Homework 7 Report

Damin Zhang

Task 1 and 2 used the *sentiment_dataset_train_40* and *sentiment_dataset_test_40*. Due to limited GPU computing ability, task 3 used *sentiment_dataset_train_3* and *sentiment_dataset_test_3*.

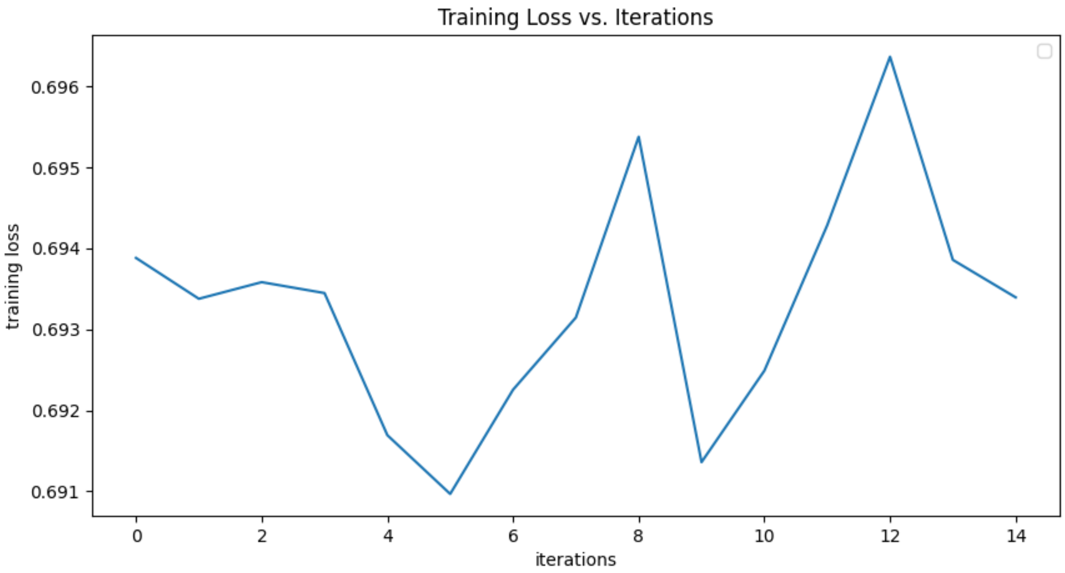
Task 1

The goal for task is seeking if there exist any alternatives to the modeling text with one-hot vectors. For this task, I chose the 2^{16} binary encoding when the vocabulary size is under 2^{16}. This encoding assigns a 16-bit code word to each word base on its index in a sorted list of the vocabulary. Since the dimension of the vector that represents a word changed from 17001 to 16, the *input_size* variable for creating *TEXTnet* model should change from *vocab_size* to *input_size=16*. Similarly, the shape of *review_tensor* also changed from *(len(review), len(self.vocab))* to *(len(review), 16)*. For code reference, check *task1.py*. Following screenshots show the results and loss plots.

