Intro to Natural Language Processing
Assignment-3

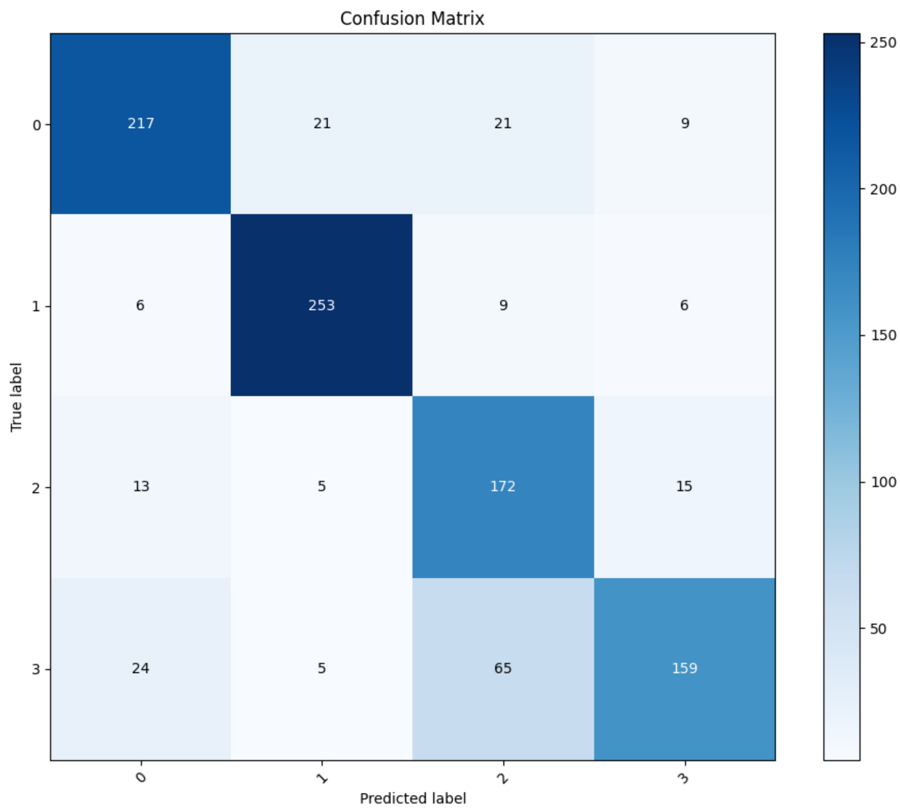
Name: Saketh Reddy Vemula
Roll No: 2022114014

5. **Analysis**

 # Compare and analyze which of the two-word vectorizing methods performs better by using performance metrics such as accuracy, F1 score, precision, recall, and the confusion matrix on both the train and test sets.

1. SVD word vectors on test sets:
   Parameters:
   BATCH_SIZE = 64, HIDDEN_DIM = 128, EMBEDDING_DIM = 100, LEARNING_RATE = 0.001, NUM_EPOCHS = 20, CONTEXT WINDOW = 4
   Training instances: top 10000 sentences from train sets
   Testing instances: top 1000 sentences from test sets

