Project Name: Word Embeddings Demo
Module Name: Backend (backend.py)
Check: Functions that collectively implemented the back end of this project

Precondition: Pre-trained GloVe Embedding (in word2vec format), which can be obtained by running the *getGloVe.sh* script.

Test Case ID	Test Scenario	Test Steps *(NOTE: The top dir of the WordED project is denoted as WEDTop)*	Test Data	Expected Results	Actual Results	Pass/ Fail
TB1a	Test function (*similar_by_word_r*) with a normal word	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a normal word to the function and press enter *word2vecs.similar_by_word_r('interesting', topn=5)*	'interesting'	Return a list of similar words with their similarities to the input word	As expected (See Fig. 1)	Pass
TB1b	Test function (*similar_by_word_r*) with an unknown word	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input an unknown word to the function and press enter *word2vecs.similar_by_word_r('rtosssssstasa', topn=10)*	'rtosssssstasa'	Report the word is an unknown word and return the pre-defined value	As expected (See Fig. 2)	Pass

TB2a	Test function (*similar_by_word_rs*) with normal words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input two normal words to the function and press enter *word2vecs.similar_by_word_rs('interesting', 'lunch', topn=5)*	‘interesting’, 'lunch'	Return a list of similar words with their similarities to the input words	As expected (See Fig. 3)	Pass
TB2b	Test function (*similar_by_word_rs*) with unknown words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input two unknown words to the function and press enter *word2vecs.similar_by_word_rs('balabalabalala', 'supermanhillaba', topn=5)*	'balabalabalala', 'supermanhillaba'	Report words are unknown and return the pre-defined values	As expected (See Fig. 4)	Pass
TB3	Test function (*similar_by_vector_r*)	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a vector that has the same embedding dimension with the Glove	The vector of ‘interesting’	Return the *topn* similar words and their similarities to the vector	As expected (See Fig.5)	Pass

		embeddings to the function and press enter *word2vecs.similar_by_vector_r(word2vecs['interesting'], topn=5)*				
TB4a	Test function (*find_between*) with normal words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input two normal words into the function and press enter *word2vecs.find_between('school', 'interesting', step=5)*	'school', 'interesting'	Report the top-2 to each split point and return the n-split words and their similarities to the n-split vector	As expected (See Fig.6)	Pass
TB4b	Test function (*find_between*) with unknown words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input two unknown words into the function and press enter *word2vecs.find_between('asdasschool', 'interesting', step=5)*	'asdasschool', 'interesting'	Report the unknown words and return the predefined values	As expected (See Fig. 7)	Pass

TB5a	Test function (*format_vocab_embedding*) with normal words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a list of normal words into the function and press enter *word2vecs.format_vocab_embedding(['today', 'is', 'sunny'])*	[‘today’, ‘is’, ‘sunny’]	Return the vocabs looked up and their embeddings	As expected (See Fig. 8)	Pass
TB5b	Test function (*format_vocab_embedding*) with unknown words	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a list of unknown words list into the function and press enter (*word2vecs.format_vocab_embedding(['TTToday', 'is', 'sunny'])*	[‘TTToday’, ‘is’, ‘sunny’]	Report unknown words	As expected (See Fig. 9)	Pass
TB6	Test function (*plot_figure*)	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a list of words and their embeddings and press enter	[‘today’, ‘is’, ‘sunny’] and the embeddings of these words	Plot a 2D figure for the distribution of these words	As expected (See Fig. 10, 11)	Pass

TB7	Test function (*plot_3D_figure*)	In Terminal 1. Go to dir WEDTop $ cd WEDTop 2. Run the below command $ python3 backend.py 3. Input a list of words and their embeddings and press enter	['today', 'is', 'sunny'] and the embeddings of these words	Plot a 3D figure for the distribution of these words	As expected (See Fig. 11, 12)	Pass

