CSSE15: NLP with Deep Learning

Assignment Report: Co-occurrence Matrix and Word Embeddings

Name: Ansh Gudibanda
Branch: ECE
Roll Number: 22207
Date of Submission: 29 August 2025

This report documents the implementation of word embeddings using a co-occurrence matrix as covered in the assignment for the subject CSSE15: NLP with Deep Learning. The task is to build count-based word vectors, reduce their dimensionality, and visualize them in two dimensions. The implementation is done in Python using NLTK, NumPy, scikit-learn, and Matplotlib.

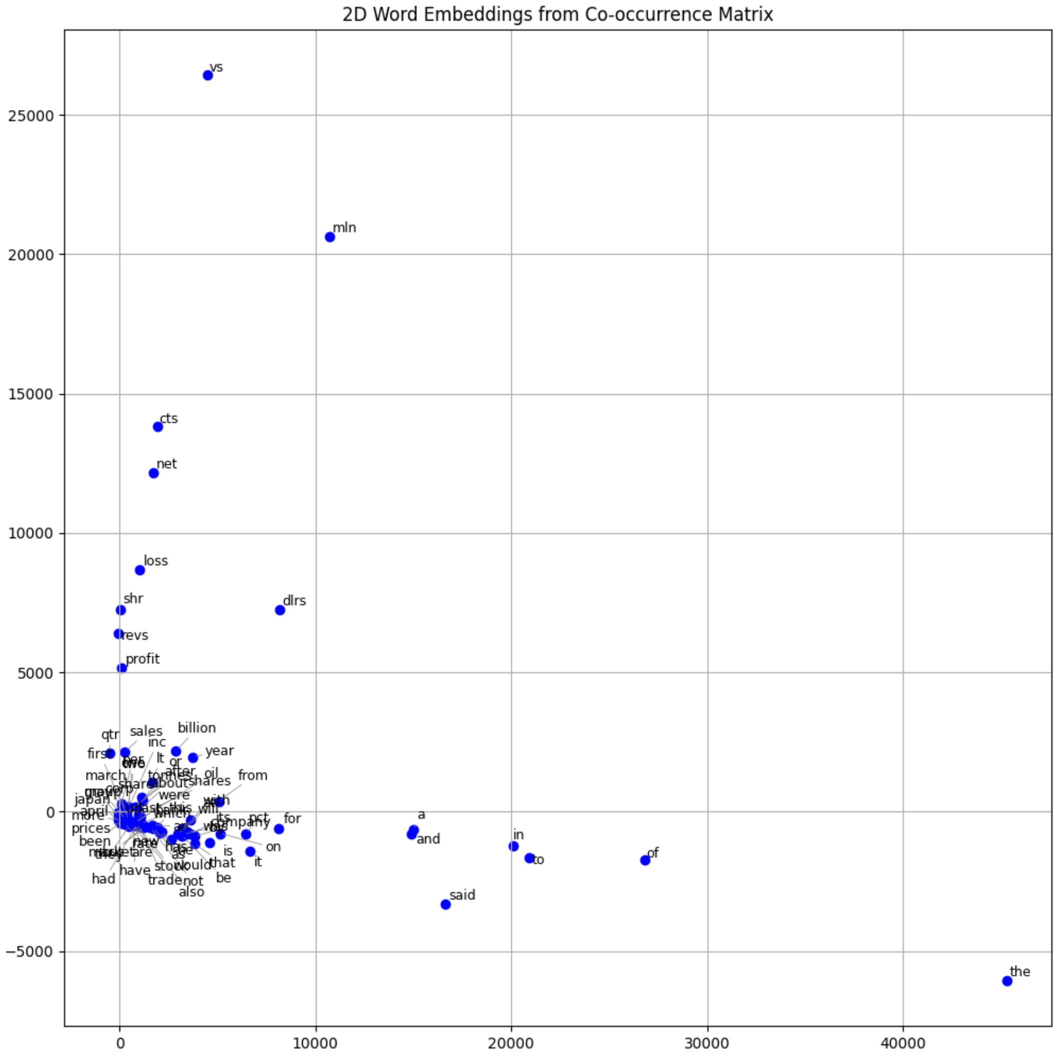
1. Methods Implemented

1. Distinct Words Extraction: A function `get_distinct_words` extracts all unique words (word types) from the input corpus after tokenization and preprocessing (lowercasing, filtering alphabetic tokens).
2. Co-occurrence Matrix Construction: A function `build_cooccurrence_matrix` builds a symmetric matrix where entry (i,j) represents the number of times word j appears within a window of size n around word i.
3. Dimensionality Reduction: The function `reduce_dimensions` uses Principal Component Analysis (PCA) to reduce the high-dimensional co-occurrence matrix into k-dimensional word embeddings.
4. 2D Plotting: The function `plot_embeddings` visualizes the reduced embeddings in two dimensions. Each word is plotted as a point, with its label displayed nearby.
5. Corpus: The Reuters corpus from NLTK was used as the dataset, which contains real-world news articles.

2. Results

The implementation successfully generated a co-occurrence matrix from the corpus, reduced its dimensionality using PCA, and visualized a subset of word embeddings in 2D space. Adjustments were made to ensure that text labels in the plot do not overlap, by using the `adjustText` library.

Figure 1: 2D Word Embeddings from Co-occurrence Matrix



3. GitHub Repository

The full implementation code has been uploaded to GitHub and can be accessed at: NLP_Assignment_1 Repository

4. Conclusion

All tasks outlined in the assignment were implemented: distinct word extraction, co-occurrence matrix construction, dimensionality reduction, and 2D visualization. The code is available on GitHub, and this report documents the workflow.