Assignment 2

ECSE344L (NLP)

Word Vector/Embedding Generation

In the assignment 1, we have gone through Tokenization, steaming, lemmatization, BOW and TF_IDF on the given Times of India news headline dataset.

In this lab, we will create our first word embedding for the given dataset using SVD and co-occurrence:

1. **Create co-occurrence matrix of the corpus with**
   a. Window size 1 named as X_1
   b. Window size 2 named as X_2
   c. Window size 3 named as X_3
   d. Window size 4 named as X_4
2. **Apply SVD on {X_1, X_2, X_3, X_4} and create a work vector matrix with following k values:**
   a. K=50 named as U_1_50, U_2_50, U_3_50, U_4_50 respectively for X_1, X_2, X_3, X_4
   b. K=100 named as U_1_100, U_2_100, U_3_100, U_4_100 respectively for X_1, X_2, X_3, X_4
   c. K=200 named as U_1_200, U2_200, U_3_200, U_4_200 respectively for X_1, X_2, X_3, X_4
   d. K=300 named as U_1_300, , U_2_200, U_3_200, U_4_200 respectively for X_1, X_2, X_3, X_4
3. **After generating U_w_k apply PCA/TSNE using to convert K dimension value to two dimension and plot random 100 words.**

Sample code to perform SVD:

```
# Decomposition and Reconstruction
keep=50
U, S, V = np.linalg.svd(X)
tU, tS, tV = U[:, 0:keep], S[0:keep], V[0:keep, :]
Xnew = np.matmul(np.matmul(tU, np.diag(tS)), tV)
print("Reconstruction Error: ",np.mean(abs(X-Xnew)))
```

Sample code for plotting:

```
#model[‘sample word’] will return word vector of sample word
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
%matplotlib inline
def display_pca_scatterplot(model, words=None, sample=0):
    word_vectors = np.array([model[w] for w in words])
    twodim = PCA().fit_transform(word_vectors)[:,:2]
    plt.figure(figsize=(6,6))
    plt.scatter(twodim[:,0], twodim[:,1], edgecolors='k', c='r')
    for word, (x,y) in zip(words, twodim):
        plt.text(x+0.05, y+0.05, word)
    plt.savefig("test.png")
    plt.show()



def display_tsne_scatterplot(model, words=None, sample=0):
    word_vectors = np.array([model[w] for w in words])
    twodim = TSNE().fit_transform(word_vectors)[:,:2]
    plt.figure(figsize=(6,6))
    plt.scatter(twodim[:,0], twodim[:,1], edgecolors='k', c='r')
    for word, (x,y) in zip(words, twodim):
        plt.text(x+0.05, y+0.05, word)
    plt.savefig("test.png")
    plt.show()

display_pca_scatterplot(model,['coffee', 'tea', 'beer', 'wine', 'brandy
', 'rum', 'champagne', 'water','spaghetti', 'borscht', 'hamburger', 'pi
zza', 'falafel', 'sushi', 'meatballs','dog', 'horse', 'cat', 'monkey',
'parrot', 'koala', 'lizard','frog'])
```