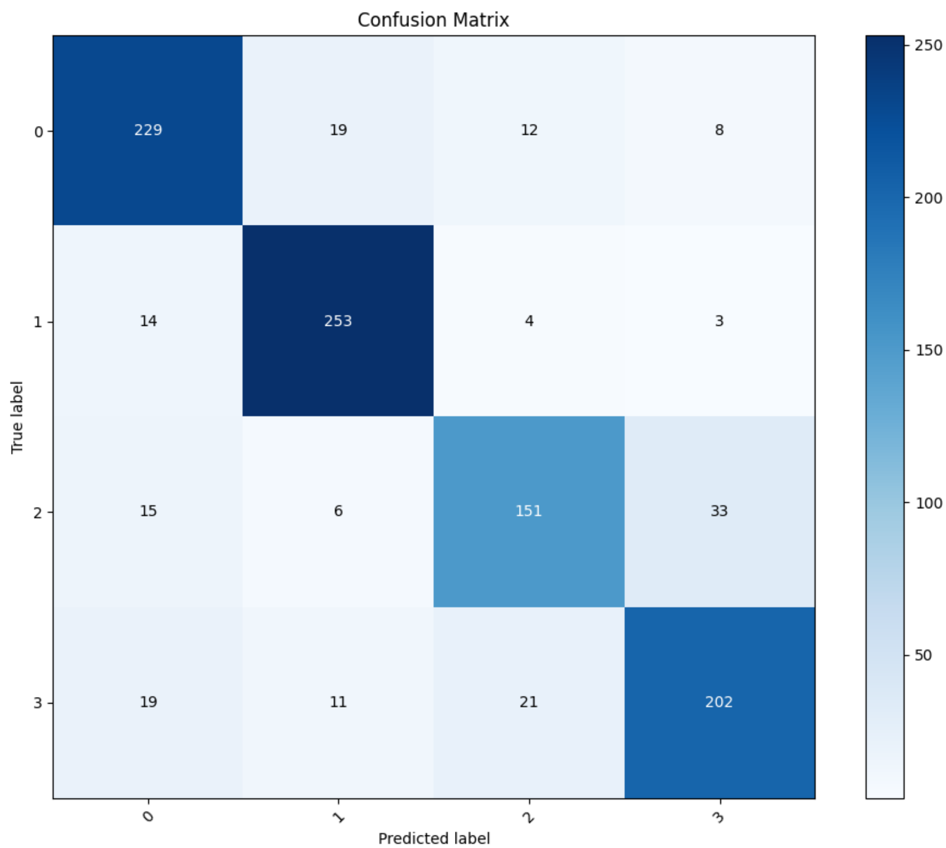
Accuracy: 0.8010
F1 Score: 0.7943
Precision: 0.8027
Recall: 0.8001



2. Skip Gram word vectors on test sets:
   Parameters:
   BATCH_SIZE = 64, HIDDEN_DIM = 128, EMBEDDING_DIM = 100, LEARNING_RATE = 0.001, NUM_EPOCHS = 20, CONTEXT_WINDOW = 4

Training instances: top 10000 sentences from train sets
Testing instances: top 1000 sentences from test sets

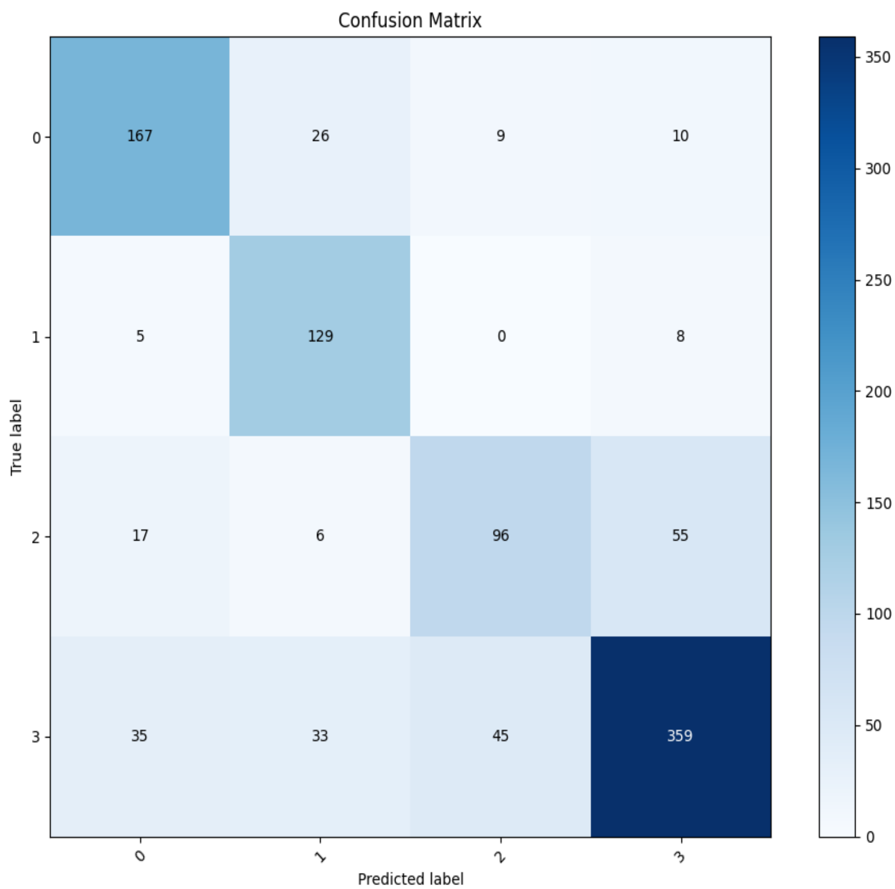
Accuracy: 0.8480
F1 Score: 0.8422
Precision: 0.8421
Recall: 0.8425



