Report

I have used 20k sentences from the training dataset to create word vectors using both Singular Value Decomposition (SVD) and Skip-gram models, with an embedding dimension of 300 for each. Additionally, for the Skip-gram with Negative Sampling (SGNS) model, I have set the number of negative samples per positive sample to 5 (k=5).

For the downstream task, I have set certain hyperparameters as constants or fixed values to maintain consistency and ensure reproducibility.

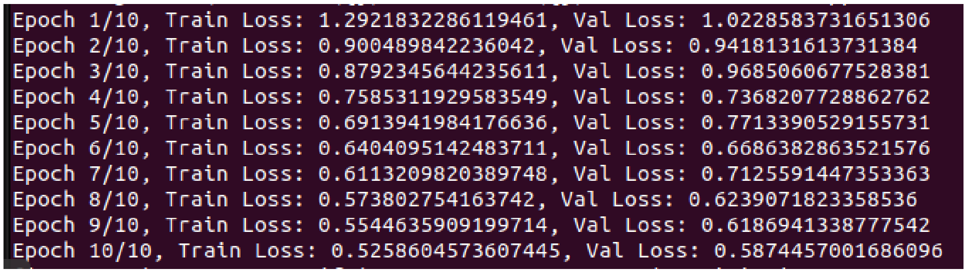
```
input_dim = 300
hidden_dim = 128
output_dim = n_classes
n_layers = 2
bidirectional = True
n_epochs = 10
lr = 0.001
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

Hyperparamter tuning:

I trained the model using different values for the context window, specifically {1, 2, 3}, to analyze how the model's accuracies change based on the context window size.

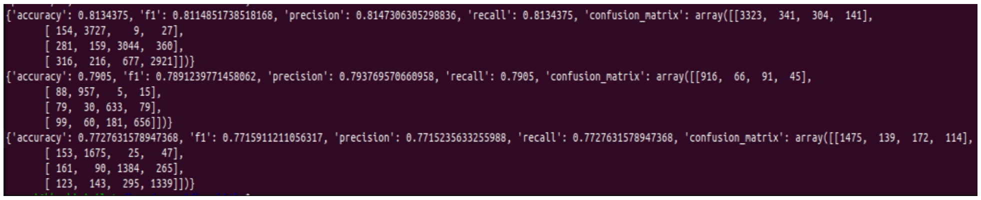
SVD

Context_window = 1



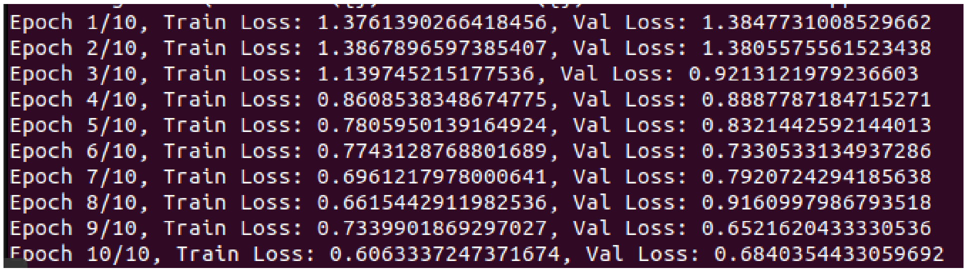
```
Epoch 1/10, Train Loss: 1.2921832286119461, Val Loss: 1.0228583731651306
Epoch 2/10, Train Loss: 0.900489842236042, Val Loss: 0.9418131613731384
Epoch 3/10, Train Loss: 0.8792345644235611, Val Loss: 0.9685060677528381
Epoch 4/10, Train Loss: 0.7585311929583549, Val Loss: 0.7368207728862762
Epoch 5/10, Train Loss: 0.6913941984176636, Val Loss: 0.7713390529155731
Epoch 6/10, Train Loss: 0.6404095142483711, Val Loss: 0.6686382863521576
Epoch 7/10, Train Loss: 0.6113209820389748, Val Loss: 0.7125591447353363
Epoch 8/10, Train Loss: 0.573802754163742, Val Loss: 0.6239071823358536
Epoch 9/10, Train Loss: 0.5544635909199714, Val Loss: 0.6186941338777542
Epoch 10/10, Train Loss: 0.5258604573607445, Val Loss: 0.5874457001686096
```

Metrics on Train, Validation and test set respectively



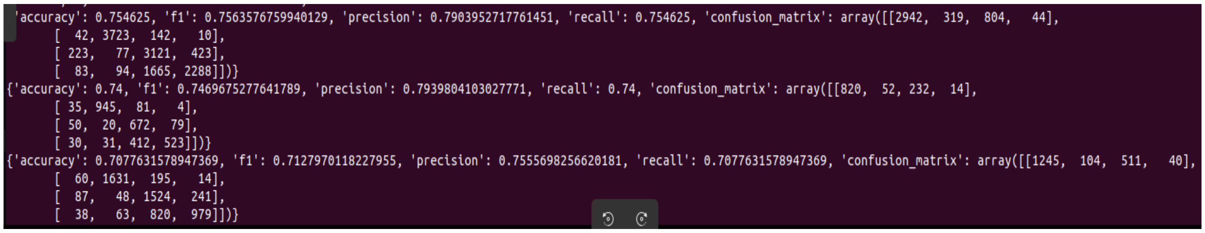
```
{'accuracy': 0.8134375, 'f1': 0.8114851738518168, 'precision': 0.8147306305298836, 'recall': 0.8134375, 'confusion_matrix': array([[3323,  341,  304,  141],