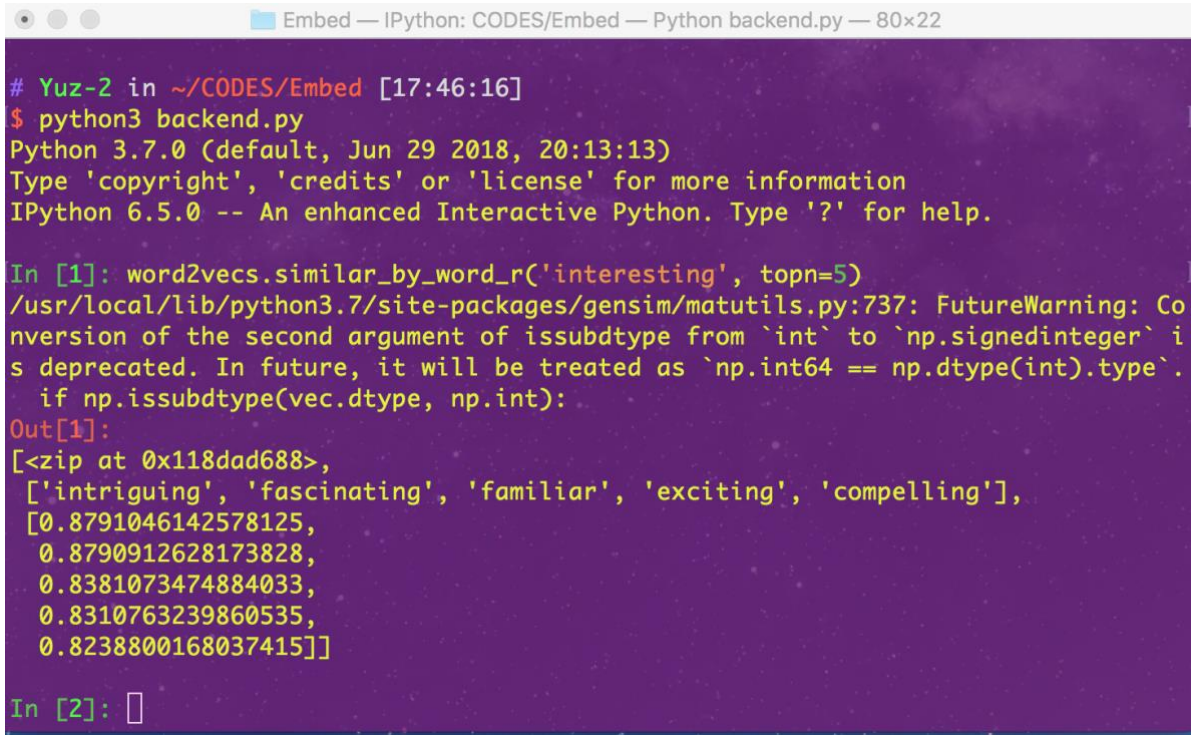


Fig. 1 Result for TB1a

```
# Yuz-2 in ~/CODES/Embed [17:49:42]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.similar_by_word_r('rtosssssstasa', topn=5)
Unknown word rtosssssstasa
Out[1]: [<zip at 0x115823b48>, ['Unknown'], [1]]
```

Fig. 2 Result for TB1b

```
Embed — IPython: CODES/Embed — Python backend.py — 94×35
# Yuz-2 in ~/CODES/Embed [17:54:08]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.similar_by_word_rs('interesting', 'lunch', topn=5)
/usr/local/lib/python3.7/site-packages/gensim/matutils.py:737: FutureWarning: Conversion of th
e second argument of issubdtype from `int` to `np.signedinteger` is deprecated. In future, it
will be treated as `np.int64 == np.dtype(int).type`.
  if np.issubdtype(vec.dtype, np.int):
Out[1]:
(<zip at 0x12cbae248>,
 ['intriguing',
  'fascinating',
  'familiar',
  'exciting',
  'compelling',
  'breakfast',
  'dinner',
  'meals',
  'buffet',
  'meal'],
 [0.8791046142578125,
  0.8790912628173828,
  0.8381073474884033,
  0.8310763239860535,
  0.8238800168037415,
  0.8713045716285706,
  0.8368837237358093,
  0.7872195243835449,
  0.7797691226005554,
  0.7667425870895386])
```

