```
%load_ext memory_profiler
```

```
import os
import gensim
import jieba
import zhconv
from gensim.corpora import WikiCorpus
from datetime import datetime
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")
```

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

!du -sh $ZhWiki
!md5 $ZhWiki
!file $ZhWiki
```

```
2.6G    zhwiki-20230501-pages-articles-multistream.xml.bz2
MD5 (zhwiki-20230501-pages-articles-multistream.xml.bz2) = 27e78ff901bcd380
zhwiki-20230501-pages-articles-multistream.xml.bz2: bzip2 compressed data,
```

```
import spacy

nlp_zh = spacy.load("zh_core_web_sm")
nlp_en = spacy.load("en_core_web_sm")
```

```
STOPWORDS =  nlp_zh.Defaults.stop_words | nlp_en.Defaults.stop_words | set(["\n"

for word in STOPWORDS.copy():
    STOPWORDS.add(zhconv.convert(word, "zh-tw"))
```

```
def preprocess_and_tokenize(text, token_min_len = 1, token_max_len = 15, lower =
    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 STOPWOR
    ]
```

```
%%time
%%memit

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

    Parsing zhwiki-20230501-pages-articles-multistream.xml.bz2...
    /opt/homebrew/anaconda3/lib/python3.9/site-packages/gensim/utils.py:1333: U
      warnings.warn("detected %s; aliasing chunkize to chunkize_serial" % entit
    peak memory: 2493.45 MiB, increment: 1468.16 MiB
    CPU times: user 11min 18s, sys: 1min 39s, total: 12min 57s
    Wall time: 15min 24s
```

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

    ['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', '此畫為拉斐爾的作品'
    ['蘇格拉底之死', '由雅克', '路易', '大卫所繪', '年', '哲學', '是研究普遍的', '基本问
    ['文學', '在狭义上', '是一种语言艺术', '亦即使用语言文字为手段', '形象化地反映客观社会
```

```
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(datetime.now()):.19}] 已寫入 {texts_num} 篇斷詞文章")

    [2023-05-09 16:23:28] 已寫入 99999 篇斷詞文章
    [2023-05-09 16:25:26] 已寫入 199999 篇斷詞文章
    [2023-05-09 16:28:39] 已寫入 299999 篇斷詞文章
    [2023-05-09 16:31:01] 已寫入 399999 篇斷詞文章
```

```
%%time

from gensim.models import word2vec
import multiprocessing

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


sentences = word2vec.LineSentence(WIKI_SEG_TXT)

model = word2vec.Word2Vec(sentences, vector_size=word_dim_size, workers=max_cpu_

output_model = f"word2vec.zh.{word_dim_size}.model"
model.save(output_model)
```

```
    Use 8 workers to train Word2Vec (dim=300)
    CPU times: user 29min 41s, sys: 3min 36s, total: 33min 17s
    Wall time: 8min 48s
```

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

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

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

```
     57M    word2vec.zh.300.model
    1.8G    word2vec.zh.300.model.syn1neg.npy
    1.8G    word2vec.zh.300.model.wv.vectors.npy
```

