學號:B0928031

姓名:鄭茹云

學系:人工智慧三甲

連結: https://colab.research.google.com/drive/1ixtWFodsla92zgMZtIz_ATeKKbaufEQw?usp=share_link

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

```
#!pip  install  -q  memory_profiler
```

```
  WARNING: The script mprof.exe is installed in 'C:\Users\layla\AppData\Roaming\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

[notice] A new release of pip available: 22.3 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
%load_ext  memory_profiler
#!pip  install  -q  zhconv
```

確認相關 Packages

```
#!pip  install  gensim
```

```
Defaulting to user installation because normal site-packages is not writeable
                                     39-win_amd64.whl (24.0 MB)
     ---------------------------------------- 24.0/24.0 MB 20.5 MB/s eta 0:00:00
Collecting smart-open>=1.8.1
  Downloading smart_open-6.3.0-py3-none-any.whl (56 kB)
     ---------------------------------------- 56.8/56.8 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: scipy>=1.7.0 in c:\users\layla\appdata\roaming\python\python39\site-packages (from gensim) (1.10.1)
Requirement already satisfied: numpy>=1.18.5 in c:\programdata\anaconda3\lib\site-packages (from gensim) (1.23.3)
Installing collected packages: smart-open, gensim
Successfully installed gensim-4.3.1 smart-open-6.3.0

[notice] A new release of pip available: 22.3 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import  os
import  urllib.request
import  gensim
import  jieba
import  zhconv
from  gensim.corpora  import  WikiCorpus
from  datetime  import  datetime  as  dt
from  typing  import  List

dict_file  =  'dict.txt.big'
if  not  os.path.isfile(dict_file):
        url  =  'https://github.com/fxsjy/jieba/raw/master/extra_dict/dict.txt.big'
        urllib.request.urlretrieve(url,  dict_file)

jieba.set_dictionary(dict_file)

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

```
import os
import hashlib

ZhWiki = "zhwiki-20230501-pages-articles.xml.bz2"

'''
!du -sh $ZhWiki
!md5sum $ZhWiki
!file $ZhWiki
'''
# Calculate file size
file_size = os.path.getsize(ZhWiki)
print(f"File size: {file_size / (1024 * 1024)} MB")

# Calculate MD5 checksum
md5_hash = hashlib.md5()
with open(ZhWiki, "rb") as file:
    for chunk in iter(lambda: file.read(4096), b""):
        md5_hash.update(chunk)
md5_checksum = md5_hash.hexdigest()
print(f"MD5 checksum: {md5_checksum}")

# Determine file type
file_type = os.path.splitext(ZhWiki)[1][1:]
print(f"File type: {file_type}")
```

儲存成功!

```
709cd4605193
File type: bz2
```

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

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

```
'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`

```
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\layla\Desktop\三下\NLP\utils\dict.txt.big ...
Dumping model to file cache C:\Users\layla\AppData\Local\Temp\jieba.ued8779b98e591ed98804b0ce2c73f009.cache
Loading model cost 1.798 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
```

```
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)

