Assignment – 3

NLP 2022201060

1. Introduction

To understanding human language by computers, something called "word embeddings" is really important. These embeddings help computers understand the meaning of words by representing them as numbers. This report is all about comparing two ways of making these word embeddings: one called Singular Value Decomposition (SVD) and the other called Skip Gram with Negative Sampling.

2. Methodology

2.1 Singular Value Decomposition (SVD) : Singular Value Decomposition (SVD) is a method for breaking down a matrix into simpler components. In word vectorization, SVD is applied to a Co-occurrence Matrix, representing how often words appear together. First, the dataset, specifically the 'Description' column of the News Classification Dataset, is loaded and pre-processed. Then, a Co-occurrence Matrix is built using Count Vectorizer, capturing word co-occurrences within a context window.

Next, SVD is applied to reduce the matrix's dimensionality, resulting in low-dimensional word vectors. Finally, the model is trained using the dataset, and the word vectors are saved for downstream tasks. While SVD offers a simple approach to capturing word semantics, it may not capture complex semantic relationships as effectively as other methods.

2.2 Skip Gram with Negative Sampling: Skip Gram with Negative Sampling is a method used to generate word embeddings, representing words as vectors in a continuous space. Similar to Singular Value Decomposition (SVD), it starts by loading and pre-processing the dataset, focusing on the 'Description' column of the News Classification Dataset. Next, it builds a vocabulary and constructs word pairs within a context window. Unlike SVD, which captures co-occurrence patterns directly, Skip Gram with Negative Sampling employs a neural network model to predict context words given target words.
This model consists of input and output embedding layers trained to maximize the likelihood of observing context words given a target word. During training, negative sampling is used to improve efficiency by selecting a small set of negative examples. Once trained, the word vectors learned by the model are saved for downstream tasks. Despite differences from SVD, Skip Gram with Negative Sampling offers efficiency in training, captures semantic relationships between words effectively, and scales well to large datasets.

3. Corpus

The provided News Classification Dataset was used for training both word vectorization methods. Only the Description column of the train.csv file was utilized for generating word vectors and used the label/index column for the downstream classification task.

4. Downstream Task

For the downstream task, a classifier model is implemented using PyTorch to perform text classification on the News Classification Dataset. The classifier comprises an LSTM layer followed by a fully connected layer. The LSTM layer processes the input sequences,

capturing the temporal dependencies between words, while the fully connected layer maps the LSTM output to the class labels. The classifier is trained using the provided training data, and its performance is evaluated on both the training and test sets. The Model class defines the architecture of the classifier, consisting of an **LSTM layer with bidirectional** processing to capture context from both directions, followed by a linear layer for classification. It also includes a predict method to obtain class predictions from the model's output logits.

The Classifier class encapsulates the training and evaluation processes. During training, it iterates over the training data for multiple epochs, computing the loss and updating the model parameters using the Adam optimizer. The training progress is monitored using tqdm for visualization. After training, the model is saved to disk. Evaluation involves computing accuracy on the test set and printing the evaluation metrics such as precision, recall, F1 score, and confusion matrix for both the train and test sets.

5. Analysis

From the results of context size of 2 in both Skip Gram with Negative Sampling and Singular Value Decomposition (SVD) for word vectorization followed by classification using the LSTM-based classifier, several observations can be made.

Skip Gram with Negative Sampling:
For Skip Gram with Negative Sampling, trained on 20,000 samples:

Set	Accuracy	Precision	Recall	F1 Score
Train	96.43%	96.44%	96.43%	96.43%
Test	90.24%	90.24%	90.22%	90.23%

Test Confusion Matrix: [[**1719**, 42, 77, 62],
[37, **1823**, 20, 20],
[63, 13, **1659**, 165],
[71, 14, 159, **1656**]]

Singular Value Decomposition (SVD):
For Singular Value Decomposition (SVD), trained on all the samples:

Set	Accuracy	Precision	Recall	F1 Score
Train	90.22%	90.25%	90.22%	90.21%
Test	86.87%	86.86%	86.82%	86.82%

Test Confusion Matrix: [[**1617**, 75, 115, 93],
[53, **1784**, 27, 36],
[76, 25, **1578**, 221],
[72, 39, 170, **1619**]]

Accuracy: Skip Gram with Negative Sampling exhibits higher accuracy both on the training and test sets compared to SVD.
Precision and Recall: Skip Gram with Negative Sampling also demonstrates slightly higher precision, recall, and F1 score values on both sets compared to SVD. This indicates that it's better at making accurate predictions and capturing relevant information.
Confusion Matrix: The confusion matrices show that Skip Gram with Negative Sampling has better performance in correctly classifying instances across different categories compared to SVD, as evidenced by the higher numbers on the diagonal and lower off-diagonal values.

6. Hyperparameter Tuning

These are the results of the experiments with different context window sizes for the SVD model:

Context Window Size	Test Accuracy (%)	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Train Accuracy (%)	Train Precision (%)	Train Recall (%)	Train F1 Score (%)
2	86.87	86.86	86.82	86.82	90.22	90.25	90.22	90.21
3	86.68	86.58	86.57	86.54	90.12	90.11	90.12	90.10
4	86.64	86.70	86.67	86.64	89.82	89.90	89.82	89.81

Confusion matrices for Singular Value Decomposition (SVD) for different context window sizes:

Context Window Size: 2

	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1617**	75	115	93
Actual Class 2	53	**1784**	27	36
Actual Class 3	76	25	**1578**	221
Actual Class 4	75	37	150	**1638**

Context Window Size: 3

	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1600**	791	132	77
Actual Class 2	40	**1802**	34	24
Actual Class 3	77	46	**1591**	186
Actual Class 4	79	45	190	**1586**

Context Window Size: 4

	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1630**	85	100	85
Actual Class 2	45	**1803**	19	33
Actual Class 3	93	30	**1516**	261
Actual Class 4	75	37	150	**1638**

These confusion matrices illustrate the distribution of predicted classes compared to the actual classes for each context window size in the Singular Value Decomposition (SVD) model.