3. SVD word vectors on train sets:
   Parameters:
   BATCH_SIZE = 64, HIDDEN_DIM = 128, EMBEDDING_DIM = 100, LEARNING_RATE = 0.001, NUM_EPOCHS = 20

   Training instances: top 10000 sentences from train sets
   Testing instances: top 1000 sentences from train sets

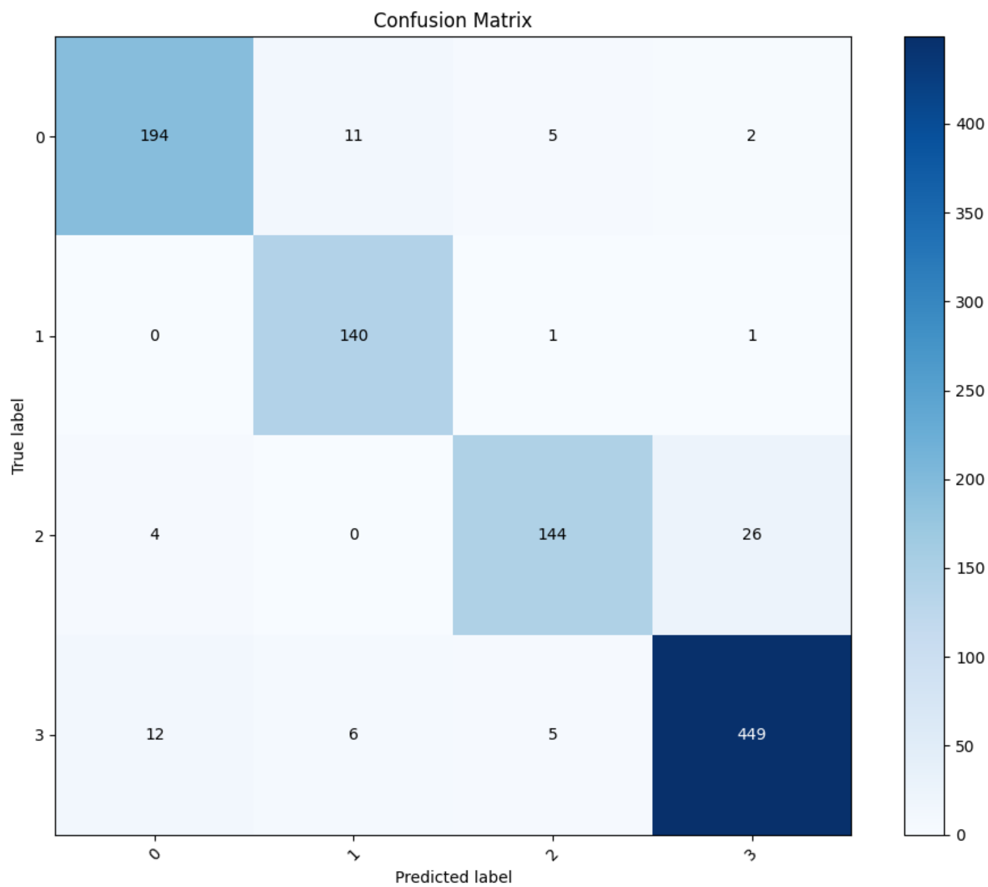
   Accuracy: 0.7510
   F1 Score: 0.7302
   Precision: 0.7204
   Recall: 0.7521



4. Skip gram word vectors on train sets:
   Parameters:
   BATCH_SIZE = 64, HIDDEN_DIM = 128, EMBEDDING_DIM = 100, LEARNING_RATE = 0.001, NUM_EPOCHS = 20

   Training instances: top 10000 sentences from train sets
   Testing instances: top 1000 sentences from train sets

