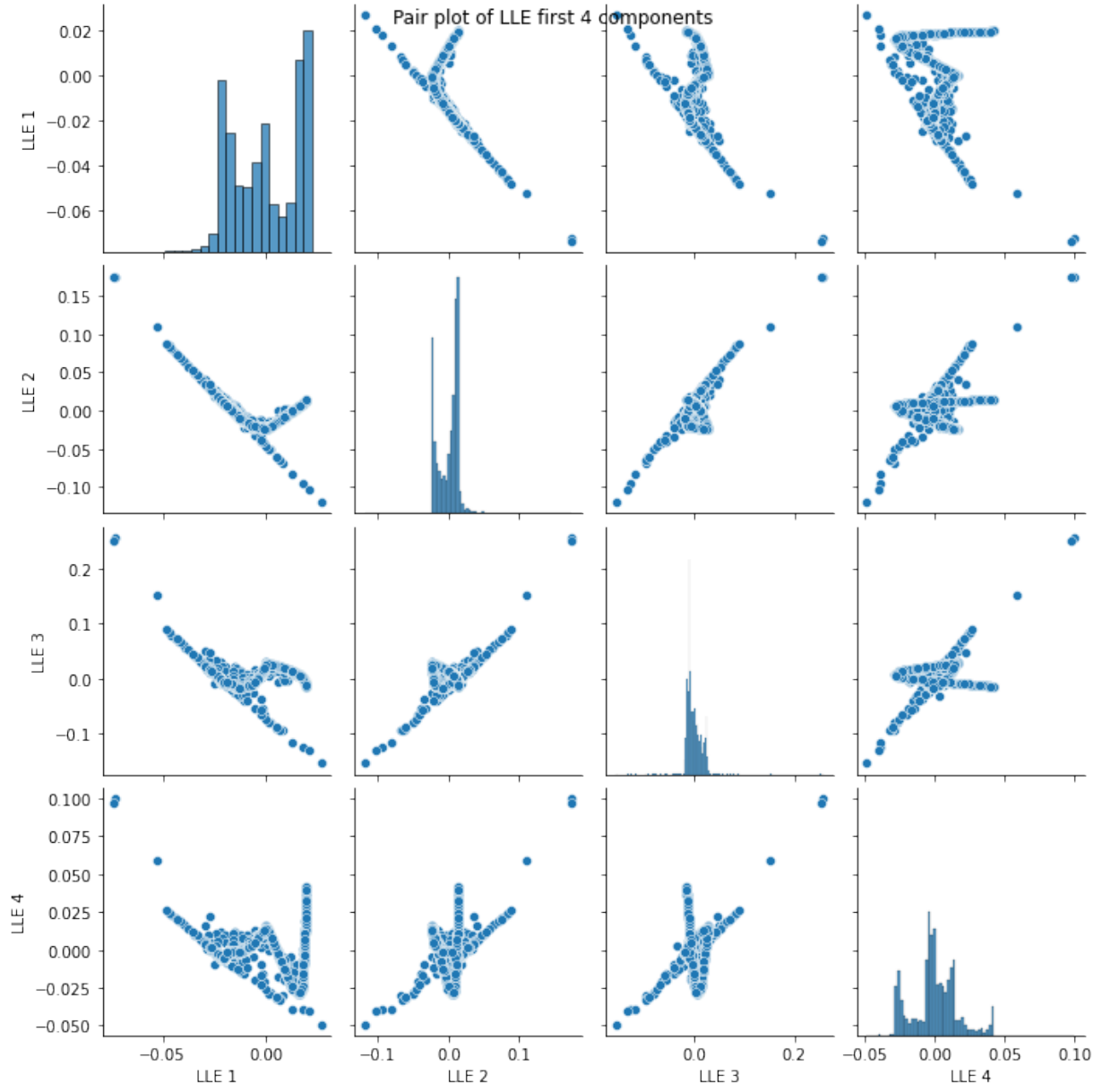
CM3: Part2 : LLE

[ ]:
```
from sklearn.manifold import LocallyLinearEmbedding

# apply LLE on word2vec test embeddings with embedding dimentionality 4
lle = LocallyLinearEmbedding(n_components=4, random_state=0)
lle_test = lle.fit_transform(list(test_embedding.values()))
```