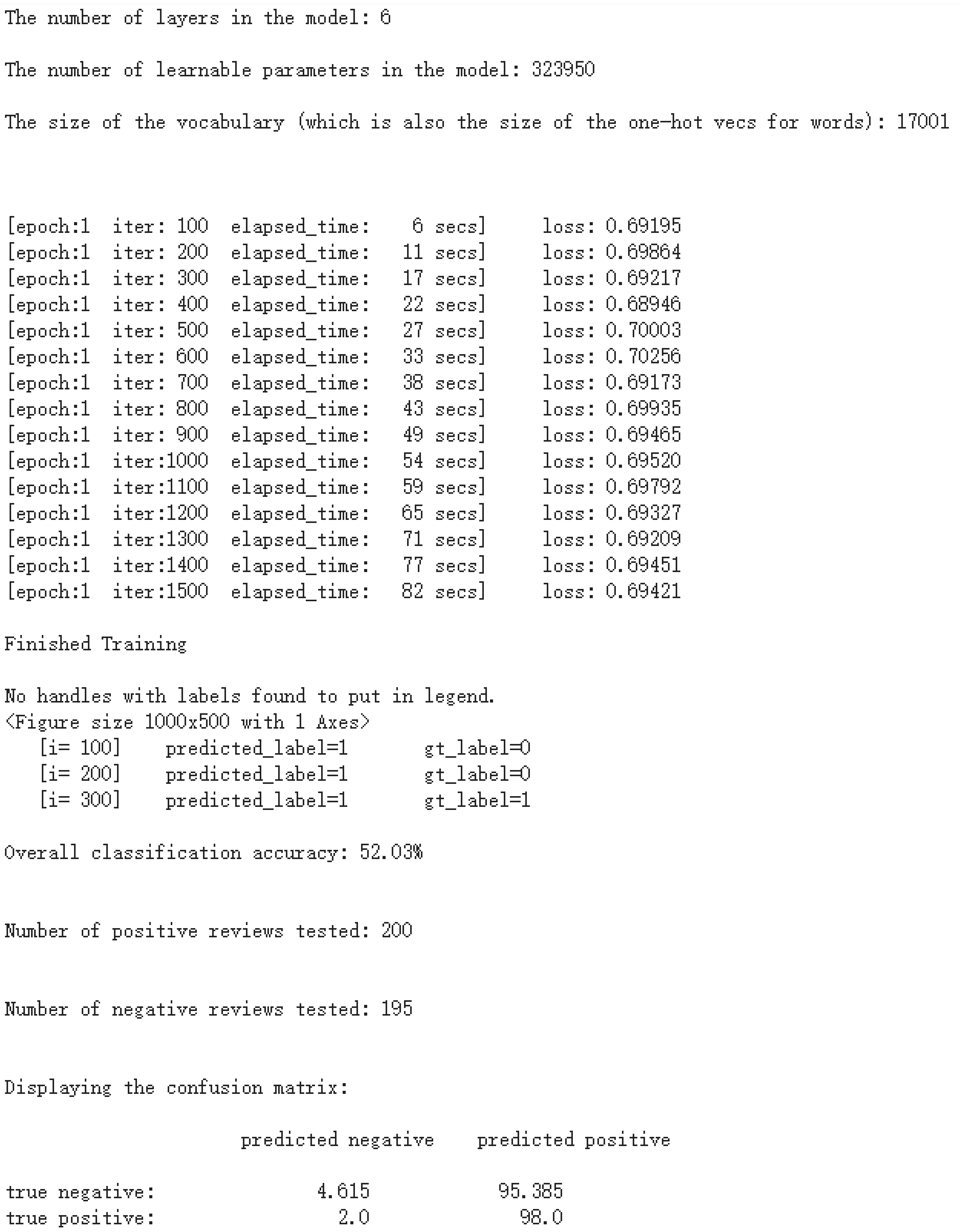
For this task, I used epoch of 1, learning rate 1e-4, momentum 0.9.