       [ 154, 3727,    9,   27],
       [ 281,  159, 3044,  360],
       [ 316,  216,  677, 2921]])}
{'accuracy': 0.7905, 'f1': 0.7891239771458062, 'precision': 0.793769570660958, 'recall': 0.7905, 'confusion_matrix': array([[916,  66,  91,  45],
       [ 88, 957,   5,  15],
       [ 79,  30, 633,  79],
       [ 99,  60, 181, 656]])}
{'accuracy': 0.7727631578947368, 'f1': 0.7715911211056317, 'precision': 0.7715235633255988, 'recall': 0.7727631578947368, 'confusion_matrix': array([[1475,  139,  172,  114],
       [ 153, 1675,   25,   47],
       [ 161,   90, 1384,  265],
       [ 123,  143,  295, 1339]])}
```

Context_window = 2



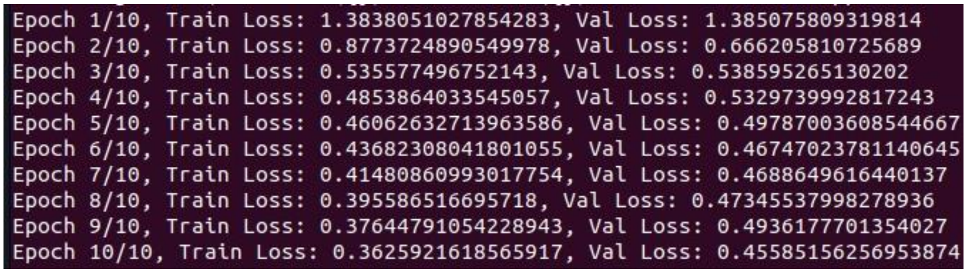
```
Epoch 1/10, Train Loss: 1.3761390266418456, Val Loss: 1.3847731008529662
Epoch 2/10, Train Loss: 1.3867896597385407, Val Loss: 1.3805575561523438
Epoch 3/10, Train Loss: 1.139745215177536, Val Loss: 0.9213121979236603
Epoch 4/10, Train Loss: 0.8608538348674775, Val Loss: 0.8887787184715271
Epoch 5/10, Train Loss: 0.7805950139164924, Val Loss: 0.8321442592144013
Epoch 6/10, Train Loss: 0.7743128768801689, Val Loss: 0.7330533134937286
Epoch 7/10, Train Loss: 0.6961217978000641, Val Loss: 0.7920724294185638
Epoch 8/10, Train Loss: 0.6615442911982536, Val Loss: 0.9160997986793518
Epoch 9/10, Train Loss: 0.7339901869297027, Val Loss: 0.6521620433330536
Epoch 10/10, Train Loss: 0.6063337247371674, Val Loss: 0.6840354433059692
```

Metrics on Train, Validation and test set respectively



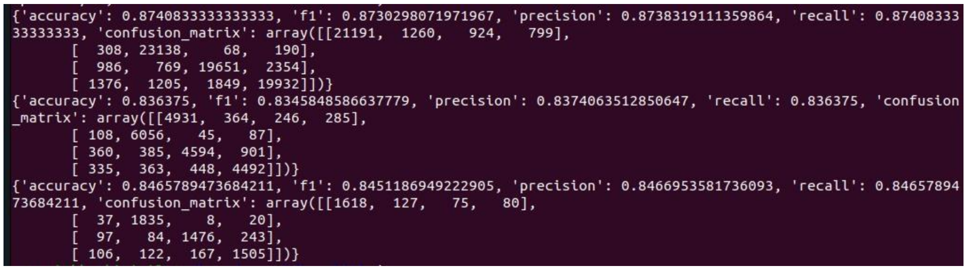
```
'accuracy': 0.754625, 'f1': 0.7563576759940129, 'precision': 0.7903952717761451, 'recall': 0.754625, 'confusion_matrix': array([[2942,  319,  804,   44],
       [  42, 3723,  142,   10],
       [ 223,   77, 3121,  423],
       [  83,   94, 1665, 2288]])}
{'accuracy': 0.74, 'f1': 0.7469675277641789, 'precision': 0.7939804103027771, 'recall': 0.74, 'confusion_matrix': array([[820,  52, 232,  14],