```
#!pip install spacy
```

```
Collecting catalogue<2.1.0,>=2.0.6
  Downloading catalogue-2.0.8-py3-none-any.whl (17 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0
  Downloading spacy_loggers-1.0.4-py3-none-any.whl (11 kB)
Collecting langcodes<4.0.0,>=3.2.0
  Downloading langcodes-3.3.0-py3-none-any.whl (181 kB)
     ---------------------------------------- 181.6/181.6 kB 5.4 MB/s eta 0:00:00
Collecting murmurhash<1.1.0,>=0.28.0
  Downloading murmurhash-1.0.9-cp39-cp39-win_amd64.whl (19 kB)
Collecting cymem<2.1.0,>=2.0.2
  Downloading cymem-2.0.7-cp39-cp39-win_amd64.whl (30 kB)
Collecting preshed<3.1.0,>=3.0.2
  Downloading preshed-3.0.8-cp39-cp39-win_amd64.whl (97 kB)
     ---------------------------------------- 97.5/97.5 kB 5.4 MB/s eta 0:00:00
Collecting thinc<8.2.0,>=8.1.8
  Downloading thinc-8.1.10-cp39-cp39-win_amd64.whl (1.5 MB)
     ---------------------------------------- 1.5/1.5 MB 13.5 MB/s eta 0:00:00
Collecting srsly<3.0.0,>=2.4.3
  Downloading srsly-2.4.6-cp39-cp39-win_amd64.whl (482 kB)
     ---------------------------------------- 482.8/482.8 kB 6.1 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.15.0 in c:\programdata\anaconda3\lib\site-packages (from spacy) (1.23.3)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site-packages (from spacy) (21.3)
Collecting cymem<2.1.0,>=2.0.2
  Downloading cymem-2.0.7-cp39-cp39-win_amd64.whl (30 kB)
Collecting pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4
  Downloading pydantic-1.10.7-cp39-cp39-win_amd64.whl (2.2 MB)
     ---------------------------------------- 2.2/2.2 MB 13.8 MB/s eta 0:00:00
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in c:\users\layla\appdata\roaming\python\python39\site-packages (from spacy) (6.3.0)
Collecting spacy-legacy<3.1.0,>=3.0.11
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl (29 kB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\layla\appdata\roaming\python\python39\site-packages (from packaging>=20.0
Collecting typing-extensions>=4.2.0
  Downloading typing_extensions-4.5.0-py3-none-any.whl (27 kB)
Requirement already satisfied: idna<4,>=2.5 in c:\users\layla\appdata\roaming\python\python39\site-packages (from requests<3.0.0,>=2.13.0->sp
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\layla\appdata\roaming\python\python39\site-packages (from requests<3.0.0,
Requirement already satisfied: certifi>=2017.4.17 in c:\users\layla\appdata\roaming\python\python39\site-packages (from requests<3.0.0,>=2.13
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\layla\appdata\roaming\python\python39\site-packages (from requests<3.0.0,>=2
```

储存成功!

```
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script spacy.exe is installed in 'C:\Users\layla\AppData\Roaming\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

[notice] A new release of pip available: 22.3 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
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.stop_words)[:20]}  ...")
print("--")
print(f"英文停用詞  Total={len(nlp_en.Defaults.stop_words)}:  {list(nlp_en.Defaults.stop_words)[:20]}  ...")
```

```
--

中文停用詞 Total=1891: ['不怕', '哦', '怎么', '' ', '心里', '7', '除', '接著', '不管', '认真', '这般', '虽则', '∪φ∈', '乘胜', '怪', '甚至', '
--
英文停用詞 Total=326: ['hers', 'below', 'out', 'twelve', 'show', 'neither', 'must', 'latterly', '' m', 'and', 'around', 'any', 'here', 'although
```

```
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))
```

儲存成功!

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]

image.png

[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.

```
def  preprocess_and_tokenize(
        text:  str,  token_min_len:  int=1,  token_max_len:  int=15,  lower:  bool=True)  ->  List[str]:
        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
        ]
```

```
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，此畫為拉斐爾"))
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']
```

```
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，此畫為拉斐爾"))
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']
```

```
#!pip  install  nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting nltk
  Downloading nltk-3.8.1-py3-none-any.whl (1.5 MB)
     ---------------------------------------- 1.5/1.5 MB 5.7 MB/s eta 0:00:00
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (from nltk) (4.64.1)
Collecting regex>=2021.8.3
  Downloading regex-2023.5.5-cp39-cp39-win_amd64.whl (267 kB)
     ---------------------------------------- 268.0/268.0 kB ? eta 0:00:00
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\layla\appdata\roaming\python\python39\site-packages (from nltk) (1.2.0)
Requirement already satisfied: colorama in c:\users\layla\appdata\roaming\python\python39\site-packages (from click->nltk) (0.4.6)
Installing collected packages: regex, nltk
Successfully installed nltk-3.8.1 regex-2023.5.5
  WARNING: The script nltk.exe is installed in 'C:\Users\layla\AppData\Roaming\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

[notice] A new release of pip available: 22.3 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
%%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_len=1)
```

```
Parsing zhwiki-20230501-pages-articles.xml.bz2...
C:\Users\layla\AppData\Roaming\Python\Python39\site-packages\gensim\utils.py:1333: UserWarning: detected Windows; aliasing chunkize to chunkize_
  warnings.warn("detected %s; aliasing chunkize to chunkize_serial" % entity)
                                                                    t: 881.02 MiB
```

储存成功!