LLE Pair-plot

[ ]:
```
# visualize the first four dimensionalities of this subspace using a pairs plot
LLE = ['LLE 1','LLE 2','LLE 3','LLE 4']
a = sns.pairplot(pd.DataFrame(lle_test,
                 columns = LLE))
a.fig.suptitle("Pair plot of LLE first 4 components")
```

[ ]:
```
Text(0.5, 0.98, 'Pair plot of LLE first 4 components')
```



LLE first finds the k-nearest neighbors of the points and approximates each data vector as a weighted linear combination of its k-nearest neighbors. It computes the weights that best reconstruct the vectors from its neighbors, then produce the low-dimensional vectors best reconstructed by these weights.

From the pair plot of the first four LLE dimensionalities, we observe: - The manifold for the distribution form a specific linear shape. This is consistent across all 4 LLE dimensions. - the LLE1 seems to be inversely coorelated with LLE 2, 3 and 4. - whereas the other dimensions LLE 2, 3 & 4 are positively correlated to each other - LLE tries to preserve the geometric features of the original non-linear feature structure

Comparison of LLE embeddings with Word2Vec embeddings

[ ]:
```
# map of test_set words and their word vectors after lle
test_embedding_lle = {}
for key, i in zip(test_embedding.keys(), range(len(lle_test))):
    test_embedding_lle[key] = lle_test[i]
```

[ ]:
```
lle_test_df = pd.DataFrame(lle_test)
cos_sim_lle = cosine_similarity(lle_test_df.iloc[:,:].values, Y=None,␣
↪dense_output=False)
cos_df_lle = pd.DataFrame(cos_sim_lle, index = test_embedding_lle.keys(), columns =␣
↪test_embedding_lle.keys())
```

Pair-wise cosine similarity: LLE

[ ]:
```
cos_df_lle
```

[ ]:

	the	world	enter	…	subtropics	proper	heatwaves
the	1.000000	0.969541	0.074335	…	-0.282073	-0.182722	-0.458161
world	0.969541	1.000000	0.280376	…	-0.327968	-0.169072	-0.531707
enter	0.074335	0.280376	1.000000	…	-0.193928	-0.446946	0.038208
cold	0.741747	0.554983	-0.526115	…	-0.068886	-0.134912	-0.112658
mode	0.890736	0.959242	0.512658	…	-0.259668	-0.377069	-0.417795
…	…	…	…	…	…	…	…
irregularly	-0.188364	-0.173012	-0.453866	…	-0.329261	0.998226	-0.445129
periodic	-0.265418	-0.286919	-0.474136	…	-0.551150	0.930601	-0.068765
subtropics	-0.282073	-0.327968	-0.193928	…	1.000000	-0.379516	-0.118824
proper	-0.182722	-0.169072	-0.446946	…	-0.379516	1.000000	-0.408676
heatwaves	-0.458161	-0.531707	0.038208	…	-0.118824	-0.408676	1.000000

[3913 rows x 3913 columns]

Pir-wise cosine similarity: Word2vec

[ ]:
```
cos_df_w2v
```

[ ]:

	the	world	enter	…	subtropics	proper	heatwaves
the	1.000000	0.937338	0.968786	…	0.940187	0.959345	0.971963
world	0.937338	1.000000	0.967958	…	0.941228	0.961900	0.965301
enter	0.968786	0.967958	1.000000	…	0.972888	0.985430	0.990646
cold	0.952239	0.893898	0.971733	…	0.949144	0.958023	0.967983
mode	0.960934	0.986443	0.989063	…	0.959360	0.983500	0.989856

```
…              …         …         …  …          …         …         …
irregularly  0.957003  0.942887  0.984662  …   0.966204  0.989548  0.987611
periodic     0.972193  0.934128  0.986972  …   0.964382  0.983412  0.986317
subtropics   0.940187  0.941228  0.972888  …   1.000000  0.969002  0.969213
proper       0.959345  0.961900  0.985430  …   0.969002  1.000000  0.990005
heatwaves    0.971963  0.965301  0.990646  …   0.969213  0.990005  1.000000

[3913 rows x 3913 columns]
```

From the above 2 dataframes of pair-wise cosine similarities, we can observe the cosine similarity values of each word in the test set and every other word in the set in the LLE and word2vec embedding space.