       [ 35, 945,  81,   4],
       [ 50,  20, 672,  79],
       [ 30,  31, 412, 523]])}
{'accuracy': 0.7077631578947369, 'f1': 0.7127970118227955, 'precision': 0.7555698256620181, 'recall': 0.7077631578947369, 'confusion_matrix': array([[1245,  104,  511,   40],
       [  60, 1631,  195,   14],
       [  87,   48, 1524,  241],
       [  38,   63,  820,  979]])}
```

Context_window=3



```
Epoch 1/10, Train Loss: 1.3838051027854283, Val Loss: 1.385075809319814
Epoch 2/10, Train Loss: 0.8773724890549978, Val Loss: 0.666205810725689
Epoch 3/10, Train Loss: 0.535577496752143, Val Loss: 0.538595265130202
Epoch 4/10, Train Loss: 0.4853864033545057, Val Loss: 0.5329739992817243
Epoch 5/10, Train Loss: 0.46062632713963586, Val Loss: 0.49787003608544667
Epoch 6/10, Train Loss: 0.43682308041801055, Val Loss: 0.46747023781140645
Epoch 7/10, Train Loss: 0.41480860993017754, Val Loss: 0.4688649616440137
Epoch 8/10, Train Loss: 0.395586516695718, Val Loss: 0.47345537998278936
Epoch 9/10, Train Loss: 0.37644791054228943, Val Loss: 0.4936177701354027
Epoch 10/10, Train Loss: 0.3625921618565917, Val Loss: 0.45585156256953874
```

Metrics on Train, Validation and test set respectively

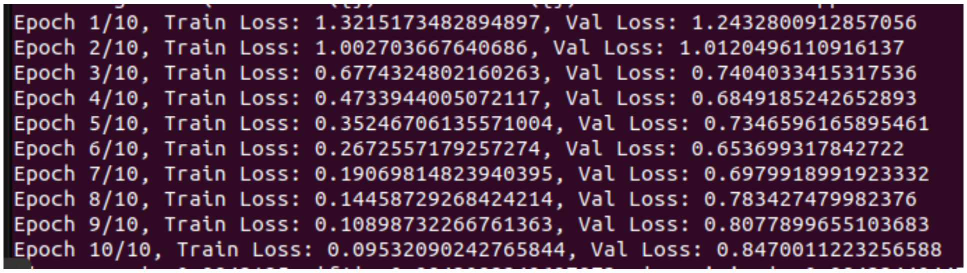


```
{'accuracy': 0.8740833333333333, 'f1': 0.8730298071971967, 'precision': 0.8738319111359864, 'recall': 0.87408333
33333333, 'confusion_matrix': array([[21191,  1260,   924,   799],
       [  308, 23138,    68,   190],
       [  986,   769, 19651,  2354],
       [ 1376,  1205,  1849, 19932]])}
{'accuracy': 0.836375, 'f1': 0.8345848586637779, 'precision': 0.8374063512850647, 'recall': 0.836375, 'confusion
_matrix': array([[4931,  364,  246,  285],
       [ 108, 6056,   45,   87],
       [ 360,  385, 4594,  901],
       [ 335,  363,  448, 4492]])}
{'accuracy': 0.8465789473684211, 'f1': 0.8451186949222905, 'precision': 0.8466953581736093, 'recall': 0.84657894
73684211, 'confusion_matrix': array([[1618,  127,   75,   80],
