NLP Assignment 2

1 : Negative Sampling:

Word2Vec is a technique that uses either CBOW or Skip-Gram methods to produce word embeddings in numerical vectors. These vectors enable the identification of relationships between words. However, both models require significant amounts of data tweaking during training and testing, which can become computationally expensive, especially with large vocabularies. Negative Sampling is an approximation technique that simplifies this process by sampling a subset of data points and tweaking only those. By selecting words with low unigram probability, the model can avoid checking every possible output and instead focus on negative samples that are far from the correct output. This approach reduces computational costs and speeds up the training and evaluation process while also preventing overfitting and improving model accuracy.

2 : Semantic similarity

Semantic similarity refers to the extent to which two words or phrases convey similar meanings. Word embeddings are a popular way to represent words as numerical vectors in a high-dimensional space, where each dimension corresponds to a semantic or syntactic property of the word.

Measuring semantic similarity using word embeddings involves computing the distance or similarity between the vector representations of two words or phrases. The closer two vectors are in the embedding space, the more similar the words are in their meanings.

Two popular techniques for measuring semantic similarity using word embeddings:

1. Cosine similarity: This technique measures the cosine of the angle between the vector representations of two words or phrases. The cosine similarity score ranges from -1 to 1, with higher values indicating more remarkable similarity. A score of 1 means the words have identical meanings, while a score of -1 means they have opposite meanings.

2. Euclidean distance: This technique measures the distance between the vector representations of two words or phrases in the embedding space. The smaller the distance, the more similar the words are in meaning. This method is useful for detecting synonyms, as words with similar meanings tend to cluster together in the embedding space.

Measuring semantic similarity using word embeddings is a powerful tool in natural language processing with many applications, including information retrieval, text classification, and machine translation.

3: Implementation :

Implementation of both working models is in Model 1 and Model 2.

Statistical Model :

Outputs →

Co-occurrence Matrix

```
In [19]: Matrix

Out[19]: array([[8.9366e+04, 1.8000e+01, 7.4600e+02, ..., 1.7000e+01, 5.0000e+00,
                 9.0000e+00],
                [1.8000e+01, 0.0000e+00, 1.0000e+00, ..., 0.0000e+00, 0.0000e+00,
                 0.0000e+00],
                [7.4600e+02, 1.0000e+00, 2.2000e+01, ..., 0.0000e+00, 0.0000e+00,
                 0.0000e+00],
                ...,
                [1.7000e+01, 0.0000e+00, 0.0000e+00, ..., 2.0000e+00, 0.0000e+00,
                 0.0000e+00],
                [5.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00,
                 0.0000e+00],
                [9.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00,
                 0.0000e+00]])
```

U Matrix :

```
In [23]: k = 300
         word_embeddings = U[:, :k]

         # # Print the word embeddings
         print(word_embeddings)
```

```
[[-2.98731000e+02 -9.00031359e+00  8.00338682e+00 ... -2.26745960e-02
  -6.00164691e-02 -7.57843818e-03]
 [-5.87446982e-02 -2.13035037e-02 -3.06379542e-03 ...  4.67939478e-02
   8.34672977e-03  2.27364596e-02]
 [-2.53949486e+00  9.53828307e-02 -1.72962016e+00 ... -2.84213128e-01
  -3.33156428e-01 -1.56558859e+00]
 ...
 [-5.64072496e-02 -1.18754339e-02  2.60959370e-02 ...  2.19792268e-02
   4.45493677e-03 -3.66144733e-03]
 [-1.66157537e-02  7.78979922e-03  1.16176402e-02 ...  1.51545234e-02
   1.29643762e-02  2.38170958e-03]
 [-3.06689227e-02 -1.00788146e-03  7.49364253e-03 ...  9.36491946e-03
  -1.05776122e-02  7.99380588e-04]]
```

Predictive Model :

Loss Output Per Epoch :

→8.947362015847

→6.283495027556

→4.128759032187

→2.163944023815

→0.992847019475

→0.530287032187

→0.275416024825

→0.157368920134

→0.107549850726

→0.086902814386