Fig. 3 Result for TB2a

```
# Yuz-2 in ~/CODES/Embed [18:02:06]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.similar_by_word_rs('balabalabalala', 'supermanhillaba', topn=5)
Unknown word
Out[1]: [<zip at 0x120270b48>, ['Unknown'], [1]]
```

Fig. 4 Result for TB2b

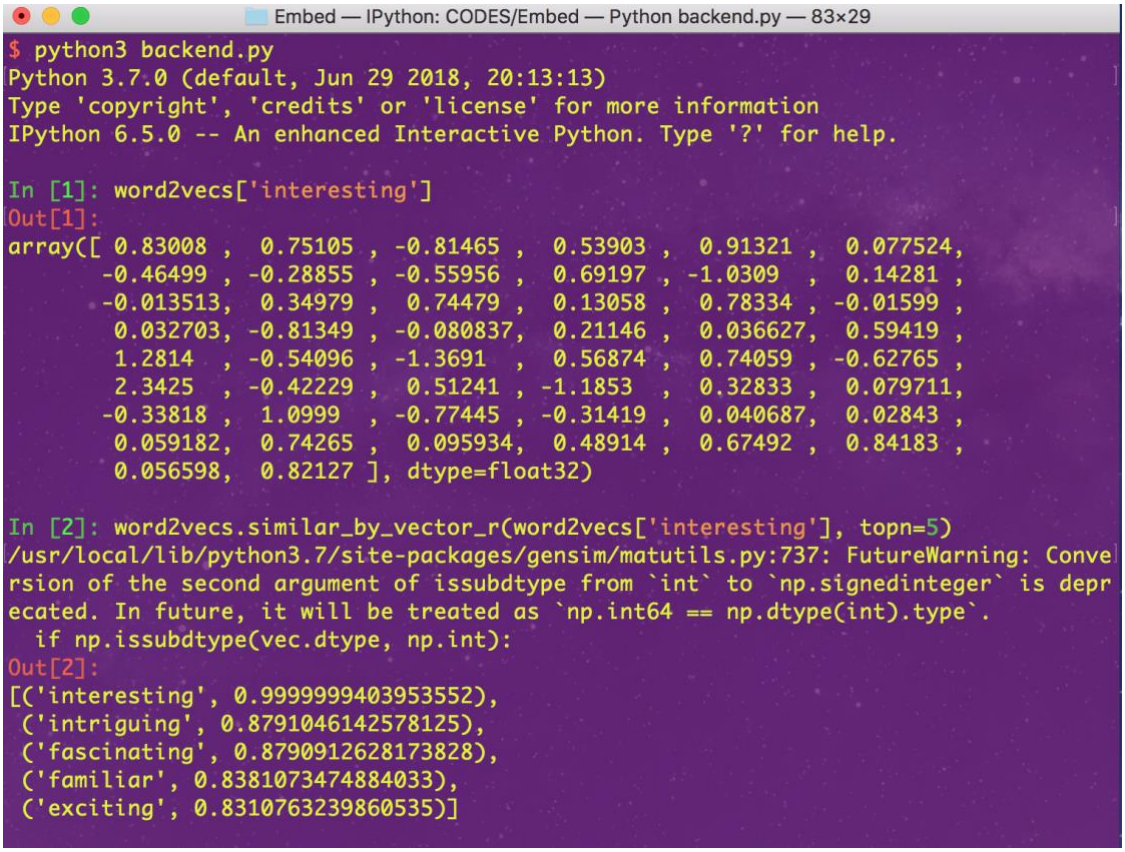