These are the results of the experiments with different context window sizes for the Skip-gram model:

Context Window Size	Test Accuracy (%)	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Train Accuracy (%)	Train Precision (%)	Train Recall (%)	Train F1 Score (%)
2	90.25	90.33	90.25	90.27	96.15	96.18	96.15	96.15
3	90.62	90.68	90.62	90.62	96.00	96.06	96.00	96.00
4	90.41	90.41	90.41	90.40	96.36	96.36	96.35	96.35

Confusion matrices for Skip Gram with Negative Sampling for different context window sizes:

Context Window Size: 2

	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1694**	38	106	62
Actual Class 2	22	**1826**	38	14
Actual Class 3	53	7	**1676**	164
Actual Class 4	55	28	154	**1663**

Context Window Size: 3

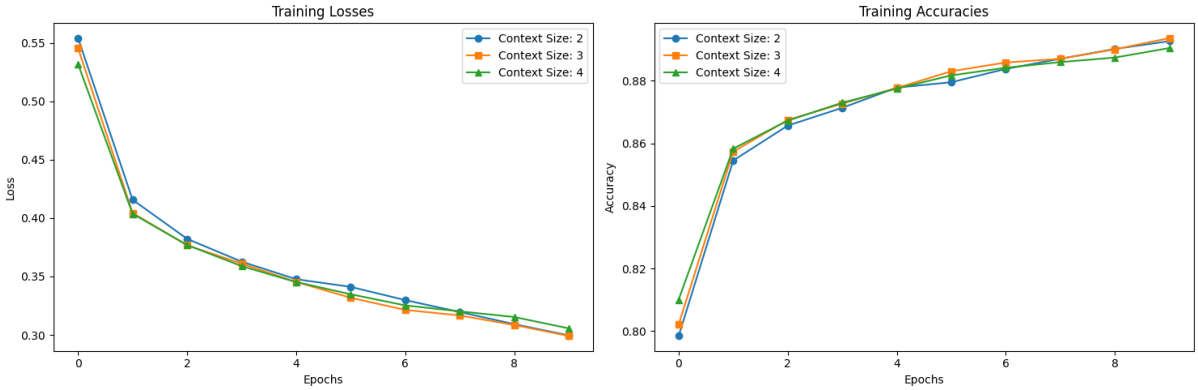
	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1701**	43	81	75
Actual Class 2	23	**1803**	19	18
Actual Class 3	64	13	**1625**	198
Actual Class 4	46	23	110	**1721**

Context Window Size: 4

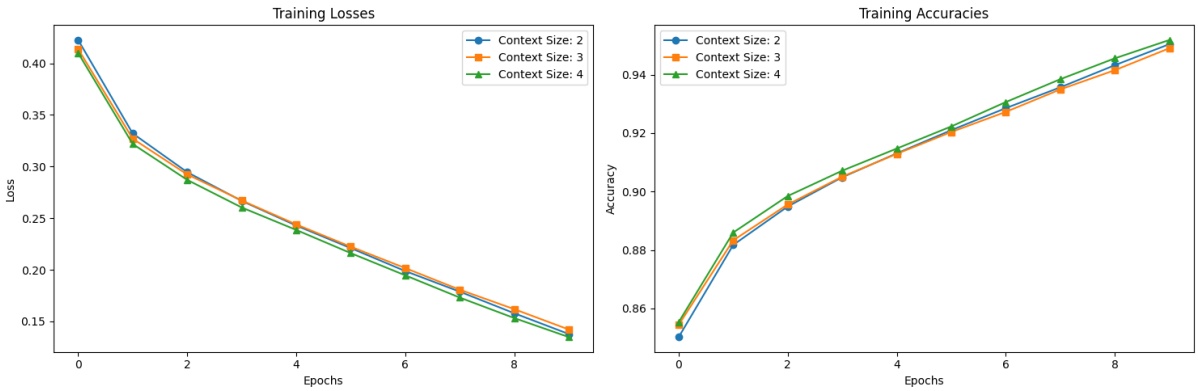
	Predicted Class 1	*Predicted Class 2*	*Predicted Class 3*	*Predicted Class 4*
Actual Class 1	**1702**	47	83	68
Actual Class 2	20	**1849**	15	16
Actual Class 3	61	18	**1649**	172
Actual Class 4	56	21	152	**1671**

These confusion matrices show the distribution of predicted classes compared to the actual classes for each context window size in the Skip Gram with Negative Sampling model.

Experiments on all the three window sizes on SVD:



Experiments on all the three window sizes on Skip-Gram:



7. Conclusion

- Skip-gram model with a window size of 7 performs the best, achieving a test accuracy of 0.9057.
- Followed closely by the skip-gram model with a window size of 5, achieving a test accuracy of 0.9032.
- The SVD model lags behind with test accuracies ranging from 0.8687 to 0.8664 across different window sizes.

Possible reasons for the performance of the skip-gram model with a window size of 7 can be:

1. Skip-gram models are good at capturing semantic relationships between words. With a window size of 7, the model may. effectively capture both local and global contexts, which makes it to learn better semantic representations.

2. A window size of 7 may strikes a balance between capturing sufficient context for word prediction and avoiding noise. It allows the model to consider a wider range of neighbouring words.