- Considering the absolute cosine similarities, we notice the cosine similarities are higher in the word2vec embedding space compared to the LLE embedding space. For instance considering the words 'world' & cold':

  LLE cosine similarity : 0.554983

  Word2vec cosine similarity : 0.893898

[ ]:
```
with open('test_embedding_lle.txt', 'w', encoding="utf-8") as f:
    f.write('%s %s\n' % (len(test_embedding_lle.keys()), 4))
    for key, value in test_embedding_lle.items():
        f.write('%s %s %s %s %s\n' % (key, value[0],value[1],value[2],value[3]))
```

[ ]:
```
lle_model = gensim.models.KeyedVectors.load_word2vec_format('test_embedding_lle.txt',␣
↪binary=False)
```

[ ]:
```
lle_model.most_similar('carbon')
```

[ ]:
```
[('methane', 0.9999682903289795),
 ('dioxide', 0.9995670318603516),
 ('greenhouse', 0.9947207570075989),
 ('activities', 0.9936545491218567),
 ('co', 0.9923534393310547),
 ('humaninduced', 0.99137282371521),
 ('emissions', 0.9912731051445007),
 ('gas', 0.9896917939186096),
 ('absence', 0.9876093864440918),
 ('21st', 0.9762848615646362)]
```

[ ]:
```
model.most_similar('carbon')
```

[ ]:
```
[('methane', 0.9804165363311768),
 ('chlorofluorocarbons', 0.9779826402664185),
 ('dioxide', 0.9746006727218628),
 ('nitrous', 0.9659712314605713),
 ('96', 0.9649980068206787),
 ('tropospheric', 0.9622126817703247),
 ('49', 0.9581986665725708),
 ('nonco2', 0.9581005573272705),
 ('oxide', 0.9576266407966614),
 ('release', 0.9563332200050354)]
```

We see the word 'carbon' is most similar to 'methane' in the LLE subspace and in the word2vec subspace. Similarly considering the cosine similarities for 'carbon' & 'dioxide':

LLE subspace : 0.9995670318603516

Word2vec subspace : 0.9746006727218628

In both the subspace, the most similar words appear in similar context as both 'dioxide' and 'methane' are relevant to 'carbon'

[ ]:
```
lle_model.most_similar('glaciers')
```

[ ]:
```
[('polar', 0.9999985098838806),
 ('rat', 0.9999204874038696),
 ('sheet', 0.9998705387115479),
 ('melt', 0.9997907876968384),
 ('west', 0.9995099306106567),
 ('mass', 0.9994706511497498),
 ('lose', 0.9992210865020752),
 ('ice', 0.9992193579673767),
 ('greenland', 0.9987722039222717),
 ('cover', 0.9987517595291138)]
```

[ ]:
```
model.most_similar('glaciers')
```

[ ]:
```
[('antarctic', 0.9970004558563232),
 ('greenland', 0.9952079653739929),
 ('loss', 0.988825798034668),
 ('antarctica', 0.9882369637489319),
 ('sheet', 0.9868824481964111),
 ('alpine', 0.9844338297843933),
 ('shrinkage', 0.9840260148048401),
 ('snow', 0.9806098341941833),
 ('landbased', 0.9798789024353027),
 ('mass', 0.9784616231918335)]
```

Considering the word 'glacier', we see both the models identify similar context words 'greenland' and 'sheet'. For the word 'sheet',

LLE : 0.9998705387115479

Word2vec: 0.9868824481964111

Similarly considering the context word 'greenland':