```
'''
%%time
%%memit

print(f"Parsing  {ZhWiki}...")
wiki_corpus  =  WikiCorpus(ZhWiki,  tokenizer_func=preprocess_and_tokenize,  token_min_len=1)
'''
```

```
UsageError: Cell magic `%%memit` not found.
```

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

```
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

```
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")
```

訓練 Word2Vec

```
from gensim.models import word2vec
import multiprocessing

WIKI_SEG_TXT = "wiki_seg.txt"
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_counts)

# 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)
```

儲存成功!

```
#! ls word2vec.zh*
```

```
'ls' 不是內部或外部命令、可執行的程式或批次檔。
```

```
import glob

files = glob.glob('word2vec.zh*')
for file in files:
        print(file)
```

```
word2vec.zh.300.model
word2vec.zh.300.model.syn1neg.npy
word2vec.zh.300.model.wv.vectors.npy
```

```
import os

files = [f for f in os.listdir('.') if f.startswith('word2vec.zh')]
for file in files:
        file_size = os.path.getsize(file)
        print(f"{file}: {file_size} bytes")
```

```
word2vec.zh.300.model: 13142217 bytes
word2vec.zh.300.model.syn1neg.npy: 493960928 bytes
word2vec.zh.300.model.wv.vectors.npy: 493960928 bytes
```

```
#!du -sh word2vec.zh*
```

```
71M     word2vec.zh.300.model
1.3G    word2vec.zh.300.model.trainables.syn1neg.npy
1.3G    word2vec.zh.300.model.wv.vectors.npy
```

查看模型以及詞向量實驗

模型其實就是巨大的 Embedding Matrix

```
print(model.wv.vectors.shape)
model.wv.vectors
```

```
(411634, 300)
array([[ 2.4865341e-01, -7.6906478e-01, -6.9945353e-01, ...,
         2.2321151e-01, -1.7378626e+00, -3.0584517e-01],
       [ 5.3441101e-01, -1.3043555e+00, -1.2721913e+00, ...,
         5.6111133e-01, -2.1026764e+00,  1.3059106e-01],
       [ 9.9232924e-01, -1.9290653e+00, -4.7821113e-01, ...,
         5.6262541e-01,  7.6808584e-01,  1.3814100e+00],
       ...,
       [-4.5188177e-02, -5.0329376e-04,  3.2227628e-02, ...,
        -7.8999475e-03,  1.0578581e-02,  3.2082967e-02],
       [-3.4778651e-02,  8.7892739e-03,  7.8873068e-02, ...,
        -7.7509448e-02,  1.0454625e-01,  2.5985707e-02],
       [ 2.5863567e-02,  8.3356071e-03,  7.4250191e-03, ...,
         3.6086902e-02, -6.2503647e-03,  1.7087938e-02]], dtype=float32)
```

收錄的詞彙

```
print(f"總共收錄了 {len(model.wv.vocab)} 個詞彙")

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

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [8], in <cell line: 1>()
----> 1 print(f"總共收錄了 {len(model.wv.vocab)} 個詞彙")
      3 print("印出 20 個收錄詞彙:")
      4 print(list(model.wv.vocab.keys())[:10])

File ~\AppData\Roaming\Python\Python39\site-
packages\gensim\models\keyedvectors.py:734, in KeyedVectors.vocab(self)
    732 @property
    733 def vocab(self):
--> 734     raise AttributeError(
    735         "The vocab attribute was removed from KeyedVector in Gensim
4.0.0.\n"
    736         "Use KeyedVector's .key_to_index dict, .index_to_key list, and
methods "
ey, attr) and .set_vecattr(key, attr, new_val)

thub.com/RaRe-Technologies/gensim/wiki/Migrating-
from-Gensim-3.x-to-4"
    739     )

AttributeError: The vocab attribute was removed from KeyedVector in Gensim 4.0.0.
Use KeyedVector's .key_to_index dict, .index_to_key list, and methods
```



```
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)[:20])
```

```
總共收錄了 411634 個詞彙
印出 20 個收錄詞彙:
['年', '月', '於', '「', '為', '日', '與', '後', '臺', '中', '對', '中國', '來', '軍', '10', '一個', '香港', '會', '馬', '12']
```