       [  37, 1835,    8,   20],
       [  97,   84, 1476,  243],
       [ 106,  122,  167, 1505]])}
```

As the size of the context window expands, we observe a corresponding rise in accuracy on the test set. This improvement is attributed to the larger window size offering a broader context, which aids in capturing extensive dependencies across the data.

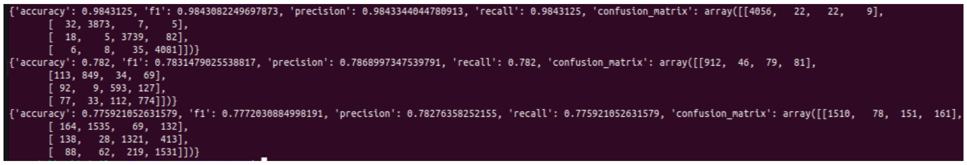
SKIP GRAM

Context_window=1



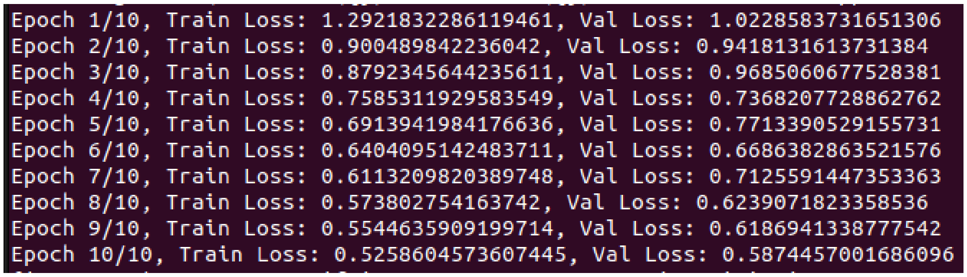
```
Epoch 1/10, Train Loss: 1.3215173482894897, Val Loss: 1.2432800912857056
Epoch 2/10, Train Loss: 1.002703667640686, Val Loss: 1.0120496110916137
Epoch 3/10, Train Loss: 0.6774324802160263, Val Loss: 0.7404033415317536
Epoch 4/10, Train Loss: 0.4733944005072117, Val Loss: 0.6849185242652893
Epoch 5/10, Train Loss: 0.35246706135571004, Val Loss: 0.7346596165895461
Epoch 6/10, Train Loss: 0.2672557179257274, Val Loss: 0.653699317842722
Epoch 7/10, Train Loss: 0.19069814823940395, Val Loss: 0.6979918991923332
Epoch 8/10, Train Loss: 0.14458729268424214, Val Loss: 0.783427479982376
Epoch 9/10, Train Loss: 0.10898732266761363, Val Loss: 0.8077899655103683
Epoch 10/10, Train Loss: 0.09532090242765844, Val Loss: 0.8470011223256588
```

Metrics on Train, Validation and test set respectively



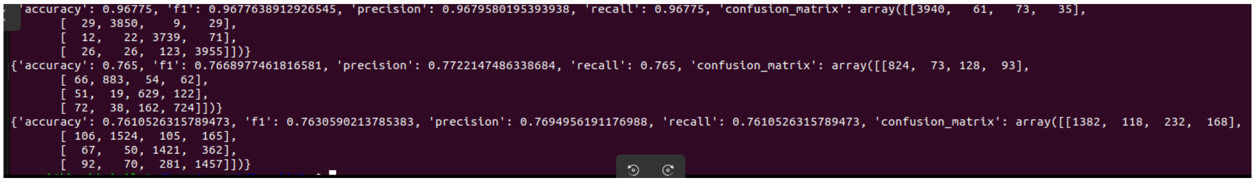
```
{'accuracy': 0.9843125, 'f1': 0.9843082249697873, 'precision': 0.9843344044780913, 'recall': 0.9843125, 'confusion_matrix': array([[4056,   22,   22,    9],
       [  32, 3873,    7,    5],
       [  18,    5, 3739,   82],
       [   6,    8,   35, 4081]])}
{'accuracy': 0.782, 'f1': 0.7831479025538817, 'precision': 0.7868997347539791, 'recall': 0.782, 'confusion_matrix': array([[912,  46,  79,  81],
       [113, 849,  34,  69],
       [ 92,   9, 593, 127],
       [ 77,  33, 112, 774]])}
{'accuracy': 0.775921052631579, 'f1': 0.7772030884998191, 'precision': 0.78276358252155, 'recall': 0.775921052631579, 'confusion_matrix': array([[1510,   78,  151,  161],
