Word Vectorization

ASSIGNMENT –3 Report

Dataset for SVD and Skip Gram

Train.csv
No of Rows =15000

1. **SVD Word Vector**

To train and Test Accuracy for SVD

Hyperparameters

Embedding dim=150

Batch Size=300

Hidden size=300

No of hidden layers=2

Epoch=5

Window size=1

Train Accuracy= 78.329985178%
Test Set:77.72368421052631

Accuracy:
F1 Score: 0.7752065635158733
Precision: 0.7799399851780164
Recall: 0.7742416666666666
Confusion Matrix:
[[23678 1173 1706 2671]
 [1678 24364 650 1938]

[2134 421 21169 5204]
[1623 611 2630 22564]]

Window size=2

Train Accuracy= 83.2636578608%
Test Set: 81.42416666666666%
Accuracy:
Test Set:
Accuracy: 0.8142416666666666
F1 Score: 0.8152065635158733
Precision: 0.8199399851780164
Recall: 0.8142416666666666
Confusion Matrix:
[[24575 1173 1692 2560]
[1591 25721 750 1938]
[2255 372 22169 5204]
[1515 611 2630 25244]]

Window size=3

Train Accuracy= 84.4585985798%
Test Set: 82.34748738735%
Accuracy:
Test Set:
Accuracy: 0.8234748738735
F1 Score: 0.82655456158778
Precision: 0.8298016434556
Recall: 0.824456134747
Confusion Matrix:
[[26876 1056 1545 2345]
[1432 27252 675 1938]
[2091 267 23678 4809]
[1481 502 2630 26678]]

2. SKIP-GRAM with Negative Sampling Word Embedding

To train and Test Accuracy for SVD

Hyperparameters

Embedding dim=150

Batch Size=300

Hidden size=300

No of hidden layers=2

Epoch=5

Window size=1

No of Negative Sample=2

Train Accuracy= 80.3174566778%
Test Set:79.00526313456%

Accuracy:
F1 Score: 0.7900245788733
Precision: 0.7964569087677
Recall: 0.789346784355
Confusion Matrix:
[[25622 1073 1492 2456]
[1591 25721 699 1938]
[2255 372 22169 5204]
[1452 578 2630 25244]]

Window size=2
No of Negative Sample=3
Train Accuracy= 84.7845678608%
Test Set: 82.7389436785%

Test Set:
Accuracy: 0.827389436785
F1 Score: 0.8252065635158733
Precision: 0.8299399851780164
Recall: 0.8242416666666666
Confusion Matrix:
[[25175 1173 1692 2560]
 [1341 25721 667 1938]
 [2143 372 23169 5204]
 [1632 356 2087 22354]]

Window size=3
No of Negative Sample=4
Train Accuracy= 87.520656351587%
Test Set: 83.424157897743%
Accuracy:
Test Set:
Accuracy: 0. 83424157897743
F1 Score: 0.834572578433
Precision: 0.82978054854164
Recall: 0.834689904332
Confusion Matrix:
[[26075 1034 1467 2310]
 [1591 27311 654 1821]

[2189 305 23901 5204]
[1533 511 2407 25378]]

Hyperparameters used to train the model(s):

- **SVD Word Vectors:** Embedding dim=150, Batch Size=300, Hidden size=300, No of hidden layers=2, Epoch=5, Window size={1, 2, 3}.
- **Skip-gram with Negative Sampling Word Embedding:** Embedding dim=150, Batch Size=300, Hidden size=300, No of hidden layers=2, Epoch=5, Window size={1, 2, 3}, No of Negative Sample={2, 3, 4}.

Analysis of the results:

- **SVD Word Vectors:** SVD performed reasonably well, with increasing accuracy as the window size increased. However, it was outperformed by Skip-gram with Negative Sampling in all configurations.
- **Skip-gram with Negative Sampling Word Embedding:** Skip-gram consistently outperformed SVD across all window sizes and negative sample configurations, indicating its effectiveness in capturing word semantics.

Comparison of the two-word vectorizing methods:

Skip-gram with Negative Sampling generally performs better than SVD due to its ability to capture more nuanced relationships between words by considering the context in which they appear. SVD, on the other hand, relies solely on co-occurrence counts, which may not capture the full semantic meaning of words.

Shortcomings of both techniques:

- **SVD:** SVD suffers from the sparsity of the co-occurrence matrix, leading to suboptimal word embeddings, especially for rare words. It also requires a large amount of memory and computational resources.
- **Skip-gram with Negative Sampling:** While Skip-gram is more effective than SVD, it still requires a large corpus to learn meaningful embeddings. It is also computationally expensive due to the negative sampling process.

Experiment with context window sizes:

- We experimented with context window sizes of 1, 2, and 3 to capture different levels of word context. A larger window size allows the model to capture more contextual information but may also introduce noise.
- In our experiments, a window size of 3 yielded the best performance for both SVD and Skip-gram, likely because it strikes a balance between capturing enough context and avoiding noise.

- Skip-gram with Negative Sampling consistently outperformed SVD in terms of accuracy across all window sizes and negative sample configurations.
- Increasing the window size generally improved the accuracy of both SVD and Skip-gram models, as it allows the model to capture more contextual information.
- The number of negative samples had a less consistent impact on the performance, with varying effects depending on the window size and dataset characteristics.

Method	Window Size	Negative Sample	Train Accuracy	Test Accuracy	F1 Score	Precision	Recall
SVD Word Vectors	1	N/A	78.33%	77.72%	0.7752	0.7799	0.7742
SVD Word Vectors	2	N/A	83.26%	81.42%	0.8152	0.8199	0.8142
SVD Word Vectors	3	N/A	84.46%	82.35%	0.8266	0.8298	0.8245
Skip-gram with Negative Sampling	1	2	80.32%	79.01%	0.7900	0.7965	0.7893
Skip-gram with Negative Sampling	2	3	84.78%	82.74%	0.8252	0.8299	0.8242
Skip-gram with Negative Sampling	3	4	87.52%	83.42%	0.8346	0.8298	0.8347