   Accuracy: 0.9270
   F1 Score: 0.9191
   Precision: 0.9210
   Recall: 0.9200



Conclusion:

1. Skip gram vectors outperforms the SVD model on all evaluation metrics for the test sets and train sets.
2. This suggest that the skip-gram word embeddings are better at capturing the semantic and syntactic relationships in the test data compared to the SVD word embeddings.
3. Accuracy increases drastically in the case of skip gram for training sets evaluation. But it surprisingly decreases for SVD.
4. Accuracies of word vectors from skip gram with negative sampling are higher in both of the cases. This indicates that, Skip-gram model is more robust and generalizes better to unseen data compared to the SVD model.
5. The Skip-gram model achieves an accuracy of 0.8480, F1-score of 0.8422, precision of 0.8421, and recall of 0.8425, which are higher than the corresponding metrics for the SVD model.

 # Write a detailed report on why one technique might perform better than the other. Also, include the possible shortcomings of both techniques

Skip gram with negative sampling outperforms SVD embeddings:

1. Capturing word contexts:
   Skip gram with negative sampling is designed to learn the relationship between context and target words. Thus, directly captures the semantic and syntactic relationships between words. In contrast, SVD vectors are derived from a word co-occurrence matrix, which may not capture the contextual information as effectively.
2. Negative sampling:
   It improves the embeddings in the case of skip gram model. There is nothing like this in SVD. Skip gram model learns to differentiate between plausible and implausible word pairs. Discrimination increases leading to better accuracy in skip gram.
3. Handling rare word:
   In SVD, rare words may not have enough co-occurrences to accurately represent the meaning, leading to poor embeddings. Since skip grams deals with stochasticity, it can handle rare words more efficiently than SVD.
4. Corpus size:
   Skip gram works better as compared to SVD on Large corpus, higher dimensions although it is slower.
5. More Entropy:
   Since skip gram introduces stochasticity, it is more driven to creativity. Hence, entropy is higher.

Possible shortcomings of SVD:

1. Capturing word order:
   Their is no scope of handling the word order.
2. Handling words with multiple meanings:
   Both SVD and skip gram model fail to this.
3. Computationally expensive:
   Requires more memory to store large matrices and computationally expensive to update and perform operations on them.
4. Inefficient use of statistics.
5. Cannot capture word similarities.
6. Disproportionate importance given to large counts.

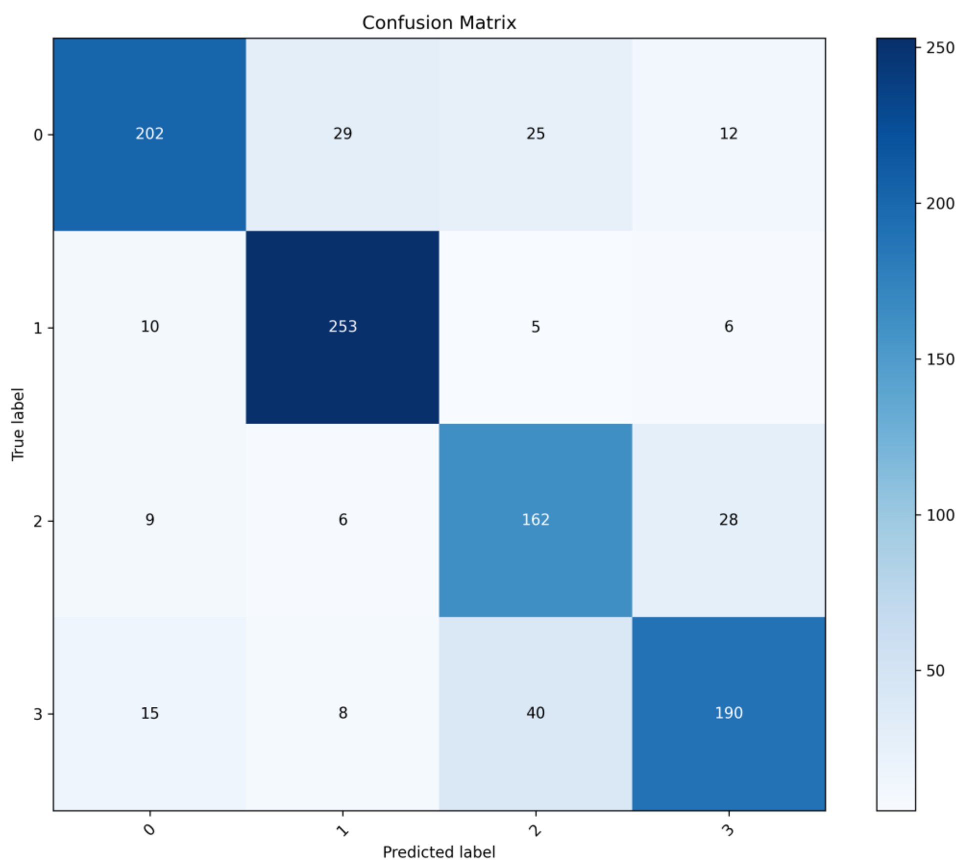
Possible shortcomings of Skip gram:

1. Fails to explicitly capture the positional information of words in word embeddings.
2. Fails to capture polysemy words.
3. Slow training. Requires more time to train the model. Hence, it can be expensive.
4. Requires more computation power and storage to achieve commendable accuracies.
5. Inefficient use of statistics.
6. Slow training.

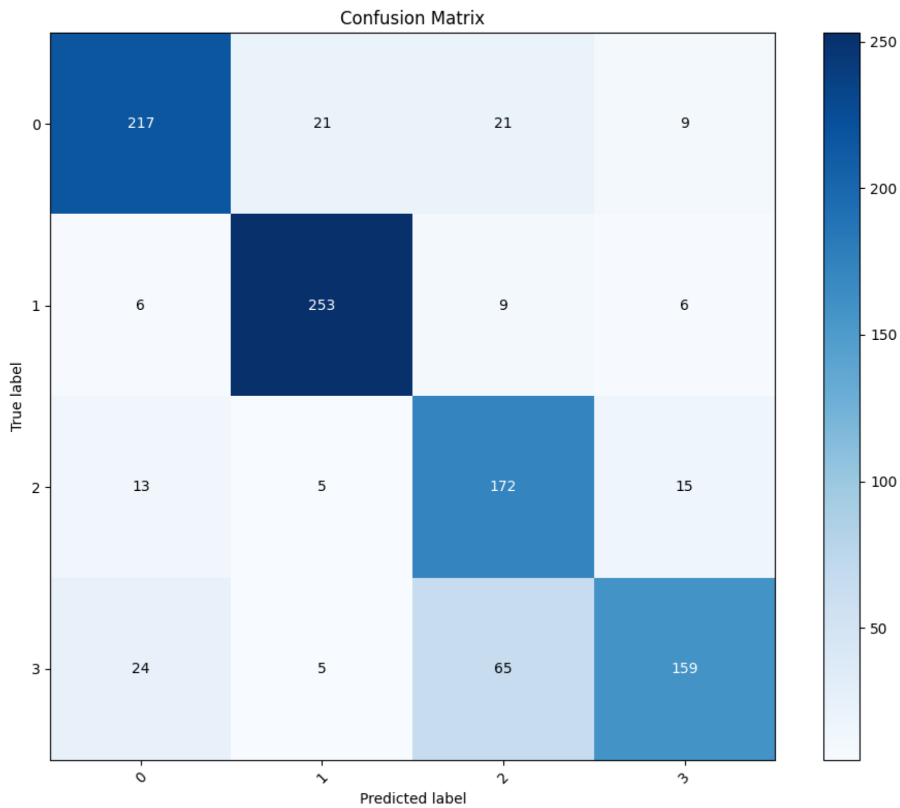
6. **Hyperparameter tuning:**

<u>SVD word vectors:</u>

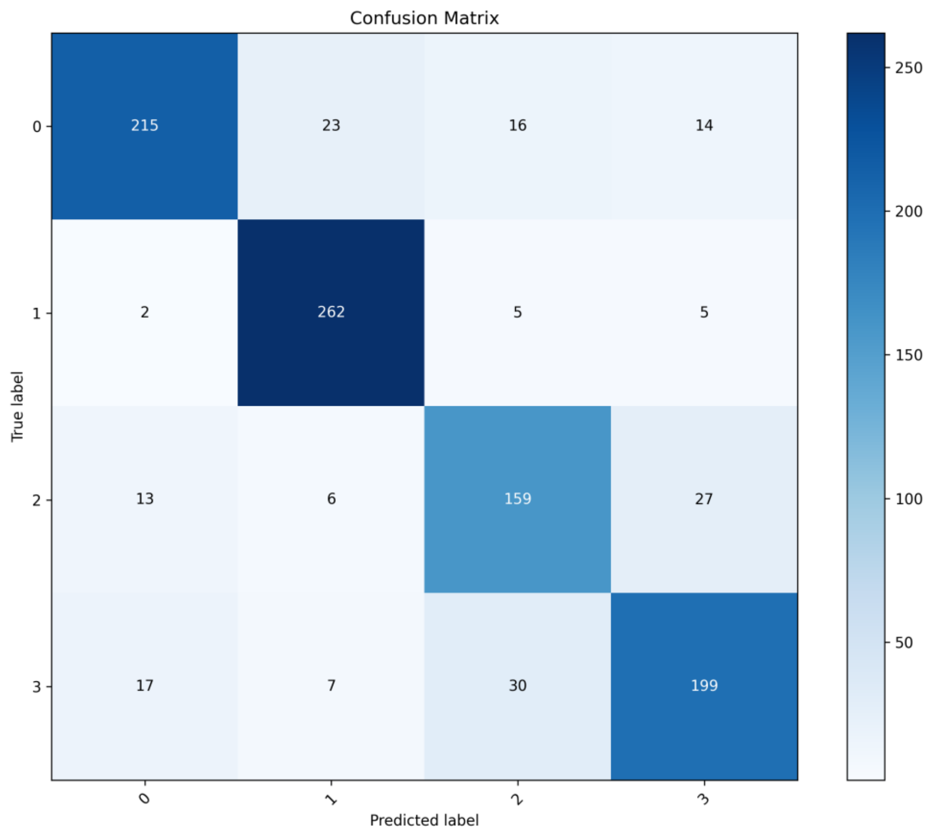
1. Context Window = 2
   Accuracy: 0.8070
   F1 Score: 0.8020
   Precision: 0.8035
   Recall: 0.8046



