WORD2VEC EXERCISE

張家瑋 博士
新漢股份有限公司創新工業4.0中心顧問

文字檔案

Input:
one document

word2vec

將被拆解成多個字元

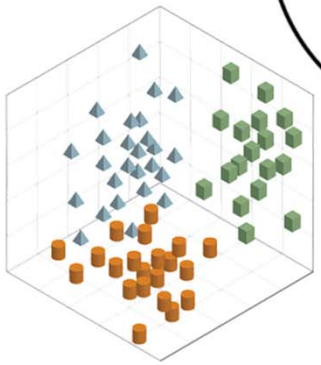
Model:

透過向量比對
找出相似的資料

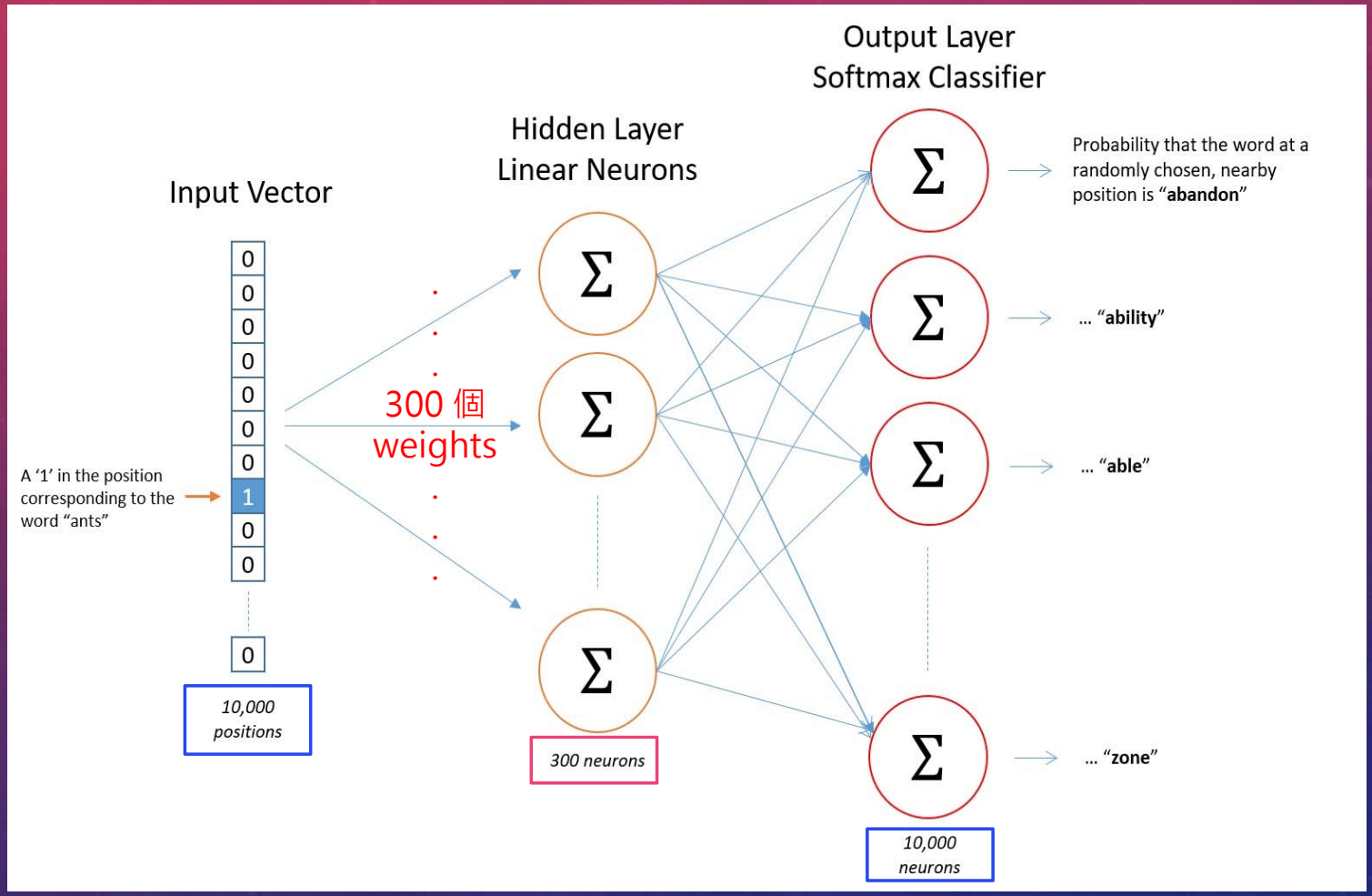
most_similar('france'):

