Introduction to NLP

Assignment-3

Submitted by - Ishit Bansal (2021101083)

Report

Analysis

SVD

```
Train Set:
Accuracy: 0.9055083333333334
Precision: 0.906025028929943
Recall: 0.9055083333333334
F1 Score: 0.9053426428469515
Confusion Matrix: [[26899   750  1314  1037]
 [  460 29134   182   224]
 [ 1016   299 25332  3353]
 [  921   244  1539 27296]]

Test Set:
Accuracy: 0.8877631578947368
Precision: 0.8882956075505102
Recall: 0.8877631578947368
F1 Score: 0.8875985193227585
Confusion Matrix: [[1679   51   98   72]
 [  33 1818   30   19]
 [  72   28 1557  243]
 [  78   21  108 1693]]
```

Skip Gram

```
Train Set:
Accuracy: 0.9737083333333333
Precision: 0.9742444474359087
Recall: 0.9737083333333333
F1 Score: 0.9737964490106757
Confusion Matrix: [[29020    89   565   326]
 [   97 29754   108    41]
 [   54    18 29441   487]
 [   85    14  1271 28630]]

Test Set:
Accuracy: 0.901578947368421
Precision: 0.9033446784072748
Recall: 0.901578947368421
F1 Score: 0.9020756205379947
Confusion Matrix: [[1690   24   95   91]
 [  21 1827   33   19]
 [  43    7 1697  153]
 [  49   14  199 1638]]
```

Comparison

We observe that SkipGram performs better than SVD on both the training and test sets in terms of accuracy, precision, recall and F1 score. This indicates that SkipGram is able to capture the semantics better. Higher precision and recall of SkipGram compared to SVD highlights its capability to generate more precise positive predictions and capture a larger portion of true positive cases.

SVD Shortcomings:

- SVD is computationally expensive (O(n^3) for nxn matrices) and requires a lot of memory for large datasets.
- SVD is not directly applicable to sparse matrices. Techniques like truncated SVD are used to approximate the decomposition for sparse matrices, but these methods have their own trade-offs and limitations.

SkipGram with Negative Sampling Shortcomings:

- Homonyms and polysemous words can have ambiguous embeddings as the model treats all occurrences of a word as a single entity without considering different senses or contexts.

- Training is computationally intensive as the negative sampling procedure requires sampling negative examples for each positive training instance.
- SkipGram typically uses a fixed-size context window around each target word. This limits the model's ability to capture long-range dependencies that span beyond the window size.

Hyperparameter Tuning

Context Window Sizes

- **2** : This window size focuses on neighboring words and is suitable for capturing local syntactic relationships.
- **5** : This window size balances capturing local context and some broader semantic associations.
- **10** : This window size targets more distant relationships and the overall semantic context within a sentence.

Performance

SVD

- Context Window Size = 2

```
Train Set:
Accuracy: 0.8882083333333334
Precision: 0.8913247614082769
Recall: 0.8882083333333334
F1 Score: 0.8883534000112535
Confusion Matrix: [[25969   920  1395  1716]
 [  423 28716   322   539]
 [  909   295 24467  4329]
 [  786   215  1566 27433]]

Test Set:
Accuracy: 0.8697368421052631
Precision: 0.8722722760788111
Recall: 0.8697368421052631
F1 Score: 0.86983440982138
Confusion Matrix: [[1608   78  107  107]
 [  37 1791   29   43]
 [  69   25 1517  289]
 [  61   16  129 1694]]
```

- Context Window Size = 5

```
Train Set:
Accuracy: 0.9055083333333334
Precision: 0.906025028929943
Recall: 0.9055083333333334
F1 Score: 0.9053426428469515
Confusion Matrix: [[26899   750  1314  1037]
 [  460 29134   182   224]
 [ 1016   299 25332  3353]
 [  921   244  1539 27296]]

Test Set:
Accuracy: 0.8877631578947368
Precision: 0.8882956075505102
Recall: 0.8877631578947368
F1 Score: 0.8875985193227585
Confusion Matrix: [[1679   51   98   72]
 [  33 1818   30   19]
 [  72   28 1557  243]
 [  78   21  108 1693]]
```