Fig. 5 Result for TB3

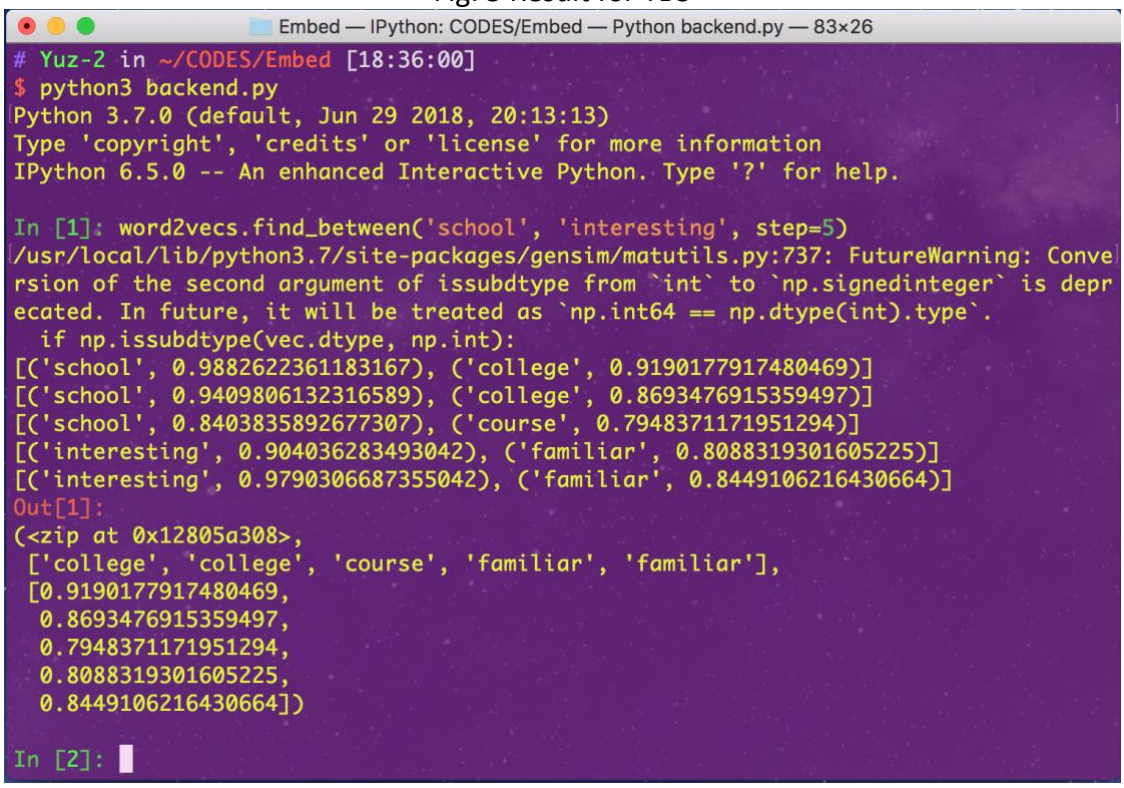


Fig. 6 Result for TB4a

```
# Yuz-2 in ~/CODES/Embed [20:43:32]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.find_between('asdasschool', 'interesting', step=5)
Unknown Words
Out[1]: (<zip at 0x11c1f7c88>, ['asdasschool', 'interesting'], [1])
```