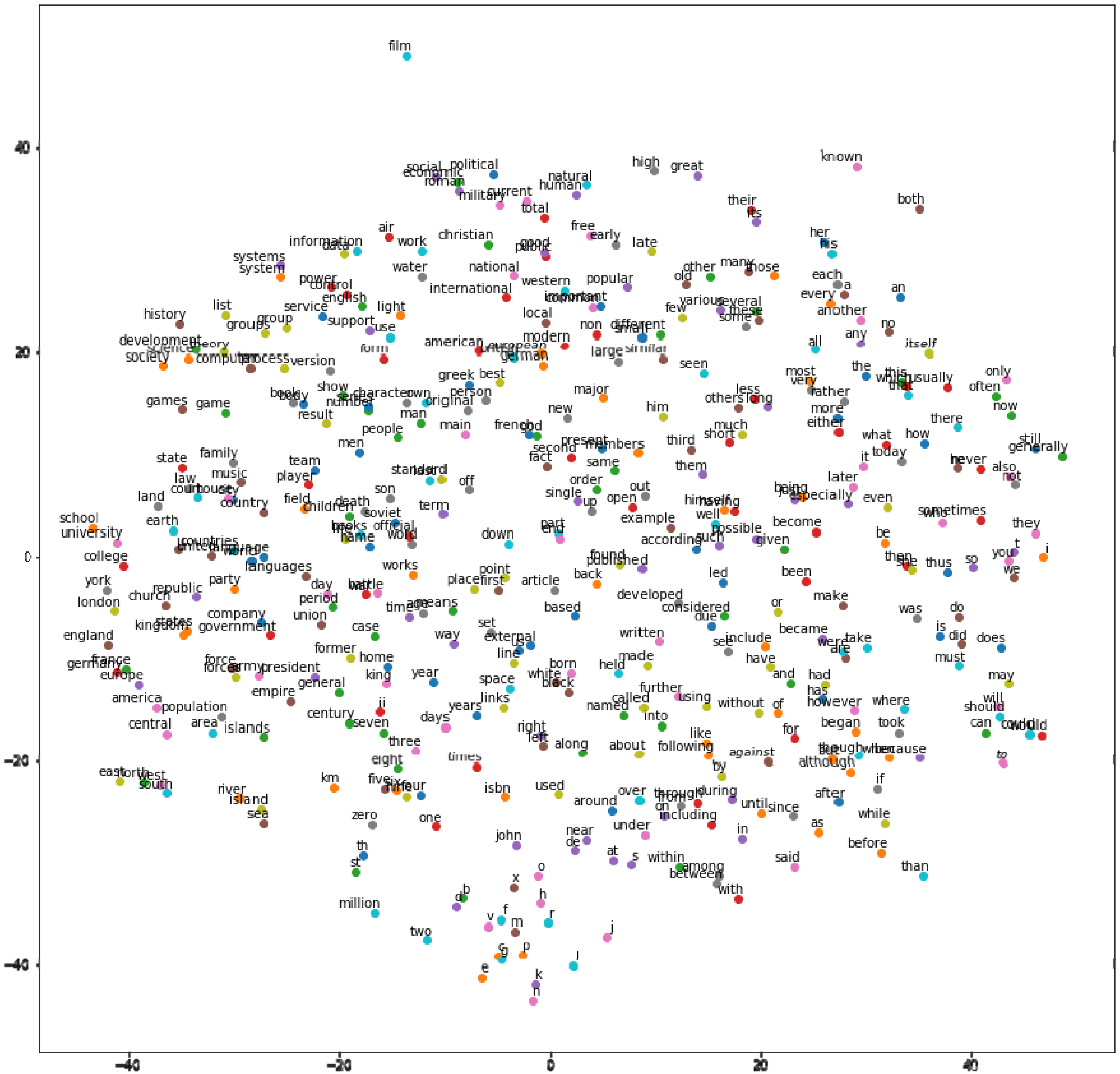
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130

highest cosine distance values in vector space of the nearest words



解析成多元維度的向量


Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
300 個 weights
A '1' in the position corresponding to the word "ants"
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
300 neurons
10,000 neurons


