初始化 `WikiCorpus` 後，能藉由 `get_texts()` 可迭代每一篇文章，它所回傳的是一個 `tokens list` ，我以空白符將這些 `tokens` 串接起來，統一輸出到同一份文字檔裡。這邊要注意一件事， `get_texts()` 受 `article_min_tokens` 參數的限制，只會回傳內容長度大於 **50** (default) 的文章。

- **article_min_tokens** *(int, optional)* – Minimum tokens in article. Article will be ignored if number of tokens is less.

秀出前 3 偏文章的前10 個 token

In [12]:

```
g = wiki_corpus.get_texts()
print(next(g)[:10])
print(next(g)[:10])
print(next(g)[:10])


# print(jieba.lcut("".join(next(g))[:50]))
# print(jieba.lcut("".join(next(g))[:50]))
```

```
['歐幾裡', '西元前', '三世', '紀的', '古希臘', '數學家', '現在', '認為', '幾何', '之父']
['蘇', '格拉', '底', '死', '雅克', '路易', '大衛', '所繪', '1787', '年']
['文學', '狹義上', '一種', '語言藝術', '語言', '文字', '為', '手段', '形象化', '客觀']
```

將處理完的語料集存下來，供後續使用

In [13]:

```
WIKI_SEG_TXT = "wiki_seg.txt"

generator = wiki_corpus.get_texts()

with open(WIKI_SEG_TXT, "w", encoding='utf-8') as output:
    for texts_num, tokens in enumerate(generator):
        output.write(" ".join(tokens) + "\n")

        if (texts_num + 1) % 100000 == 0:
            print(f"[{str(dt.now()):.19}] 已寫入 {texts_num} 篇斷詞文章")
```

```
[2023-05-12 10:36:21] 已寫入 99999 篇斷詞文章
[2023-05-12 10:50:00] 已寫入 199999 篇斷詞文章
```

```
[2023-05-12 11:02:32] 已寫入 299999 篇斷詞文章
[2023-05-12 11:14:47] 已寫入 399999 篇斷詞文章
[2023-05-12 11:27:53] 已寫入 499999 篇斷詞文章
[2023-05-12 11:38:59] 已寫入 599999 篇斷詞文章
[2023-05-12 11:50:57] 已寫入 699999 篇斷詞文章
[2023-05-12 12:03:26] 已寫入 799999 篇斷詞文章
```

訓練 Word2Vec

In [15]:

```
%%time

from gensim.models import word2vec
import multiprocessing

max_cpu_counts = multiprocessing.cpu_count()
word_dim_size = 300  #  設定 word vector 維度
print(f"Use {max_cpu_counts} workers to train Word2Vec (dim={word_dim_size})")


# 讀取訓練語句
sentences = word2vec.LineSentence(WIKI_SEG_TXT)

# 訓練模型
model = word2vec.Word2Vec(sentences, vector_size=word_dim_size, workers=max_cpu_coun

# 儲存模型
output_model = f"word2vec.zh.{word_dim_size}.model"
model.save(output_model)
```

```
Use 8 workers to train Word2Vec (dim=300)
Wall time: 1h 15min 30s
```

儲存的模型總共會產生三份檔案

In [17]:

```
! dir word2vec.zh*
```

```
 磁碟區 C 中的磁碟是 OS
 磁碟區序號:  FC67-B6D0

 C:\Users\user\Downloads 的目錄

2023/05/12  下午 03:31        58,905,889 word2vec.zh.300.model
2023/05/12  下午 03:31     2,161,575,728 word2vec.zh.300.model.syn1neg.npy
2023/05/12  下午 03:30     2,161,575,728 word2vec.zh.300.model.wv.vectors.npy
               3 個檔案   4,382,057,345 位元組
               0 個目錄  10,616,492,032 位元組可用
```