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

```
    (1578559, 300)
    array([[-5.5526245e-01,  1.3259746e-01, -1.0660629e+00, ...,
             1.1430104e+00, -1.6921420e+00,  2.9938367e-01],
           [-1.0336232e+00,  5.3094292e-01, -7.8201252e-01, ...,
             7.0376503e-01, -1.8065145e+00, -8.5755032e-01],
           [-2.3009388e+00,  1.2452873e+00, -1.8620787e+00, ...,
             1.5956832e-01, -1.4018891e+00, -6.8957549e-01],
           ...,
           [-3.4782466e-02,  3.2090057e-02,  1.9380085e-02, ...,
            -3.1573884e-02,  1.8495960e-02, -1.2443333e-03],
           [-4.6306767e-02,  4.7765542e-02,  2.1769753e-02, ...,
            -3.8761143e-02, -8.3485330e-03, -9.4354460e-03],
           [ 8.8675199e-03, -5.9071090e-02, -5.8128562e-02, ...,
             7.1382105e-02,  1.4362816e-02,  7.8127548e-02]], dtype=float32)
```

```
vec = model.wv['數學家']
```

```
print(vec.shape)
vec

    (300,)
    array([-1.27786085e-01,  6.97323382e-01,  1.72649875e-01,  4.39359061e-02,
            6.64952397e-01, -3.16695243e-01, -7.22664118e-01,  7.34595299e-01,
           -5.01322329e-01, -2.41414875e-01, -3.42354596e-01,  2.28264451e-01,
            1.51635788e-03,  2.48132840e-01, -7.47404039e-01, -8.59439850e-01,
           -8.89822602e-01,  6.54842779e-02, -2.88151443e-01, -1.14040649e+00,
           -3.10163736e-01,  3.35338688e-03,  2.05355808e-02,  2.04405710e-01,
            4.36992705e-01,  2.02027410e-01,  2.61837304e-01, -1.12827063e+00,
           -3.33715290e-01,  5.83666205e-01, -8.37641120e-01, -5.62594891e-01,
           -1.14669139e-02, -7.51727641e-01, -2.15703368e-01, -1.11746892e-01,
            3.34301859e-01,  4.48547214e-01,  3.00215065e-01,  1.61787331e-01,
            7.55512357e-01,  2.81010479e-01, -8.65790844e-01, -4.39300954e-01,
           -1.91570699e-01,  2.03706086e-01,  2.93884248e-01, -3.59201074e-01,
           -1.50100815e+00,  3.65546852e-01,  1.45399943e-01, -1.19380899e-01,
            5.00896163e-02,  3.88028681e-01,  3.39963317e-01,  4.95500773e-01,
           -1.66295528e-01, -4.76028770e-01, -8.21013629e-01, -1.17870219e-01,
            4.05754298e-02, -8.38811159e-01, -3.60174030e-01,  8.14881772e-02,
            2.15626275e-03, -1.19358230e+00, -3.85677546e-01, -1.28972948e+00,
           -8.00500959e-02,  1.03168130e+00,  1.64019808e-01, -3.42538238e-01,
           -5.77008247e-01, -5.11489213e-01, -7.62048066e-01, -3.04360781e-03,
           -7.73207128e-01, -5.46071231e-01,  8.94546974e-03,  1.16762459e+00,
            1.52408674e-01, -9.09266770e-02,  1.49868476e+00, -1.04806483e+00,
           -5.42928219e-01,  8.86955500e-01, -4.16929394e-01,  4.10856068e-01,
            9.32247519e-01,  8.52952421e-01, -1.22937910e-01,  8.02406296e-02,
            6.60591960e-01, -4.38071638e-01, -3.64441365e-01, -1.39743432e-01,
            6.60219312e-01, -7.11655080e-01, -1.51682332e-01, -5.82821906e-01,
            5.04355915e-02,  2.41198525e-01, -4.28868920e-01, -8.02138388e-01,
            8.26735795e-01,  6.73015654e-01, -1.52661875e-01,  1.06569827e+00,
            8.55211541e-02,  1.01745749e+00, -2.20664382e-01, -2.77928077e-02,
            7.54249990e-01, -4.59738433e-01,  1.84031054e-02, -3.85671675e-01,
           -1.56205758e-01, -1.26461518e+00,  6.11914933e-01,  4.10615534e-01,
           -3.85434091e-01, -4.79394048e-01,  1.03487933e+00,  2.85662383e-01,
           -2.70555262e-02, -4.56405342e-01,  1.79889345e+00, -5.83157167e-02,
           -4.88117605e-01, -1.01885116e+00, -4.92202312e-01, -6.51369333e-01,
            1.31738842e-01,  4.76525217e-01,  6.85872078e-01,  6.27466083e-01,
            2.72292286e-01,  4.84887481e-01,  1.08355665e+00, -4.36814576e-01,
           -3.05541486e-01, -1.22043937e-01,  1.57572961e+00, -2.13530332e-01,
            1.24720953e-01,  7.68686086e-02,  4.50163335e-01,  1.11395061e+00,
           -3.39257121e-01, -2.49542519e-01,  7.19033897e-01,  1.04908086e-01,
            2.41330236e-01,  1.12151563e+00,  8.78720343e-01, -3.90396446e-01,
            2.82287180e-01,  3.79876256e-01, -1.11166932e-01, -9.83964279e-02,
            3.44172269e-01, -5.56732357e-01, -3.11707526e-01, -5.76732993e-01,
           -2.22552747e-01, -9.12441015e-02,  1.14051044e+00, -1.04689407e+00,
           -7.89170980e-01,  7.57203877e-01,  1.55394331e-01,  8.11617523e-02,
           -3.30299705e-01, -5.53432889e-02, -3.39242145e-02, -2.10399851e-01,
            3.86936404e-02, -8.93779770e-02, -1.07358670e+00,  1.61734009e+00,
           -5.00458777e-01, -1.13094173e-01,  3.43402147e-01,  3.59345853e-01,
            6.76875263e-02, -9.42175239e-02, -1.15093160e+00,  4.02587146e-01,
           -1.03070176e+00,  2.80828983e-01, -8.81094575e-01, -3.29105854e-02,
           -1.87014848e-01, -1.86245441e-01,  8.37286469e-03,  7.38601536e-02,
           -5.08308887e-01, -8.20093751e-02, -7.94892535e-02,  2.85848558e-01,
           -6.02228753e-02, -1.32618099e-01,  2.26877093e-01, -5.10094047e-01,
           -1.60541490e-01,  4.90036160e-01, -7.69753754e-02, -6.23379230e-01,
            3.32453623e-02,  3.36899370e-01,  8.07607293e-01,  1.65264064e-03,
           -6.49253249e-01, -3.02718091e-03,  9.42748249e-01, -1.27786338e-01,
           -4.49351162e-01,  5.35585344e-01,  5.82217634e-01,  8.28861892e-01,
           -1.04474947e-02, -5.98110497e-01, -5.94642162e-01, -6.30808175e-02,
           -1.60072446e-01,  1.27308056e-01, -4.34563830e-02, -6.25413477e-01,
           -9.98363048e-02, -1.02990699e+00,  5.97555518e-01,  1.69101608e+00,
```

```
word = "這肯定沒見過 "

try:
    vec = model.wv[word]
except KeyError as e:
    print(e)
```

```
    "Key '這肯定沒見過 ' not present"
```