40
20
0
−20
−40
−40
−20
0
20
40
film
social
political
economic
roman
military
current
natural
human
total
high
great
known
their
its
both
air
information
data
work
christian
free
early
late
systems
system
water
national
power
control
english
international
western
popular
other
many
those
each
an
history
list
group
groups
service
support
light
local
few
various
several
these
some
every
another
no
use
american
modern
non
different
small
all
any
itself
development
science
society
computer
process
version
form
european
german
large
similar
seen
most
very
the
this
which
that
usually
only
often
games
game
greek
best
major
less
others
long
rather
more
now
book
body
show
number
character
own
person
original
man
result
new
him
either
there
what
how
people
main
french
god
third
much
short
it
today
still
generally
men
state
family
team
player
standard
off
present
second
members
fact
same
them
never
also
not
law
music
son
order
out
later
land
country
field
children
death
soviet
term
single
up
open
himself
having
especially
even
sometimes
who
school
university
earth
example
well
possible
become
they
down
part
end
according
such
given
be
countries
language
world
languages
works
found
published
back
led
been
then
thus
so
you
we
college
point
place
first
article
developed
make
i
york
republic
party
day
battle
war
period
london
church
company
union
case
means
based
considered
due
or
was
do
kingdom
government
states
way
set
external
written
include
became
take
is
did
does
england
former
home
line
made
born
held
see
have
were
are
must
force
forces
army
president
france
germany
europe
king
year
white
black
space
and
had
has
may
empire
general
ii
further
using
will
should
america
population
century
years
called
named
links
without
of
however
where
central
area
islands
seven
days
right
left
into
began
took
can
could
would
three
about
along
following
like
for
because
against
though
although
to
eight
times
by
if
east
north
west
south
river
island
sea
km
five
six
four
isbn
used
over
through
during
after
around
from
on
until
since
while
zero
one
including
under
in
as
john
near
de
before
th
st
at
s
within
among
between
with
said
than
o
x
b
d
h
million
f
v
m
r
j
two
c
g
p
e
l
k
n

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
```

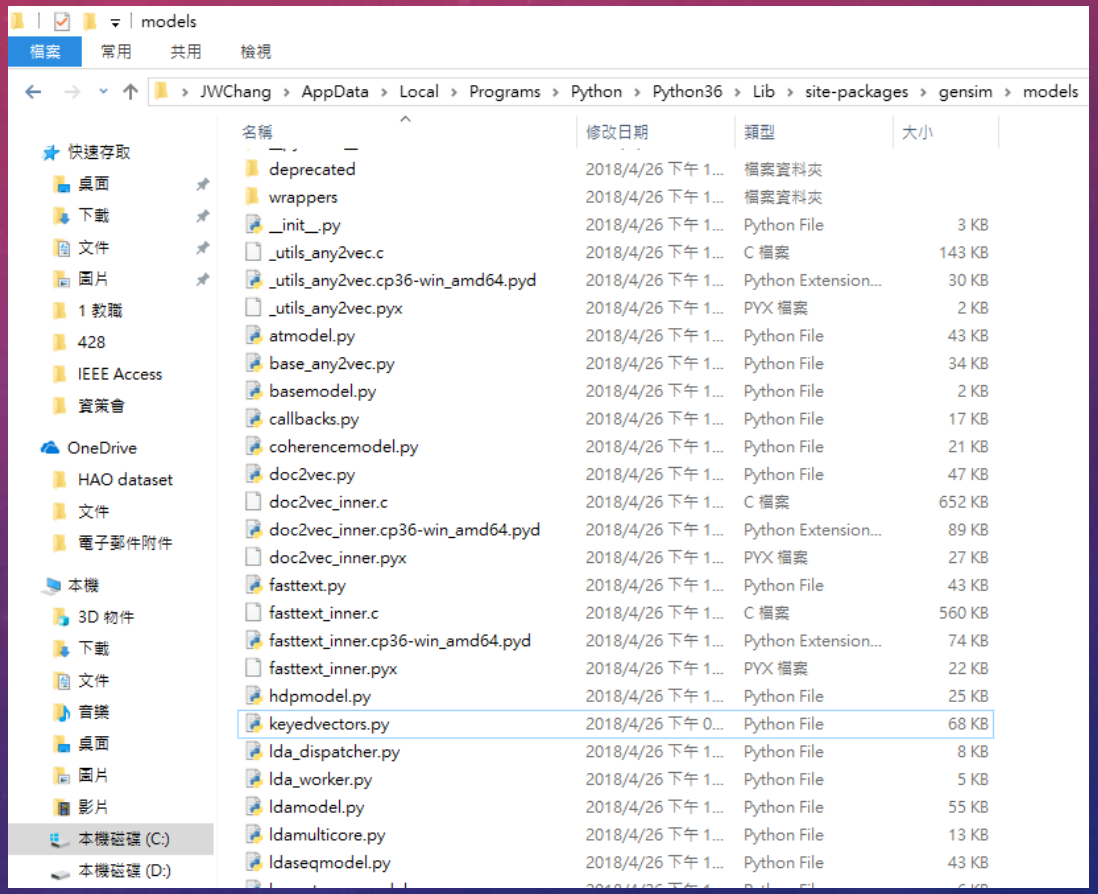
```
輸入格式( Ex: 爸爸,媽媽,....註:最多三個詞彙)
老師
詞彙相似詞前 5 排序
班導,0.6360481977462769
班導師,0.6360464096069336
代課,0.6358826160430908
級任,0.6271134614944458
班主任,0.6270170211791992
```

```
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)
```

THANK YOU