詞彙的向量

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

```
(300,)
array([-7.1025603e-02, -8.7070930e-01, -1.0018113e+00, -2.7349892e-01,
       -2.6700180e+00,  1.7203747e-01,  2.4088979e+00, -5.0068849e-01,
       -6.8385017e-01,  1.0539443e+00, -2.0111990e-01,  8.1220043e-01,
        9.0139151e-01,  2.0250010e-01,  4.4711018e-01,  3.2401684e-01,
       -5.1359761e-01,  1.2582105e+00,  6.6389138e-01, -7.6496220e-01,
        2.0346980e+00, -1.8037039e+00, -1.3502221e+00, -1.3266807e+00,
       -9.1472268e-02,  4.6907529e-01,  8.3529925e-01, -3.3579949e-02,
       -1.2841368e+00, -1.4216433e+00, -3.5353911e+00,  1.1974534e+00,
        8.2766318e-01,  2.3681536e+00,  1.1823212e+00,  2.2282634e+00,
       -1.1295012e+00, -7.7084042e-02, -8.8242024e-02, -2.0059049e+00,
        8.6848080e-01,  1.7723404e+00,  1.1191266e-01, -5.4771823e-01,
       -1.0560691e+00,  3.1923905e-01, -1.6110977e+00,  7.6948851e-01,
        1.7764758e+00, -4.1563356e-01,  8.8769341e-01, -2.5054393e-02,
       -9.8841256e-01, -1.3670897e+00,  8.8928884e-01,  1.4349517e-01,
       -5.0076336e-01,  6.7688175e-04, -6.9675851e-01, -5.6250870e-01,
       -2.2038779e+00,  2.3359618e+00, -4.8630396e-01, -5.8755124e-01,
        1.5162646e+00, -1.5606683e-01, -3.2269570e-01,  9.7901446e-01,
```

```
-6.7830347e-02, -1.6763237e+00,  1.4974518e-01,  8.8288420e-01,
-1.5042067e+00, -1.5393220e+00,  6.8613964e-01, -2.4993780e-01,
 3.7151146e+00,  1.9222531e+00, -3.2432282e-01,  8.4878367e-01,
-2.3847899e+00,  9.1686010e-01, -2.8682804e-01,  7.6469141e-01,
-5.2214909e-01, -1.2796396e+00, -4.7635797e-01, -4.4060424e-01,
-1.0426049e+00, -1.5698349e+00, -1.2845125e+00,  8.7414676e-01,
-3.2721850e-01,  8.0625141e-01,  1.3290088e+00, -3.3482575e-01,
-7.3224092e-01, -7.9838473e-01,  3.1034178e-01, -5.5610061e-01,
-2.3447311e+00,  1.5389574e+00, -3.7563962e-01, -6.7984116e-01,
-1.6031644e+00, -2.2542870e+00,  1.1136377e+00, -7.7640009e-01,
-6.9507706e-01,  5.5509162e-01,  1.9361519e+00, -2.0140285e+00,
 6.9867337e-01, -8.0800861e-01,  6.8793881e-01,  9.2369276e-01,
 4.5925575e-01, -3.4418264e-01, -6.2820923e-01,  9.4493580e-01,
-1.5932498e+00,  3.8376486e-01,  2.0887007e-01, -9.6492112e-01,
-1.7015384e+00,  5.7308990e-01, -1.0141536e+00,  1.7230620e+00,
 4.3569222e-01, -2.7334032e+00, -8.0261636e-01,  2.9696259e-01,
-1.4146224e+00,  8.1596053e-01, -7.7437423e-02, -2.7436908e-02,
 6.7634374e-02,  3.6793417e-01,  1.1319535e+00,  1.3444062e+00,
 3.3016562e-01,  8.1670624e-01, -1.6268456e+00, -1.0191444e+00,
-1.3740608e+00,  2.3476911e-01,  2.0238154e+00, -1.2681553e+00,
-1.1619594e+00,  1.7512807e-01,  4.7322330e-03,  1.4966218e+00,
 3.6402503e-01,  2.1933897e+00, -8.3032614e-01,  2.4669494e-01,
 1.6633941e+00, -6.5829444e-01,  1.4298936e+00, -2.0692706e+00,
-2.6872131e-01, -4.6227354e-01,  1.8587925e+00, -1.4903002e+00,
-2.7901175e+00,  2.3679776e+00, -1.2524782e+00, -1.6006221e-01,
 5.7917118e-01, -1.5736918e+00, -4.4254923e-01, -4.0143639e-01,
-5.5490983e-01, -1.1724381e+00,  8.1628673e-02,  5.3863090e-01,
 9.1064435e-01,  2.2464035e+00, -2.9734719e-01,  5.3590590e-01,
-7.3416740e-01,  1.3204632e+00, -7.0319116e-01, -4.7288388e-01,
-2.1737897e+00,  2.2321484e+00, -9.3299264e-01, -1.2974383e-01,
 6.9006777e-01,  9.0082443e-01, -1.4102949e+00,  4.1561884e-01,
-4.0199986e-01,  8.6203372e-01,  6.8842781e-01, -7.5543362e-01,
 1.4272561e+00,  2.0946093e+00, -8.6279249e-01, -8.6610848e-01,
-3.3131346e-02, -6.9607250e-02, -9.5628065e-01,  2.2601898e-01,
 1.1895827e+00,  2.2224230e-01, -8.3598095e-01, -4.8851666e-01,
-3.9265946e-02,  1.9369383e-01, -2.7198167e+00, -1.0466301e+00,
 1.6594436e+00, -7.2110665e-01, -1.5204869e-01,  8.0592388e-01,
-5.7732707e-01,  8.4545231e-01, -1.2421224e+00,  2.0234318e+00,
-2.3866025e-01, -2.0705917e+00, -1.3564705e+00,  3.4966695e-01,
```