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

```
    [('飲品', 0.9077415466308594),
     ('服飾', 0.8622954487800598),
     ('冰淇淋', 0.8549118638038635),
     ('零食', 0.8486245274543762),
     ('化妝品', 0.8456978797912598),
     ('啤酒', 0.8391772508621216),
     ('食品', 0.8333563804626465),
     ('手錶', 0.8316587805747986),
     ('家電', 0.8273059725761414),
     ('咖啡', 0.826672375202179)]
```

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

```
    [('truck', 0.7631158828735352),
     ('wagon', 0.7157680988311768),
     ('seat', 0.7157661318778992),
     ('porsche', 0.7081111669540405),
     ('motor', 0.706754207611084),
     ('volkswagen', 0.7012067437171936),
     ('cadillac', 0.7010031342506409),
     ('motorcycle', 0.6970956921577454),
     ('saloon', 0.6943933963775635),
     ('coupe', 0.693434476852417)]
```

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

    [('instagram', 0.8760004639625549),
     ('臉書', 0.8241063952445984),
     ('twitter', 0.795880138874054),
     ('專頁', 0.786584198474884),
     ('facebook專頁', 0.7610263228416443),
     ('推特', 0.7491406798362732),
     ('yahoo', 0.7351948618888855),
     ('myspace', 0.7329583764076233),
     ('blogger', 0.732337474822998),
     ('新浪微博', 0.7301666736602783)]


model.wv.most_similar("詐欺")

    [('恐嚇', 0.8735519647598267),
     ('性騷擾', 0.8674851059913635),
     ('盜竊', 0.8672724366188049),
     ('竊盜', 0.854965329170227),
     ('騷擾', 0.8486077189445496),
     ('解決問題', 0.8465505242347717),
     ('不負責任', 0.8458543419837952),
     ('賣淫', 0.8389670252799988),
     ('的態度', 0.8382847905158997),
     ('羞辱', 0.8377043008804321)]


model.wv.most_similar("合約")

    [('總值', 0.8325327634811401),
     ('耗資超過', 0.7923861742019653),
     ('年內', 0.786107063293457),
     ('由於covid', 0.7669464945793152),
     ('百萬歐元', 0.7648088932037354),
     ('另加', 0.7599743604660034),
     ('計劃在', 0.7575161457061768),
     ('億美金', 0.7558749318122864),
     ('預算為', 0.7556796073913574),
     ('花費', 0.7482377886772156)]


model.wv.similarity("連結", "鏈結")

    0.45945182


model.wv.similarity("連結", "陰天")

    0.30392832
```

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

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

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

```
    True
```