       [ 164, 1535,   69,  132],
       [ 138,   28, 1321,  413],
       [  88,   62,  219, 1531]])}
```

Context_Window = 2



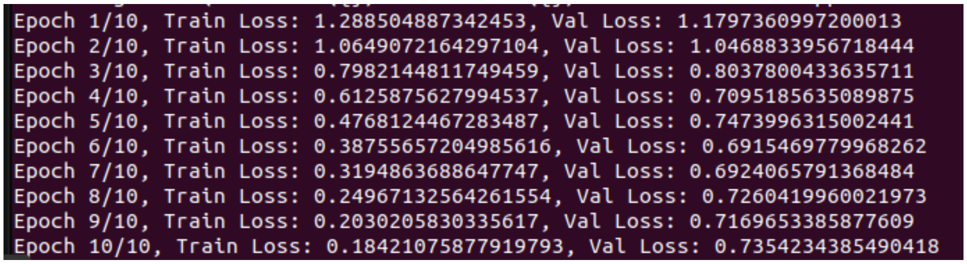
```
Epoch 1/10, Train Loss: 1.2921832286119461, Val Loss: 1.0228583731651306
Epoch 2/10, Train Loss: 0.900489842236042, Val Loss: 0.9418131613731384
Epoch 3/10, Train Loss: 0.8792345644235611, Val Loss: 0.9685060677528381
Epoch 4/10, Train Loss: 0.7585311929583549, Val Loss: 0.7368207728862762
Epoch 5/10, Train Loss: 0.6913941984176636, Val Loss: 0.7713390529155731
Epoch 6/10, Train Loss: 0.6404095142483711, Val Loss: 0.6686382863521576
Epoch 7/10, Train Loss: 0.6113209820389748, Val Loss: 0.7125591447353363
Epoch 8/10, Train Loss: 0.573802754163742, Val Loss: 0.6239071823358536
Epoch 9/10, Train Loss: 0.5544635909199714, Val Loss: 0.6186941338777542
Epoch 10/10, Train Loss: 0.5258604573607445, Val Loss: 0.5874457001686096
```

Metrics on Train, Validation and test set respectively



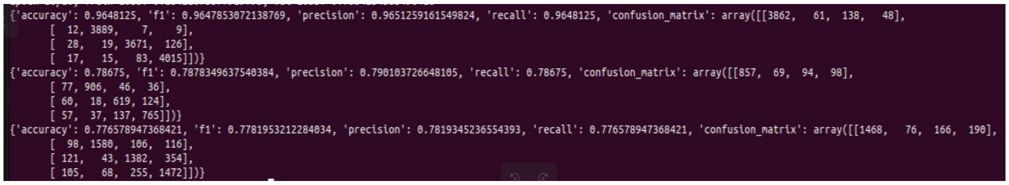
```
'accuracy': 0.96775, 'f1': 0.9677638912926545, 'precision': 0.9679580195393938, 'recall': 0.96775, 'confusion_matrix': array([[3940,   61,   73,   35],
       [  29, 3850,    9,   29],
       [  12,   22, 3739,   71],
       [  26,   26,  123, 3955]])}
{'accuracy': 0.765, 'f1': 0.7668977461816581, 'precision': 0.7722147486338684, 'recall': 0.765, 'confusion_matrix': array([[824,  73, 128,  93],
       [ 66, 883,  54,  62],
       [ 51,  19, 629, 122],
       [ 72,  38, 162, 724]])}
{'accuracy': 0.7610526315789473, 'f1': 0.7630590213785383, 'precision': 0.7694956191176988, 'recall': 0.7610526315789473, 'confusion_matrix': array([[1382,  118,  232,  168],
       [ 106, 1524,  105,  165],
       [  67,   50, 1421,  362],
       [  92,   70,  281, 1457]])}
```

Context_window = 3



```
Epoch 1/10, Train Loss: 1.288504887342453, Val Loss: 1.1797360997200013
Epoch 2/10, Train Loss: 1.0649072164297104, Val Loss: 1.0468833956718444