Fig. 7 Result for TB4b

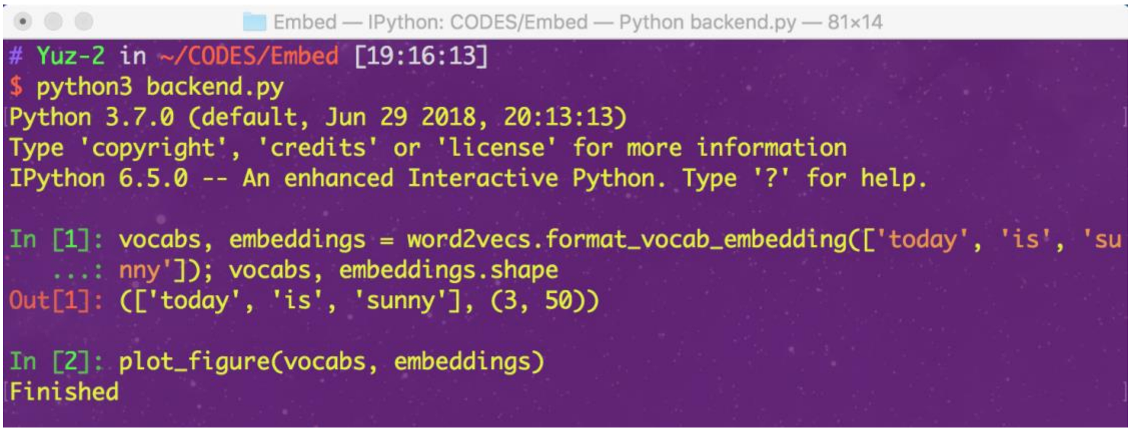
Embed — IPython: CODES/Embed — Python backend.py — 91×41

```
# Yuz-2 in ~/CODES/Embed [19:01:09]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.format_vocab_embedding(['today', 'is', 'sunny'])
Out[1]:
(['today', 'is', 'sunny'],
 array([[ 2.7751e-04,  4.2673e-01, -8.2938e-02,  2.7601e-01,  6.4721e-01,
         -9.1728e-01, -6.3471e-01, -2.8023e-01, -6.6653e-01, -2.8436e-01,
         -6.4249e-02, -4.3626e-01, -1.0830e-01, -3.5818e-01,  7.2311e-01,
          6.5368e-01, -2.9573e-01,  1.2007e-01, -2.9959e-02, -2.0594e-01,
          2.0017e-01,  1.6421e-01,  1.5202e-01, -2.4855e-02,  5.2887e-01,
         -1.3625e+00, -5.6036e-01,  1.7777e-01, -9.1003e-02,  9.7549e-02,
          3.5102e+00,  1.0631e-01,  6.5602e-02, -8.0777e-02, -1.2553e-01,
         -6.9932e-01, -1.5068e-02,  3.9353e-01, -2.8195e-03,  2.0635e-01,
         -4.7726e-01, -1.2639e-01,  2.9399e-01,  1.0000e-01,  3.4015e-04,
          6.2769e-01, -4.5344e-01,  3.9615e-01,  1.8857e-02,  1.7536e-01],
        [ 6.1850e-01,  6.4254e-01, -4.6552e-01,  3.7570e-01,  7.4838e-01,
          5.3739e-01,  2.2239e-03, -6.0577e-01,  2.6408e-01,  1.1703e-01,
          4.3722e-01,  2.0092e-01, -5.7859e-02, -3.4589e-01,  2.1664e-01,
          5.8573e-01,  5.3919e-01,  6.9490e-01, -1.5618e-01,  5.5830e-02,
         -6.0515e-01, -2.8997e-01, -2.5594e-02,  5.5593e-01,  2.5356e-01,
         -1.9612e+00, -5.1381e-01,  6.9096e-01,  6.6246e-02, -5.4224e-02,
          3.7871e+00, -7.7403e-01, -1.2689e-01, -5.1465e-01,  6.6705e-02,
         -3.2933e-01,  1.3483e-01,  1.9049e-01,  1.3812e-01, -2.1503e-01,
         -1.6573e-02,  3.1200e-01, -3.3189e-01, -2.6001e-02, -3.8203e-01,
          1.9403e-01, -1.2466e-01, -2.7557e-01,  3.0899e-01,  4.8497e-01],
        [ 2.8303e-01,  8.2852e-01, -7.9144e-01,  9.6621e-02, -2.1481e-01,
         -1.3104e+00, -8.4040e-01,  3.9867e-01, -3.2569e-01, -4.3573e-03,
         -2.8412e-01, -6.5879e-01,  9.3098e-01, -7.6139e-02,  5.5169e-03,
          1.0263e+00, -5.9263e-02, -8.6535e-02, -4.2904e-01, -6.5045e-01,
         -6.0974e-01,  1.3374e+00,  5.4669e-01,  8.9622e-02,  8.3857e-01,
          7.3393e-01, -5.9149e-02,  1.6792e+00,  6.5027e-01,  8.6267e-01,
          1.6334e+00,  4.7072e-02,  3.7430e-01, -6.7205e-03,  5.5574e-01,
         -3.4922e-01, -2.1940e-01,  4.0909e-01, -2.1448e-01, -6.2203e-01,
         -4.8862e-01,  5.2183e-01, -2.9933e-01, -7.0177e-01, -2.6734e-02,
          2.0646e-01,  1.4541e-01, -1.2577e+00,  1.4762e-01,  6.7417e-01]],
       dtype=float32))
```