LLE: 0.9987722039222717

Word2vec: 0.9952079653739929

[ ]:
```
lle_model.most_similar('june')
```

[ ]:
```
[('fact', 0.9999924302101135),
 ('december', 0.9999922513961792),
 ('write', 0.9999914169311523),
 ('on', 0.9999827146530151),
 ('6', 0.9999823570251465),
 ('august', 0.999946653842926),
```

```
 ('observations', 0.999942421913147),
 ('2008', 0.9999364614486694),
 ('suggest', 0.9999234676361084),
 ('2018', 0.9999197721481323)]
```

[ ]:
```
lle_model.similarity("june","july")
```

[ ]:
```
0.982526
```

[ ]:
```
model.most_similar('june')
```

[ ]:
```
[('july', 0.9989757537841797),
 ('january', 0.9989748001098633),
 ('confirm', 0.9987897276878357),
 ('april', 0.9987339973449707),
 ('october', 0.9985716342926025),
 ('1988', 0.9983054399490356),
 ('august', 0.9980118870735168),
 ('february', 0.9979966282844543),
 ('december', 0.9978994727134705),
 ('seven', 0.9976174831390381)]
```

For the word 'june', we observe that LLE model doesn't identify the context words efficienty as we obtain many out of context words as similar words like 'write', 'fact', 'suggest'. While W2vec model identifies similar words in the right context (month words) such as 'july', 'january', 'december'.

