Assignment 2&3
50 marks

Modified COALS

Last date for submission
20-September-2024

Implement a modified version of COALS algorithm

Implement a modified version of COALS (refer to this paper for COALS implementation) using the COVID corpus.

(a) Instead of the using correlation, use the ratio of the probabilities $\frac{p_{ij}}{p_{jk}}$ to obtain the co-occurrence matrix

(b) Try to limit the vocabulary size to ~7K to compute the word embedding.

(c) Display the vocabulary size and the size of the matrix.

(d) Identify five nouns and verbs (relevant to COVID19) from the corpus.

(e) Generate five similar words for each (chosen in step b) and display the result with cosine distance for each one.

(f) Use multi-dimensional scaling to visualize certain concepts (relevant to COVID 19) as shown in Fig.1. Take three concepts that you feel are relevant to this corpus and use a maximum of 10 words per concept. Make sure that the plot is clear and not cluttered

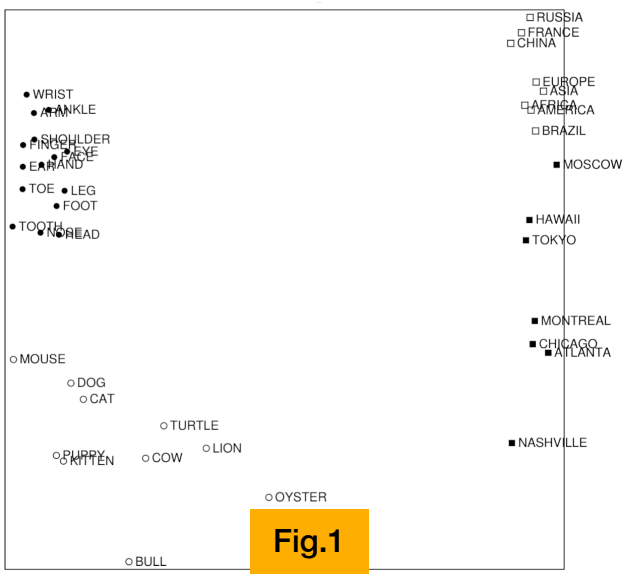

Fig.1

Note: Follow all the instructions given in Assignment 1.

You may submit these two assignments as one single assignment

	Tasks	Total marks
Assignment 2	Forming the cooccurrence matrix (a)	25 marks
Assignment 3	Similarity measure	5 marks
Assignment 3	Listing of similar words	5 (1 for each)
Assignment 3	Graph with clear visualisation	15 marks

Additional Information

Large matrices are very common in machine learning and NLP applications. There are several ways to handle large matrix. If the matrix is sparse, then you may try https://docs.scipy.org/doc/scipy/reference/sparse.html. Otherwise, you may use pandas or use h5py.

Same code is given for your convenience -You may use it, but at your own risk 😄

```
import numpy as np
import pandas as pd

df = pd.DataFrame(np.zeros(
                         (vocab_count, vocab_count)),
                         dtype=np.int16)
hdf5file = pd.HDFStore('cm_hal.hdf5', 'w') #file size = 1.8G
hdf5file['data'] = df
hdf5file.close()
```

```
#Direct approach using h5py
import h5py
f = h5py.File('halmatrix.hdf5','w')
hm = f.create_dataset("HAL_CM",
                         (vocab_count, vocab_count),
                         dtype=np.int16, compression="gzip"))

#Access the matrix 'HAL_CM' using hm or using f['HAL_CM']
# hm[1:], hm[:1], hm[1,1],hm[100,23]... or
a = f['HAL_CM'][1,234]
```

```
#Creating word vectors – left to right
grams11 = ngrams(tokens, 11, pad_right=True, pad_left=True)

f = h5py.File('matrix.hdf5', 'w')

coals = f.create_dataset('L2R',
                         (vocab_count, vocab_count),
                         dtype=np.int16)

ramp = [0, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
# Note this is useful for HAL. For COALS, the middle token is the
target word
for index, grams in enumerate(grams11):

    if not None in grams:
        # Left to right
        for idx, gram in enumerate(grams):
           coals[vocab_idx[grams[0]],
                    vocab_idx[grams[idx]]] += ramp[idx]
```

```
#To reduce the matrix size, you may want to reduce the
# vocabulary size.

freq_dist = FreqDist(tokens)
vocab = []
for word in tokens:
     if not stop_words(word) and freq_dist[word] > 20:
        vocab.append(word)
```