In [19]:

```
!dir /s -sh word2vec.zh*
```

```
 磁碟區 C 中的磁碟是 OS
 磁碟區序號:  FC67-B6D0

 C:\Users\user\Downloads 的目錄

2023/05/12  下午 03:31        58,905,889 word2vec.zh.300.model
2023/05/12  下午 03:31     2,161,575,728 word2vec.zh.300.model.syn1neg.npy
2023/05/12  下午 03:30     2,161,575,728 word2vec.zh.300.model.wv.vectors.npy
               3 個檔案   4,382,057,345 位元組

     檔案數目總計:
```

```
3 個檔案    4,382,057,345 位元組
0 個目錄   11,326,091,264 位元組可用
```

查看模型以及詞向量實驗

模型其實就是巨大的 Embedding Matrix

In [20]:
```
print(model.wv.vectors.shape)
model.wv.vectors
```

Out[20]:
```
(1801313, 300)
array([[ 3.22756481e+00, -6.68628037e-01, -6.49345806e-03, ...,
        -7.24847853e-01,  1.96346617e+00, -1.05247903e+00],
       [ 1.79310417e+00, -2.84394592e-01,  4.04559463e-01, ...,
        -1.48417127e+00,  1.86413646e+00,  4.32749063e-01],
       [ 2.52263713e+00,  7.74057686e-01, -1.28231728e+00, ...,
        -1.32995212e+00,  1.23376107e+00,  3.30840766e-01],
       ...,
       [ 4.03458513e-02,  4.77395765e-02, -1.92876637e-03, ...,
        -2.75534410e-02, -1.66336279e-02,  2.65944358e-02],
       [ 7.07197413e-02, -5.00116264e-04, -1.56356003e-02, ...,
        -2.48628911e-02, -3.98174226e-02,  6.40553469e-03],
       [ 2.25636158e-02,  5.90194426e-02, -1.25466725e-02, ...,
         6.36959542e-03, -1.44702541e-02,  1.54394777e-02]], dtype=float32)
```

收錄的詞彙

In [25]:
```
print(f"總共收錄了 {len(model.wv.index_to_key)} 個詞彙")

print("印出 20 個收錄詞彙:")
print(list(model.wv.index_to_key[:10]))
```

```
總共收錄了 1801313 個詞彙
印出 20 個收錄詞彙:
['年', '月', '日', '於', '為', '「', '與', '後', '臺', '中']
```

