

WORD2VEC EXERCISE

張家瑋 博士

國立臺中科技大學資訊工程系助理教授

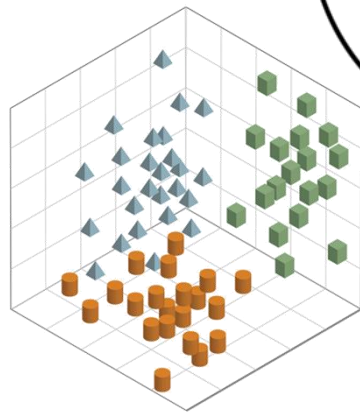
文字檔案

Input:
one document



Lorem ipsum dolor
sit amet, consete-
tur sadipscing elitr,
sed diam nonumy
alrimod tempor
invidunt ut labore
et dolore magna
aliquyam erat, sed
diam voluptua. At
vero eos et

word
vectors



word2vec

將被拆解成多個字元

Model:



vector space

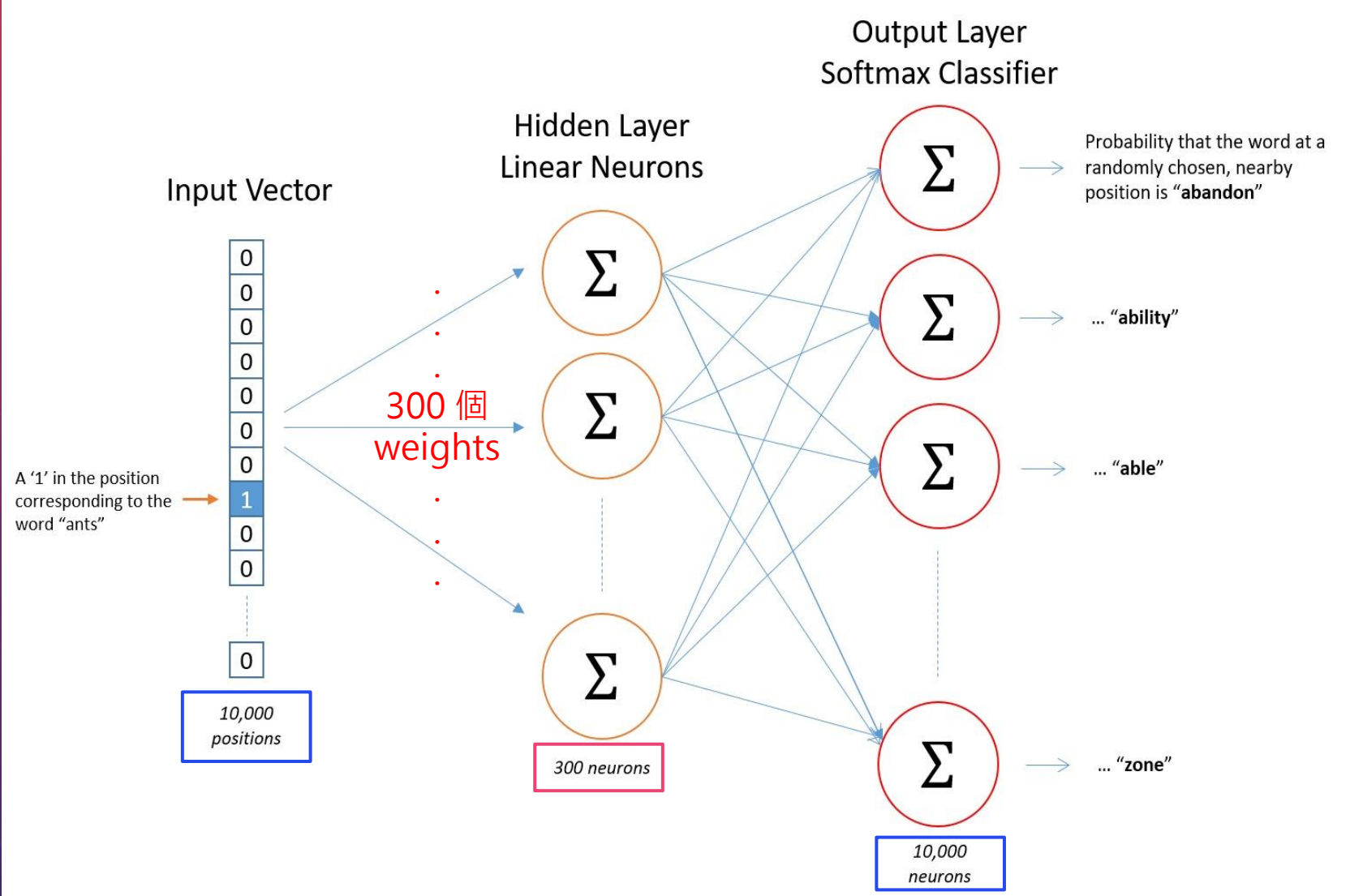
解析成多元維度的向量

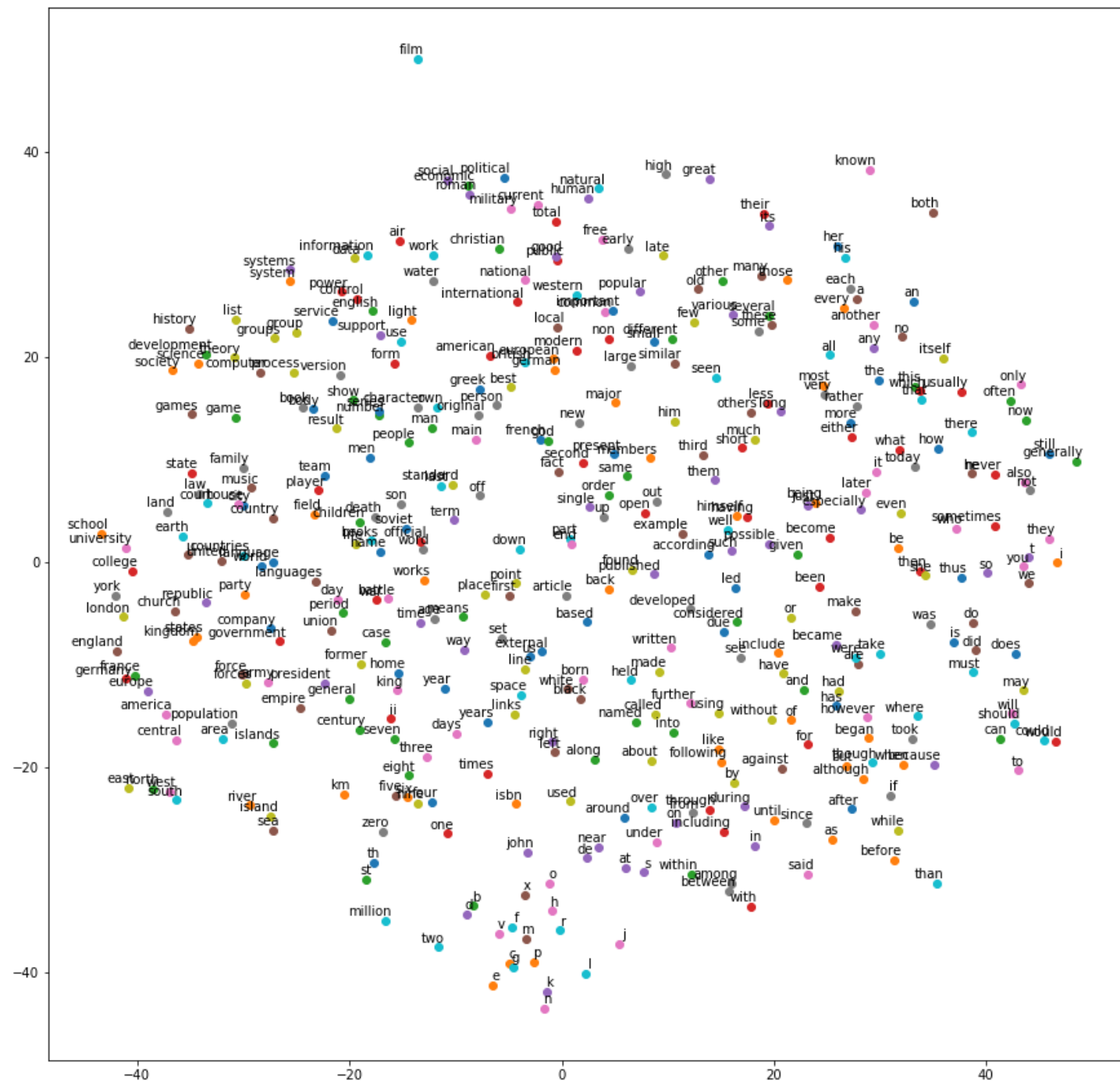