Fig. 8 Result for TB5a

```
Embed — IPython: CODES/Embed — Python backend.py — 75×25
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: word2vecs.format_vocab_embedding(['TTToday', 'is', 'sunny'])
Unknown word TTToday
Out[1]:
(['TTToday', 'is', 'sunny'],
 array([[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  0.00000000e+00],
        [ 6.18499994e-01,  6.42539978e-01, -4.65519994e-01,
```

Fig. 9 Result for TB5b



```
Embed — IPython: CODES/Embed — Python backend.py — 81×14
# Yuz-2 in ~/CODES/Embed [19:16:13]
$ python3 backend.py
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: vocabs, embeddings = word2vecs.format_vocab_embedding(['today', 'is', 'su
   ...: nny']); vocabs, embeddings.shape
Out[1]: (['today', 'is', 'sunny'], (3, 50))

In [2]: plot_figure(vocabs, embeddings)
Finished
```

Fig. 10 Result for TB6

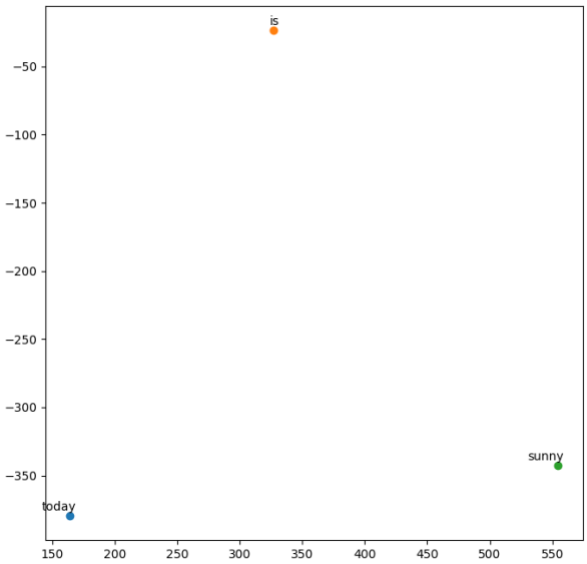


Fig. 11 Result for TB6

Fig. 11 Result for TB7

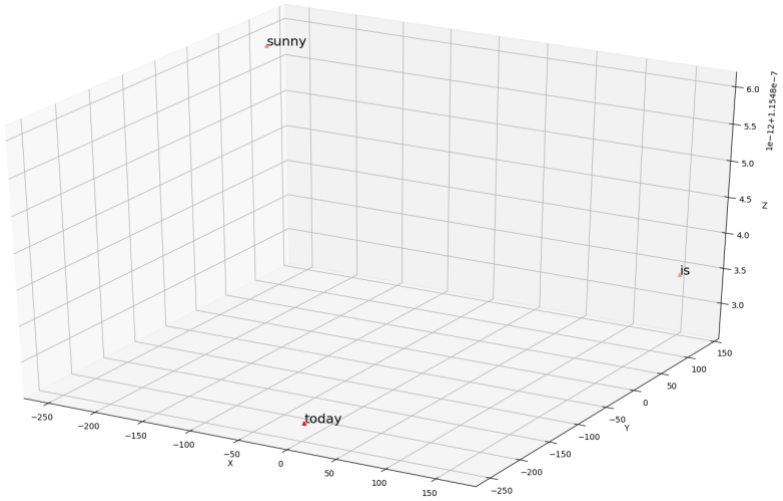


Fig. 12 Result for TB7