詞彙的向量

In [26]:
```
vec = model.wv['數學家']
print(vec.shape)
vec
```

Out[26]:
```
(300,)
array([-1.940246  ,  1.6505309 ,  1.0854734 ,  1.401144  , -0.25091517,
        1.0239282 , -1.3820399 , -2.811773  ,  0.3111333 , -0.6596898 ,
       -1.5042866 , -0.7564088 , -0.8212461 , -0.4765307 , -1.2927719 ,
       -0.8531906 , -0.7075069 , -0.8716696 ,  0.2739434 , -1.1815969 ,
        1.7641044 , -1.5286896 , -0.51143724, -1.1281426 ,  1.0585376 ,
        0.21238711, -0.8038455 ,  0.87532294, -0.27359983,  1.1084766 ,
       -0.5124968 ,  1.7217313 ,  0.83563304,  2.1702962 , -0.02430506,
       -1.0999883 ,  3.2630434 , -0.10428905, -1.6954764 ,  0.6983506 ,
        3.953743  , -1.3603883 ,  1.2077994 ,  0.05053153,  1.0036622 ,
       -0.60440135, -1.1242665 ,  0.16520737, -3.0259433 ,  0.06720125,
        1.4082462 ,  1.9589189 , -0.90282786, -0.93939465,  0.7188142 ,
       -2.3146715 , -0.7964    , -1.3582488 , -0.35489583,  0.9129189 ,
       -1.8613447 ,  1.4618058 ,  2.9617126 ,  0.65916586,  0.02952656,
       -0.09400466, -0.76651955, -1.1972091 , -0.6028393 ,  0.7808874 ,
       -3.0506215 ,  1.4372452 , -1.4220777 ,  2.9107502 , -2.4384196 ,
       -1.0207781 , -0.10364573,  1.6433896 ,  1.718988  , -1.1150713 ,
       -0.64721453,  0.7161948 ,  1.1266543 ,  1.287269  , -4.9966283 ,
        1.9730465 , -2.0460439 , -3.432815  , -2.9666343 ,  1.0949254 ,
```

```
 0.77363306, -0.28423   , -0.5173404 ,  1.0718031 , -0.2661315 ,
-1.1332496 ,  0.06595913,  0.54648787, -0.37642437,  1.0381314 ,
 0.4728064 ,  0.36000407,  2.1982188 , -1.844111  , -0.8546902 ,
 0.5864991 , -1.2206031 ,  3.0158632 , -1.3531069 ,  1.4690264 ,
-0.58305544,  2.3401315 ,  0.5911433 ,  0.19645812, -1.1239351 ,
-0.02848452,  2.5594761 ,  1.4559128 ,  1.2846568 ,  2.1532667 ,
-0.8027333 , -0.17942132, -1.3279808 , -2.0796251 , -2.686797  ,
 0.92693543,  0.85263693,  1.8562212 , -1.082946  , -2.8691297 ,
 0.37992612, -2.5201185 , -1.0021503 ,  0.74461406,  0.91014355,
-1.4353482 ,  1.2825133 , -0.2876428 ,  2.496     , -1.1210824 ,
 1.0493466 , -2.221758  ,  1.600431  ,  0.6199657 , -0.9636711 ,
-2.783084  ,  0.36532047,  0.8983542 ,  1.45812   ,  2.052321  ,
-0.63695633,  0.59602165,  2.1318226 , -0.9525817 ,  0.2252669 ,
 0.45255038,  1.813835  , -1.0538371 ,  0.7072578 ,  1.7323018 ,
-0.25840056,  2.6640933 ,  0.76362807, -0.6317767 ,  1.0654871 ,
-1.1880298 ,  0.73670393, -0.02614931,  0.8435081 , -0.44232264,
 1.0037297 ,  1.5836929 , -0.3229909 , -0.63948786, -0.615166  ,
-0.5778    , -3.0420344 ,  0.42951688,  2.6018546 , -0.4208729 ,
-1.1971543 , -0.4862916 , -1.8372937 , -0.6139389 ,  0.22883338,
 1.26013   , -0.80233634, -1.5332932 ,  0.5231704 ,  1.7358515 ,
 1.0166514 , -1.609746  ,  1.9978428 ,  1.3754689 ,  0.3402466 ,
 2.3270843 , -1.2117224 ,  0.63779926, -0.05299924, -0.6062632 ,
 1.574506  , -1.2792199 ,  1.9122884 ,  0.404354  , -0.7725193 ,
 1.1717694 , -0.70245034,  1.2745806 ,  0.47769973,  0.65867776,
 0.7459386 , -0.04760182,  3.674368  ,  1.0253173 ,  0.8673194 ,
 2.7313068 , -1.4925879 ,  2.2661777 , -1.7365729 , -0.38202238,
-0.34893307, -0.67333186,  0.6758014 ,  0.6349169 ,  3.9284813 ,
-1.2497529 ,  1.340793  ,  0.8572792 ,  1.9608026 ,  2.8331316 ,
 0.8345332 ,  0.24826375,  1.6219624 , -0.8535226 , -2.9454963 ,
-1.8617648 ,  0.04777721,  2.4145474 , -0.5789118 ,  1.0621003 ,
 0.42721993,  1.6157753 ,  3.828479  , -1.2322401 ,  1.4775805 ,
 1.6284382 , -0.12348561, -1.0513303 , -0.29652685,  2.3136814 ,
-1.5035198 , -0.46769157,  0.01604326,  1.6639165 ,  2.3916535 ,
-2.1925159 , -1.6507769 , -2.3505998 , -0.50942475,  2.5095205 ,
-0.06022272,  0.3151313 ,  0.99926656,  0.12229704, -0.4678139 ,
-4.5844393 ,  0.5321876 , -0.19930014, -2.7562883 , -0.02521963,
 0.4746354 , -2.4037929 , -1.4276005 , -0.71650535,  2.821137  ,
-0.5100718 , -0.0873304 , -0.87296677,  1.6345754 ,  1.1863828 ,
 1.653123  , -2.1689625 , -0.8332281 ,  0.55436975,  0.80119807,
 0.42251858, -1.9258693 ,  1.5305592 ,  0.33653846,  2.4528265 ,
 0.6529142 , -3.8513498 , -3.045313  , -0.1794811 , -0.97713035,
-0.35049215, -1.4884567 , -0.96047413,  0.1682047 , -0.7605489 ],
dtype=float32)
```