- Context Window Size = 10

```
Train Set:
Accuracy: 0.9116333333333333
Precision: 0.9126153228840616
Recall: 0.9116333333333333
F1 Score: 0.9116187184646952
Confusion Matrix: [[26348   917  1729  1006]
 [  161 29510   211   118]
 [  573   216 26881  2330]
 [  635   269  2439 26657]]

Test Set:
Accuracy: 0.8957894736842106
Precision: 0.8970712150879531
Recall: 0.8957894736842106
F1 Score: 0.8959381792567103
Confusion Matrix: [[1647   59  116   78]
 [  17 1837   29   17]
 [  42   19 1669  170]
 [  49   23  173 1655]]
```

- Combined

Context Window Size	Accuracy	Precision	Recall	F1_Score	Confusion Matrix
2	0.8697368421052631	0.8722722760788111	0.8697368421052631	0.86983440982138	[[1608 78 107 107] [37 1791 29 43] [69 25 1517 289] [61 16 129 1694]]
5	0.8877631578947368	0.8882956075505102	0.8877631578947368	0.8875985193227585	[[1679 51 98 72] [33 1818 30 19] [72 28 1557 243] [78 21 108 1693]]
10	0.8957894736842106	0.8970712150879531	0.8957894736842106	0.8959381792567103	[[1647 59 116 78] [17 1837 29 17] [42 19 1669 170] [49 23 173 1655]]

- Best context window size

  We observe that context size = 10 gives maximum accuracy on the test set.

SkipGram

- Context Window Size = 2

```
Train Set:
Accuracy: 0.9497416666666667
Precision: 0.9508833645462933
Recall: 0.9497416666666667
F1 Score: 0.9496591164105659
Confusion Matrix: [[28438   304   355   903]
 [  103 29787    20    90]
 [  699   229 26778  2294]
 [  267    74   693 28966]]

Test Set:
Accuracy: 0.8953947368421052
Precision: 0.8969339297630881
Recall: 0.8953947368421052
F1 Score: 0.8952788674747271
Confusion Matrix: [[1701   45   54  100]
 [  26 1835   14   25]
 [  82   21 1560  237]
 [  58   28  105 1709]]
```

- Context Window Size = 5

```
Train Set:
Accuracy: 0.9737083333333333
Precision: 0.9742444474359087
Recall: 0.9737083333333333
F1 Score: 0.9737964490106757
Confusion Matrix: [[29020    89   565   326]
 [   97 29754   108    41]
 [   54    18 29441   487]
 [   85    14  1271 28630]]

Test Set:
Accuracy: 0.901578947368421
Precision: 0.9033446784072748
Recall: 0.901578947368421
F1 Score: 0.9020756205379947
Confusion Matrix: [[1690   24   95   91]
 [  21 1827   33   19]
 [  43    7 1697  153]
 [  49   14  199 1638]]
```

- Context Window Size = 10

```
Train Set:
Accuracy: 0.9677916666666667
Precision: 0.9691314022270019
Recall: 0.9677916666666667
F1 Score: 0.967825186834218
Confusion Matrix: [[29103   136   271   490]
 [   80 29842    22    56]
 [  232    36 27540  2192]
 [  128    23   199 29650]]

Test Set:
Accuracy: 0.9073684210526316
Precision: 0.9094448803242379
Recall: 0.9073684210526316
F1 Score: 0.907279656801976
Confusion Matrix: [[1719   35   61   85]
 [  22 1842   19   17]
 [  75   12 1570  243]
 [  47   12   76 1765]]
```

- Combined

Context Window Size	Accuracy	Precision	Recall	F1_Score	Confusion Matrix
2	0.8953947368421052	0.8969339297630881	0.8953947368421052	0.8952788674747271	[[1701 45 54 100] [26 1835 14 25] [82 21 1560 237] [58 28 105 1709]]
5	0.901578947368421	0.9033446784072748	0.901578947368421	0.9020756205379947	[[1690 24 95 91] [21 1827 33 19] [43 7 1697 153] [49 14 199 1638]]
10	0.9073684210526316	0.9094448803242379	0.9073684210526316	0.907279656801976	[[1719 35 61 85] [22 1842 19 17] [75 12 1570 243] [47 12 76 1765]]

- Best context window size

  We observe that context size = 10 gives maximum accuracy on the test set.

Reasons

We observe that the context window size of 10 achieved the best performance using both SVD and SkipGram with Negative Sampling. This is because:

- A larger context window size allows the model to consider more words surrounding the target word. This helps capture diverse semantic relationships and associations.
- A larger context window size enables the model to capture long-range dependencies.
- A larger context window provides a more diverse range of contextual information leading to better word embeddings.