2. Context Window = 4
   Accuracy: 0.8010
   F1 Score: 0.7943
   Precision: 0.8027
   Recall: 0.8001

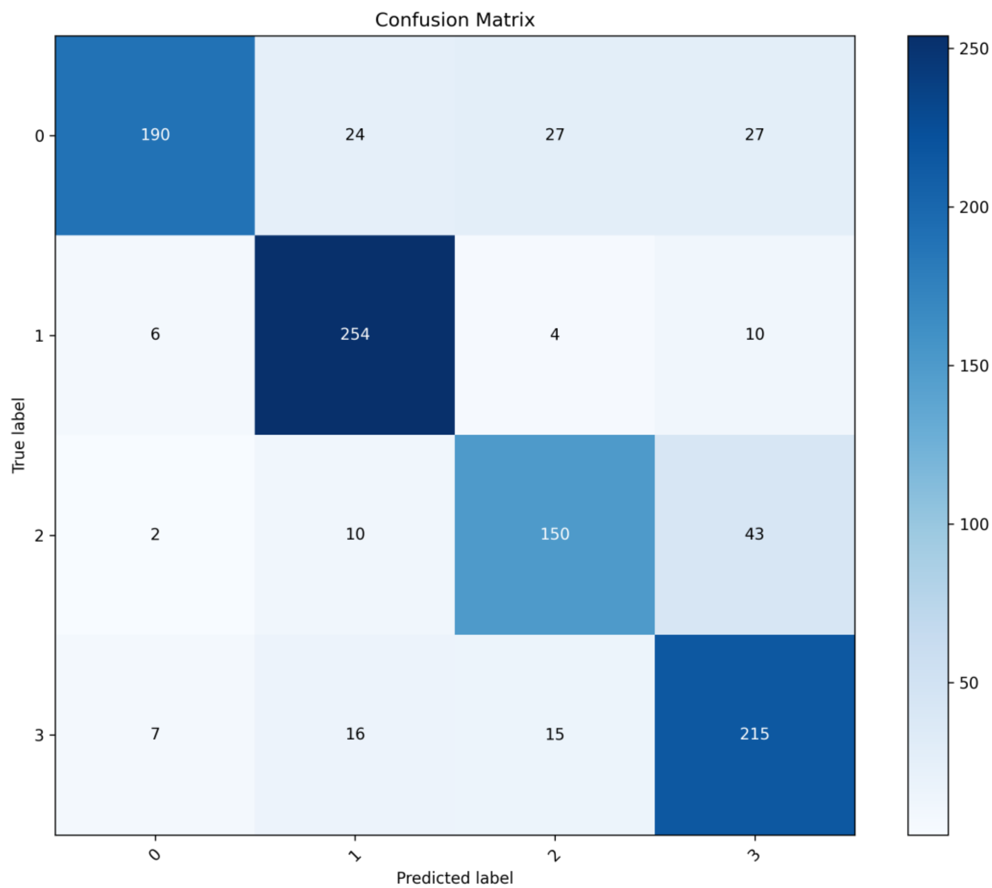


3. Context Window = 6:
   Accuracy: 0.8350
   F1 Score: 0.8291
   Precision: 0.8298