沒見過的詞彙

In [27]:

```
word = "這肯定沒見過 "

# 若強行取值會報錯
try:
    vec = model.wv[word]
except KeyError as e:
    print(e)
```

```
"Key '這肯定沒見過 ' not present"
```

查看前 10 名相似詞

`model.wv.most_similar` 的 `topn` 預設為 10

In [28]:

```
model.wv.most_similar("飲料", topn=10)
```

Out[28]:
```
[('飲品', 0.8262518048286438),
 ('果汁', 0.670158326625824),
 ('瓶裝', 0.6659455895423889),
 ('含酒精', 0.6587675213813782),
 ('酒類', 0.645626962184906),
 ('罐裝', 0.644216775894165),
 ('軟飲料', 0.6433597207069397),
 ('酸奶', 0.6340711116790771),
 ('無糖', 0.632826566696167),
 ('橙汁', 0.6287429332733154)]
```

In [29]:
```
model.wv.most_similar("car")
```

Out[29]:
```
[('truck', 0.7084474563598633),
 ('seat', 0.6935095191001892),
 ('tikita', 0.6852614879608154),
 ('motorcycle', 0.6626434922218323),
 ('chevrolet', 0.653346598148346),
 ('wagon', 0.6471164226531982),
 ('saloon', 0.6429990530014038),
 ('jeep', 0.6407321095466614),
 ('cab', 0.6400420069694519),
 ('tourer', 0.6365245580673218)]
```

In [30]:
```
model.wv.most_similar("facebook")
```

Out[30]:
```
[('臉書', 0.8122718930244446),
 ('instagram', 0.7546572089195251),
 ('專頁', 0.7438607811927795),
 ('貼文', 0.7278022766113281),
 ('面書', 0.7024548053741455),
 ('twitter', 0.6924057006835938),
 ('推特', 0.6915072202682495),
 ('臉書粉', 0.6865820288658142),
 ('臉書上', 0.6790109276771545),
 ('粉絲團', 0.6705224514007568)]
```

In [38]:
```
model.wv.most_similar("欺騙")
```

Out[38]:
```
[('欺瞞', 0.673052191734314),
 ('揭穿', 0.6688473224639893),
 ('愚弄', 0.6669619679450989),
 ('騙', 0.6602681875228882),
 ('所騙', 0.6532294154167175),
 ('冒充', 0.6476141810417175),
 ('要脅', 0.6459336280822754),
 ('拆穿', 0.6436660885810852),
 ('騙取', 0.6334155201911926),
 ('誤信', 0.6321043968200684)]
```

In [32]:
```
model.wv.most_similar("合約")
```

Out[32]:
```
[('新合約', 0.7732728719711304),
 ('合同', 0.755433976650238),
 ('合約將', 0.751802921295166),
 ('年合約', 0.745762050151825),
 ('簽約', 0.7218555212020874),
 ('續約', 0.7128554582595825),
 ('合約並', 0.697862446308136),
 ('其合約', 0.6487120985984802),
```

```
('合約期', 0.6437252759933472),
('租約', 0.6310086846351624)]
```

計算 **Cosine** 相似度

In [33]:
```
model.wv.similarity("連結", "鍵接")
```

Out[33]:
```
0.7429394
```

In [34]:
```
model.wv.similarity("連結", "陰天")
```

Out[34]:
```
0.0104204295
```

讀取模型

In [35]:
```
print(f"Loading {output_model}...")
new_model = word2vec.Word2Vec.load(output_model)
```

```
Loading word2vec.zh.300.model...
```

In [36]:
```
model.wv.similarity("連結", "陰天") == new_model.wv.similarity("連結", "陰天")
```

Out[36]:
```
True
```