沒見過的詞彙

```
word  =  "這肯定沒見過  "
```

儲存成功!

```
        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

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

```
[('飲品', 0.8662290573120117),
 ('果汁', 0.7706913352012634),
 ('含酒精', 0.7685386538505554),
 ('零食', 0.7678558826446533),
 ('罐裝', 0.7570993304252625),
 ('酒類', 0.7484425902366638),
 ('軟性', 0.7421537637710571),
 ('牛奶', 0.7306413650512695),
 ('巧克力', 0.7246869802474976),
 ('化妝品', 0.7241660952568054)]
```

```
model.wv.most_similar("car")
```

```
[('wagon', 0.8511000871658325),
 ('driver', 0.8491283655166626),
 ('vehicle', 0.8252284526824951),
 ('hybrid', 0.8243876695632935),
 ('smart', 0.8156096935272217),
 ('sport', 0.8129329681396484),
 ('custom', 0.8125067353248596),
 ('fast', 0.8081808090209961),
 ('motorcycle', 0.8053828477859497),
 ('motor', 0.8019233345985413)]
```

```
model.wv.most_similar("facebook")
```

```
[('專頁', 0.871256947517395),
 ('臉書', 0.8551862239837646),
 ('twitter', 0.8375945687294006),
 ('instagram', 0.8244563341140747),
 ('微博', 0.7601678967475891),
 ('貼文', 0.7546928524971008),
 ('推特', 0.7539433836936951),
 ('討論區', 0.7311115264892578),
 ('網誌', 0.724394679069519),
 ('留言', 0.7222513556480408)]
```

```
model.wv.most_similar("合約")
```

```
[('合同', 0.7685959935188293),
 ('續約', 0.7603106498718262),
 ('簽約', 0.7441477179527283),
 ('新合約', 0.7412732839584351),
 ('年合約', 0.689154326915741),
 ('到期', 0.6774435639381409),
 ('簽下', 0.6688246726989746),
 ('部頭', 0.6581522226333618),
 ('租約', 0.6483007073402405),
 ('球團', 0.6064363718032837)]
```

計算 Cosine 相似度

```
model.wv.similarity("連結", "陰天")
```

```
0.028523978
```

讀取模型

```
print(f"Loading {output_model}...")
new_model = word2vec.Word2Vec.load(output_model)
```



```
model.wv.similarity("連結", "陰天") == new_model.wv.similarity("連結", "陰天")
```

```
True
```