透過向量比對
找出相似的資料

most_similar('france'):

spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130

highest cosine
distance values
in vector space
of the nearest
words





INSTALL PACKAGES

- pip install genism
- pip install jieba
- pip install hanziconv

Word2Vec: <https://github.com/Alex-CHUN-YU/Word2vec>

步驟

1. 將 wiki 的 xml 轉換成 txt
(wiki_xml2txt.ipynb)
2. 將簡體轉繁體，並且斷詞去除廢字
(segmentation.ipynb)
3. 訓練Word2Vec模型
(train.ipynb)
4. 使用Word2Vec模型
(WordVec_LoadPretrainModel.ipynb)

Word2Vec: <https://github.com/Alex-CHUN-YU/Word2vec>

結果

1. 輸入一個詞彙會找出前5名相似
2. 輸入兩個詞彙會算出兩者之間相似度
3. 輸入三個詞彙爸爸之於老公,如媽媽之於老婆

輸入格式(Ex: 爸爸,媽媽,....註:最多三個詞彙)

老師

詞彙相似詞前 5 排序

班導,0.6360481977462769

班導師,0.6360464096069336

代課,0.6358826160430908

級任,0.6271134614944458

班主任,0.6270170211791992

輸入格式(Ex: 爸爸,媽媽,....註:最多三個詞彙)

爸爸,媽媽

計算兩個詞彙間 Cosine 相似度

0.780765200371

輸入格式(Ex: 爸爸,媽媽,....註:最多三個詞彙)