   Recall: 0.8302



Skip gram word vectors:

1. Context Window = 2:
   Accuracy: 0.8090
   F1 Score: 0.8038
   Precision: 0.8141
   Recall: 0.8044



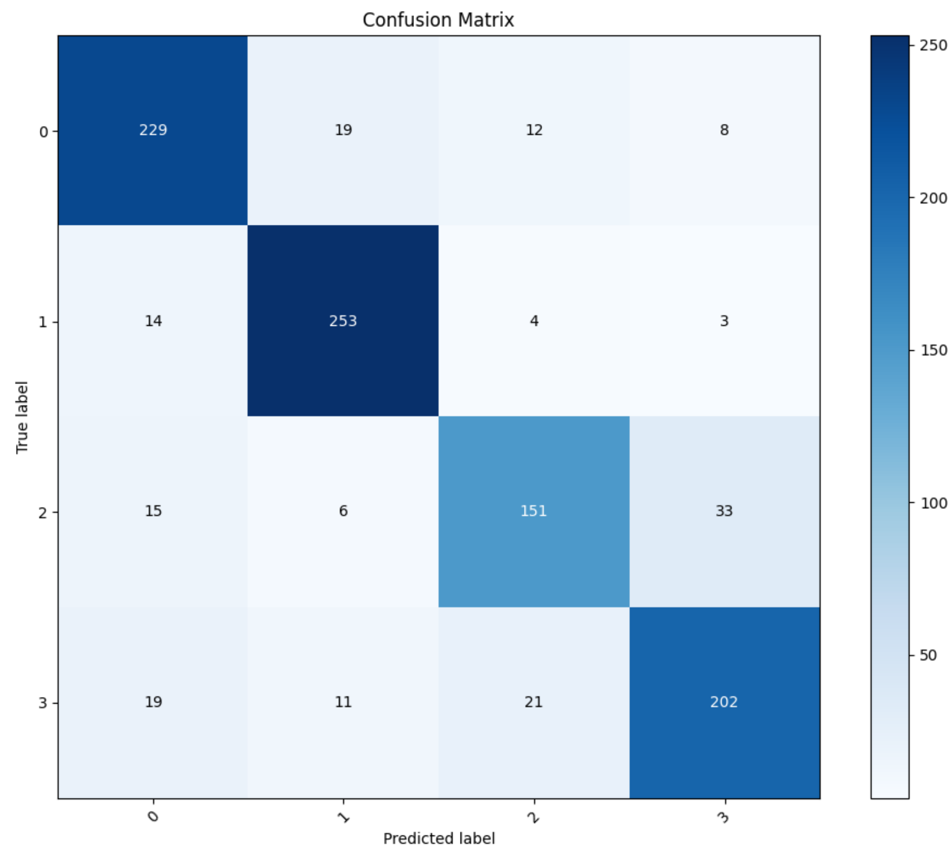
2. Context Window = 4:

Accuracy: 0.8480

F1 Score: 0.8422

Precision: 0.8421

Recall: 0.8425



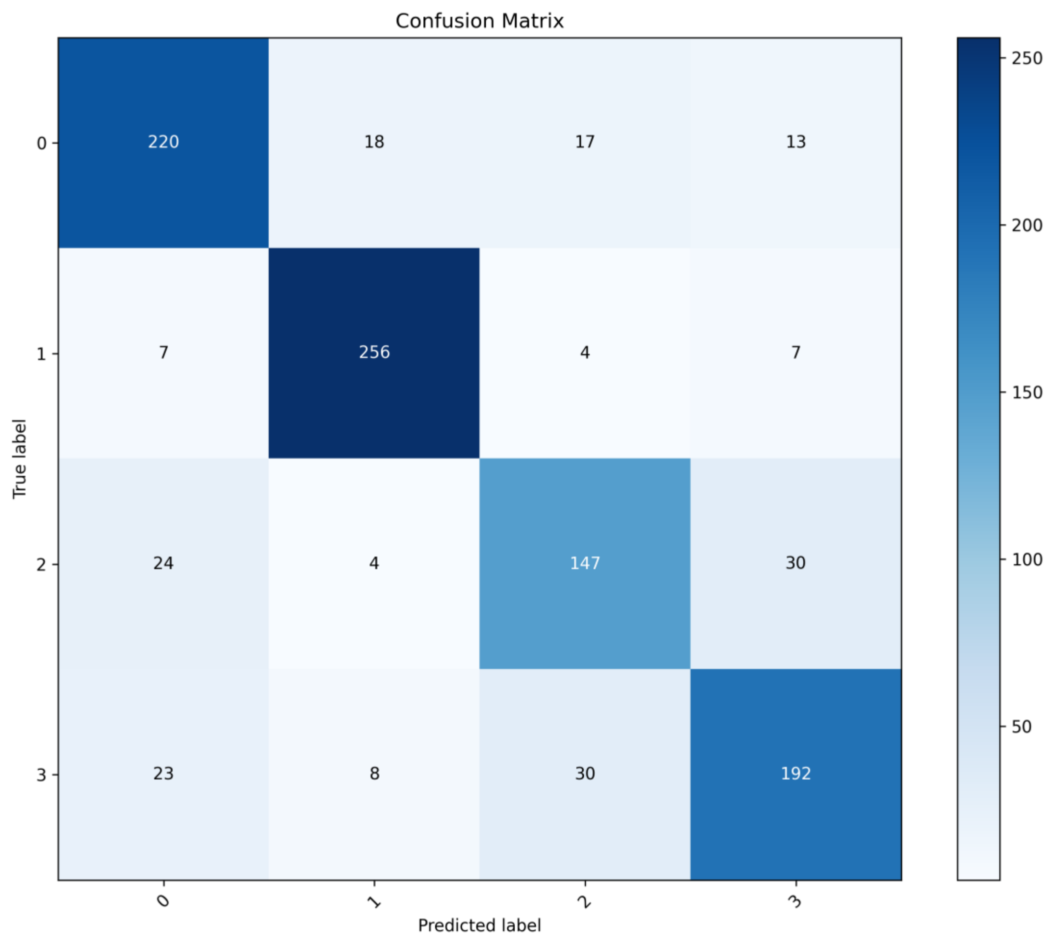
3. Context Window = 6:

Accuracy: 0.8150

F1 Score: 0.8078

Precision: 0.8085

Recall: 0.8078



Reason for these specific context window sizes:

1. Context <= 2 captures syntactic information whereas context window <= 5 and > 2 captures semantic information.
2. They are neither too big nor too small.
3. Can show evidence for tradeoff between Syntactic and Semantic knowledge, which gives the best results.

For SVD:
Context window of size = 6 gave the best result with F1 score of approximately 83%

For Skip gram with negative sampling:
Context window of size = 4 gave the best result with F1 score of approximately 84%

Observations:

1. This shows that the tradeoff of syntactic and semantic knowledge occurs before for Skip gram word vector model than SVD model.
2. SVD tend to capture more semantic relationships between words, which may benefit from a large context window that encompasses broader linguistic patterns.
3. Skip gram with negative sampling is better at capturing the perfect balance between syntactic information and semantic information.
4. Since SVD uses dimensionality reduction techniques, it may require larger context window size. Whereas Skip gram directly learns the embeddings layer from the training data, hence, a smaller window size may be sufficient to capture the most relevant local contexts.