Considering the words 'june' and 'july', we observe a higher cos similarity in Word2vec subspace than LLE:

LLE cosine similarity: 0.982526

Word2vec cosine similarity: 0.9989757537841797

KNN graph for LLE

[ ]:
```
from sklearn.neighbors import kneighbors_graph
A_lle = kneighbors_graph(lle_test_df, 5)
```

[ ]:
```
A_lle.toarray()
```

[ ]:
```
array([[0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

We obtain the adjacency matrix, which has the value 1 for the 5 closest neighbours of each word and 0 for every other datapoint.

Comparing the adjacency matrix for the word2vec embeddings and the the LLE emeddings for the test set, we obtain the f1_score of the embeddings indicating the extent of similarity between them.

The f1-score is particularly useful in this case when the class distubution is uneven. It specifies on average what percent of neighbors for each word are same in word2vec embeddings and LLE embeddings.

[ ]:
```
A_lle = A_lle.toarray()
```

[ ]:
```
row,column= A_lle.shape
```

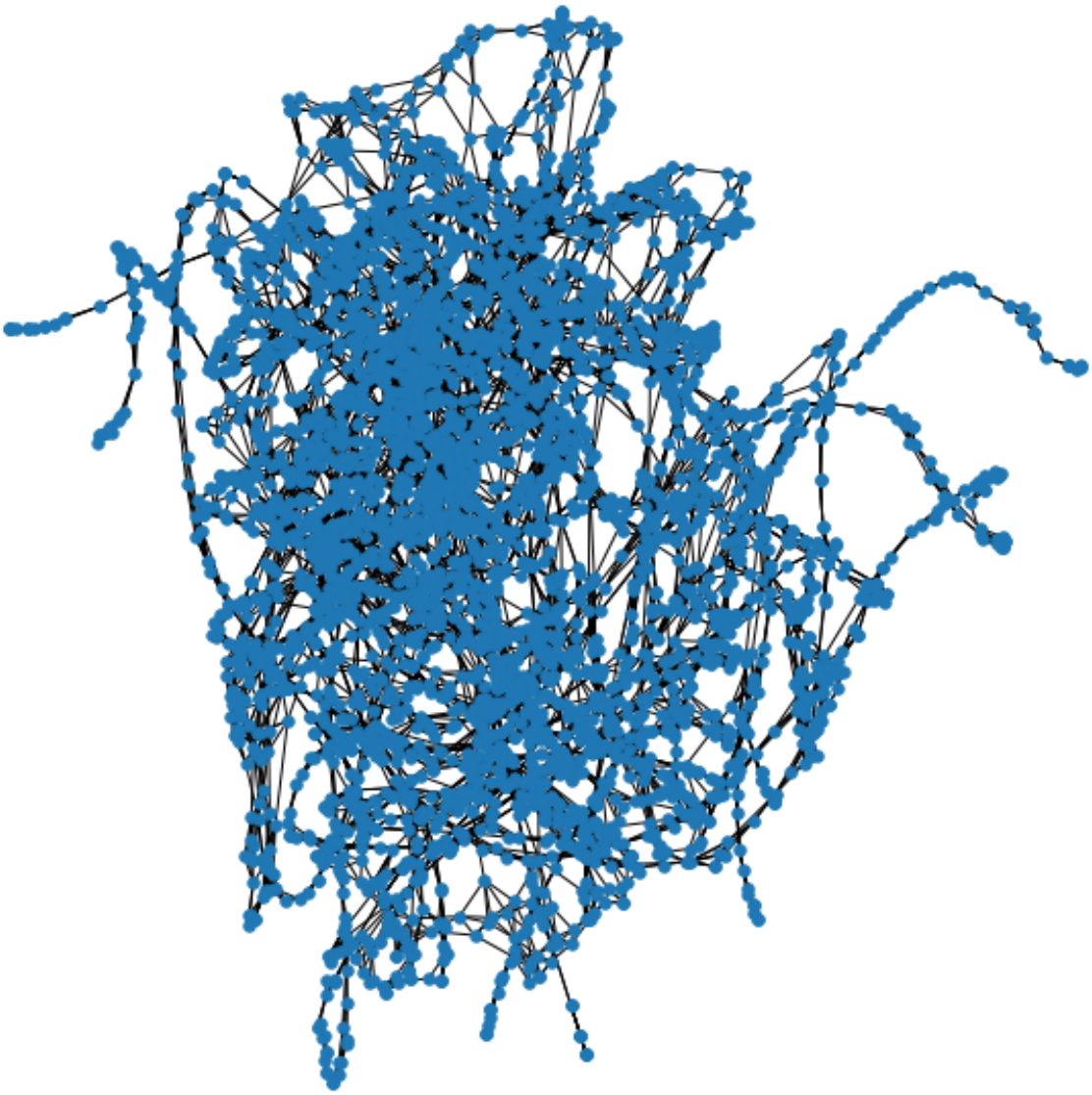
[ ]:
```
f1_score_lle_sum =0