爸爸,老公,媽媽

爸爸之於老公,如媽媽之於

老婆,0.5401346683502197

蠢萌,0.5245970487594604

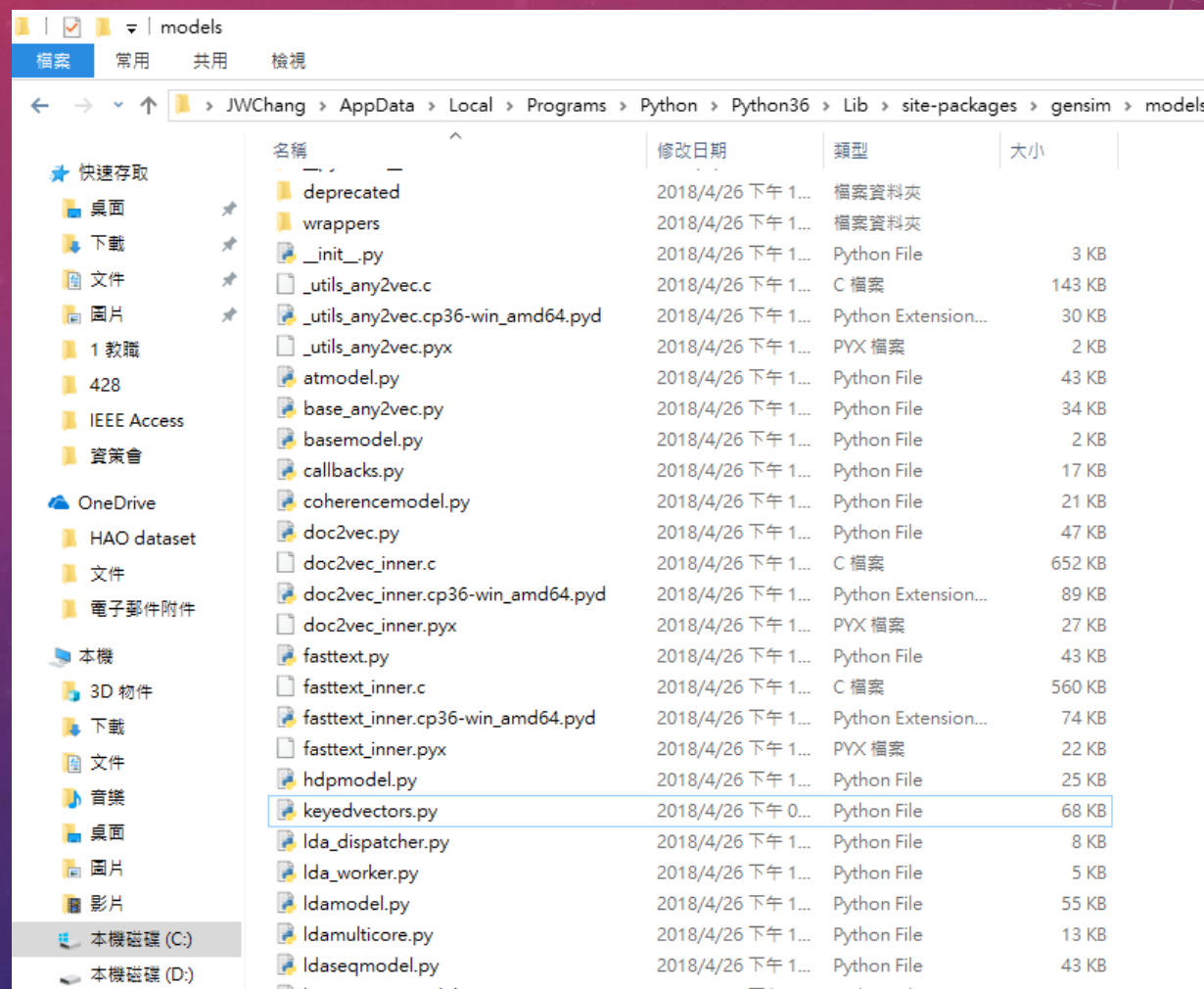
夠秤,0.5059393048286438

駁命,0.4888317286968231

孔爵,0.4857243597507477

修改 GENSIM

C:\Users\user\AppData\Local\Programs\Python\Python36\Lib\site-packages\gensim\models\keyedvectors.py



修改 GENSIM

```
193 class WordEmbeddingsKeyedVectors(BaseKeyedVectors):
194     """Class containing common methods for operations over word vectors."""
195
196     def __init__(self, vector_size):
197         super(WordEmbeddingsKeyedVectors, self).__init__(vector_size=vector_size)
198         self.vectors_norm = None
199         self.index2word = []
200         self.zeroVec = []
201         for i in range(300):
202             self.zeroVec.append(0.0)
```

修改 GENSIM

```
255 def word_vec(self, word, use_norm=False):
256     """
257     Accept a single word as input.
258     Returns the word's representations in vector space, as a 1D numpy array.
259
260     If `use_norm` is True, returns the normalized word vector.
261
262     Examples
263     -----
264     >>> trained_model['office']
265     array([ -1.40128313e-02, ...])
266
267     """
268     if word in self.vocab:
269         if use_norm:
270             result = self.vectors_norm[self.vocab[word].index]
271         else:
272             result = self.vectors[self.vocab[word].index]
273
274         result.setflags(write=False)
275         return result
276     else:
277         return array(self.zeroVec)
278     #raise KeyError("word '%s' not in vocabulary" % word)
```

The background is a vertical gradient from red at the top to blue at the bottom, with a subtle pattern of white dots. On the left side, there are several white circular and semi-circular lines, some with arrows indicating a clockwise direction. A prominent circular scale with degree markings from 140 to 260 is visible, with the numbers increasing in a clockwise direction. The text "THANK YOU" is positioned on the right side of the image.

THANK YOU