Epoch 3/10, Train Loss: 0.7982144811749459, Val Loss: 0.8037800433635711
Epoch 4/10, Train Loss: 0.6125875627994537, Val Loss: 0.7095185635089875
Epoch 5/10, Train Loss: 0.4768124467283487, Val Loss: 0.7473996315002441
Epoch 6/10, Train Loss: 0.38755657204985616, Val Loss: 0.6915469779968262
Epoch 7/10, Train Loss: 0.3194863688647747, Val Loss: 0.6924065791368484
Epoch 8/10, Train Loss: 0.24967132564261554, Val Loss: 0.7260419960021973
Epoch 9/10, Train Loss: 0.2030205830335617, Val Loss: 0.7169653385877609
Epoch 10/10, Train Loss: 0.18421075877919793, Val Loss: 0.7354234385490418
```

Metrics on Train, Validation and test set respectively



```
{'accuracy': 0.9648125, 'f1': 0.9647853072138769, 'precision': 0.9651259161549824, 'recall': 0.9648125, 'confusion_matrix': array([[3862,   61,  138,   48],
       [  12, 3889,    7,    9],
       [  28,   19, 3671,  126],
       [  17,   15,   83, 4015]])}
{'accuracy': 0.78675, 'f1': 0.7878349637540384, 'precision': 0.790103726648105, 'recall': 0.78675, 'confusion_matrix': array([[857,  69,  94,  98],
       [ 77, 906,  46,  36],
       [ 60,  18, 619, 124],
       [ 57,  37, 137, 765]])}
{'accuracy': 0.776578947368421, 'f1': 0.7781953212284034, 'precision': 0.7819345236554393, 'recall': 0.776578947368421, 'confusion_matrix': array([[1468,   76,  166,  190],
       [  98, 1580,  106,  116],
       [ 121,   43, 1382,  354],
       [ 105,   68,  255, 1472]])}
```

However here as context_window size changes the accuracy doesn’t seem to be affecting much. This might be since we are training the word embeddings on a small dataset, and the test accuracy is already high. Enlarging the context window in a skip-gram model may not consistently enhance accuracy, often due to increased semantic drift, data sparsity, heightened computational complexity, loss of local context, and the risk of overfitting.

CONCLUSION:

We observe that skip-gram model (with negative sampling) performs better than svd in all metrics.