for i in range(column):
  f1_score_lle_sum = f1_score_lle_sum + f1_score(A_word2vec[:,i], A_lle[:,i])
f1_score_lle = f1_score_lle_sum/column
```

[ ]:
```
f1_score_lle
```

[ ]:
```
0.1253260163010702
```

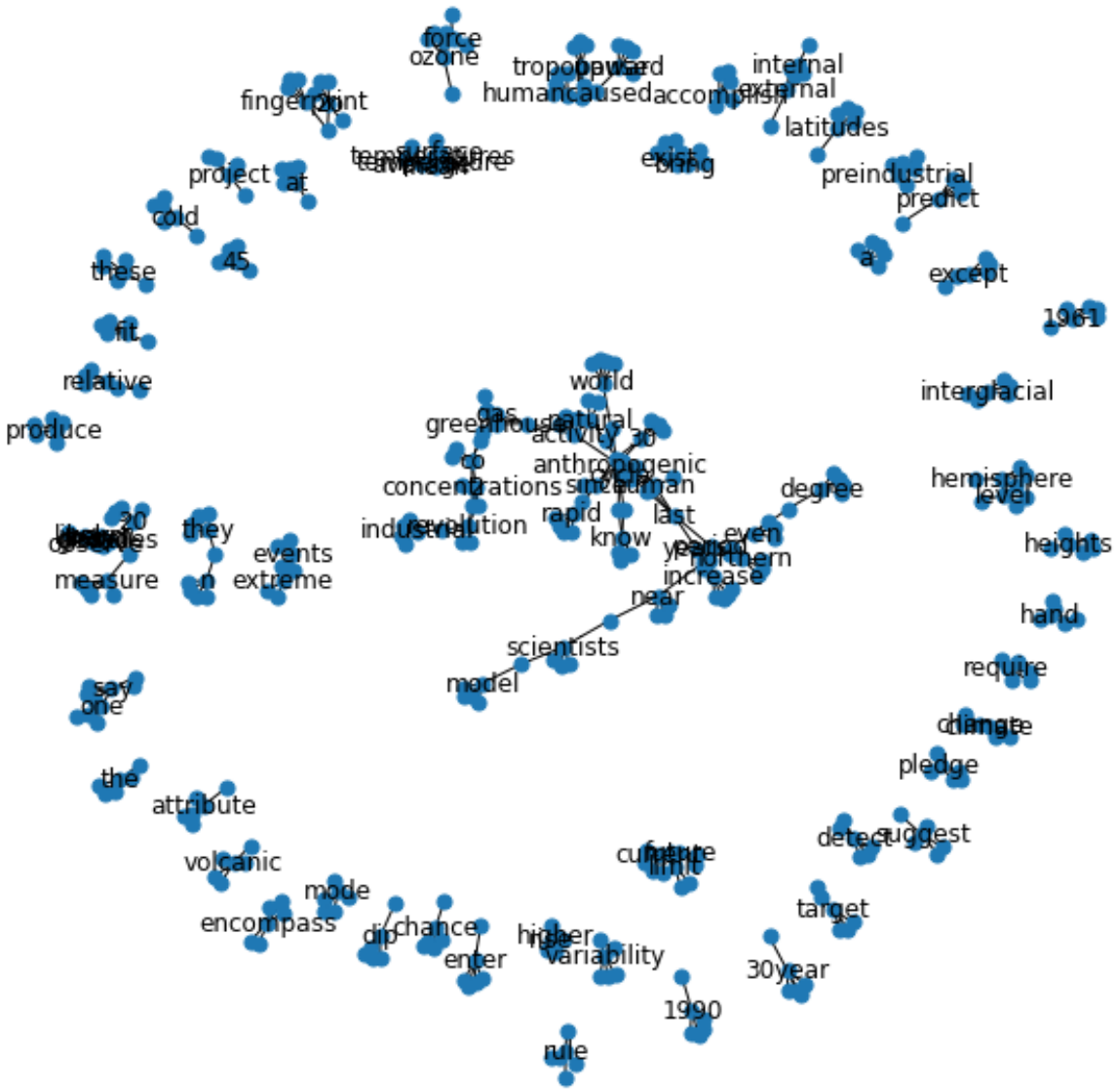
On comparing the KNN-graphs (sparce matrix) columnwise, we compute the average f1_score for the LLE and the Word2vec test embedding. We observe that only 12% of the neighbours of each word in the word2vec embeddings space are same as neighbours in the LLE embedding space. Thus LLE is able to maintain the semantic similarity in global context of the word vectors only to small extent . Since LLE is based on local piece-wise embeddings, it doesn't retain the global context.

[ ]:
```
fig = plt.figure(figsize=(8,8))
show_graph(A_lle)
```



[ ]:
```
test_embedding_lle_100 = dict(itertools.islice(test_embedding_lle.items(),101))
test_word_labels_lle = make_label_dict(test_embedding_lle_100.keys())
```

[ ]:
```
fig = plt.figure(figsize=(8,8))
show_graph_with_labels(A_lle[:101], test_word_labels_lle)
```



From visualization of KNN graph of LLE embeddings for subset of words from test set, we observe that the semantic relations between words is preserved in the LLE subspace: - Words like ‘climate’ and ‘change’ appear closer as was the case in w2vec embeddings - Words like ‘greenhouse’ and ‘c02’ appear closer as was the case in w2vec embeddings. - Words like ‘external’ and ‘internal’ which appear in similar context are together as was the case with w2vec embeddings. - We also notice similar context words like ‘future’ and ‘current’ close to each other (similar to w2vec embeddings) - Words ‘northern’ and ‘cold’ are not closer to each other (contrary to w2vec embeddings) - Words ‘increase’ and ‘higher’ are not closer to each other (contrary to w2vec embeddings)