TEXTnet with one-hot encoding

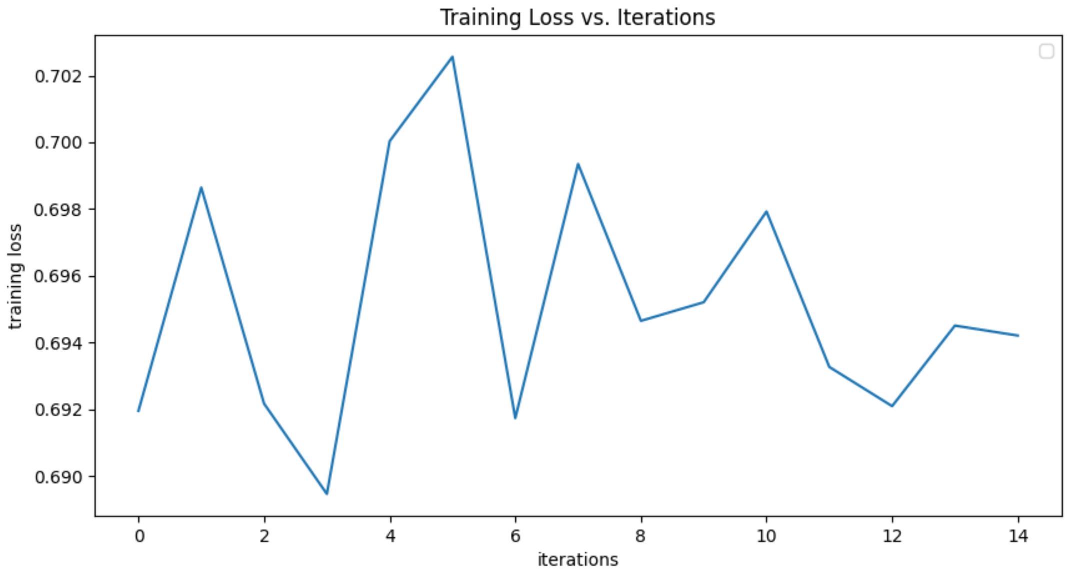
```
The number of layers in the model: 6

The number of learnable parameters in the model: 10718770

The size of the vocabulary (which is also the size of the one-hot vecs for words): 17001



[epoch:1  iter: 100  elapsed_time:   13 secs]     loss: 0.69388
[epoch:1  iter: 200  elapsed_time:   25 secs]     loss: 0.69338
[epoch:1  iter: 300  elapsed_time:   37 secs]     loss: 0.69358
[epoch:1  iter: 400  elapsed_time:   49 secs]     loss: 0.69345
[epoch:1  iter: 500  elapsed_time:   65 secs]     loss: 0.69169
[epoch:1  iter: 600  elapsed_time:   77 secs]     loss: 0.69097
[epoch:1  iter: 700  elapsed_time:   90 secs]     loss: 0.69226
[epoch:1  iter: 800  elapsed_time:  101 secs]     loss: 0.69315
[epoch:1  iter: 900  elapsed_time:  116 secs]     loss: 0.69538
[epoch:1  iter:1000  elapsed_time:  126 secs]     loss: 0.69136
[epoch:1  iter:1100  elapsed_time:  138 secs]     loss: 0.69249
[epoch:1  iter:1200  elapsed_time:  151 secs]     loss: 0.69428
[epoch:1  iter:1300  elapsed_time:  164 secs]     loss: 0.69637
[epoch:1  iter:1400  elapsed_time:  176 secs]     loss: 0.69386
[epoch:1  iter:1500  elapsed_time:  190 secs]     loss: 0.69340

Finished Training

No handles with labels found to put in legend.
<Figure size 1000x500 with 1 Axes>
   [i= 100]    predicted_label=1       gt_label=0
   [i= 200]    predicted_label=1       gt_label=0
   [i= 300]    predicted_label=1       gt_label=1

Overall classification accuracy: 50.76%


Number of positive reviews tested: 200


Number of negative reviews tested: 195


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                 0.0              100.0
true positive:                 0.0              100.0
```



TEXTnet with binary encoding



```
The number of layers in the model: 6

The number of learnable parameters in the model: 323950

The size of the vocabulary (which is also the size of the one-hot vecs for words): 17001



[epoch:1  iter: 100  elapsed_time:    6 secs]     loss: 0.69195
[epoch:1  iter: 200  elapsed_time:   11 secs]     loss: 0.69864
[epoch:1  iter: 300  elapsed_time:   17 secs]     loss: 0.69217
[epoch:1  iter: 400  elapsed_time:   22 secs]     loss: 0.68946
[epoch:1  iter: 500  elapsed_time:   27 secs]     loss: 0.70003
[epoch:1  iter: 600  elapsed_time:   33 secs]     loss: 0.70256
[epoch:1  iter: 700  elapsed_time:   38 secs]     loss: 0.69173
[epoch:1  iter: 800  elapsed_time:   43 secs]     loss: 0.69935
[epoch:1  iter: 900  elapsed_time:   49 secs]     loss: 0.69465
[epoch:1  iter:1000  elapsed_time:   54 secs]     loss: 0.69520
[epoch:1  iter:1100  elapsed_time:   59 secs]     loss: 0.69792
[epoch:1  iter:1200  elapsed_time:   65 secs]     loss: 0.69327
[epoch:1  iter:1300  elapsed_time:   71 secs]     loss: 0.69209
[epoch:1  iter:1400  elapsed_time:   77 secs]     loss: 0.69451
[epoch:1  iter:1500  elapsed_time:   82 secs]     loss: 0.69421

Finished Training

No handles with labels found to put in legend.
<Figure size 1000x500 with 1 Axes>
   [i= 100]    predicted_label=1       gt_label=0
   [i= 200]    predicted_label=1       gt_label=0
   [i= 300]    predicted_label=1       gt_label=1

Overall classification accuracy: 52.03%


Number of positive reviews tested: 200


Number of negative reviews tested: 195


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:               4.615            95.385
true positive:                 2.0              98.0
```



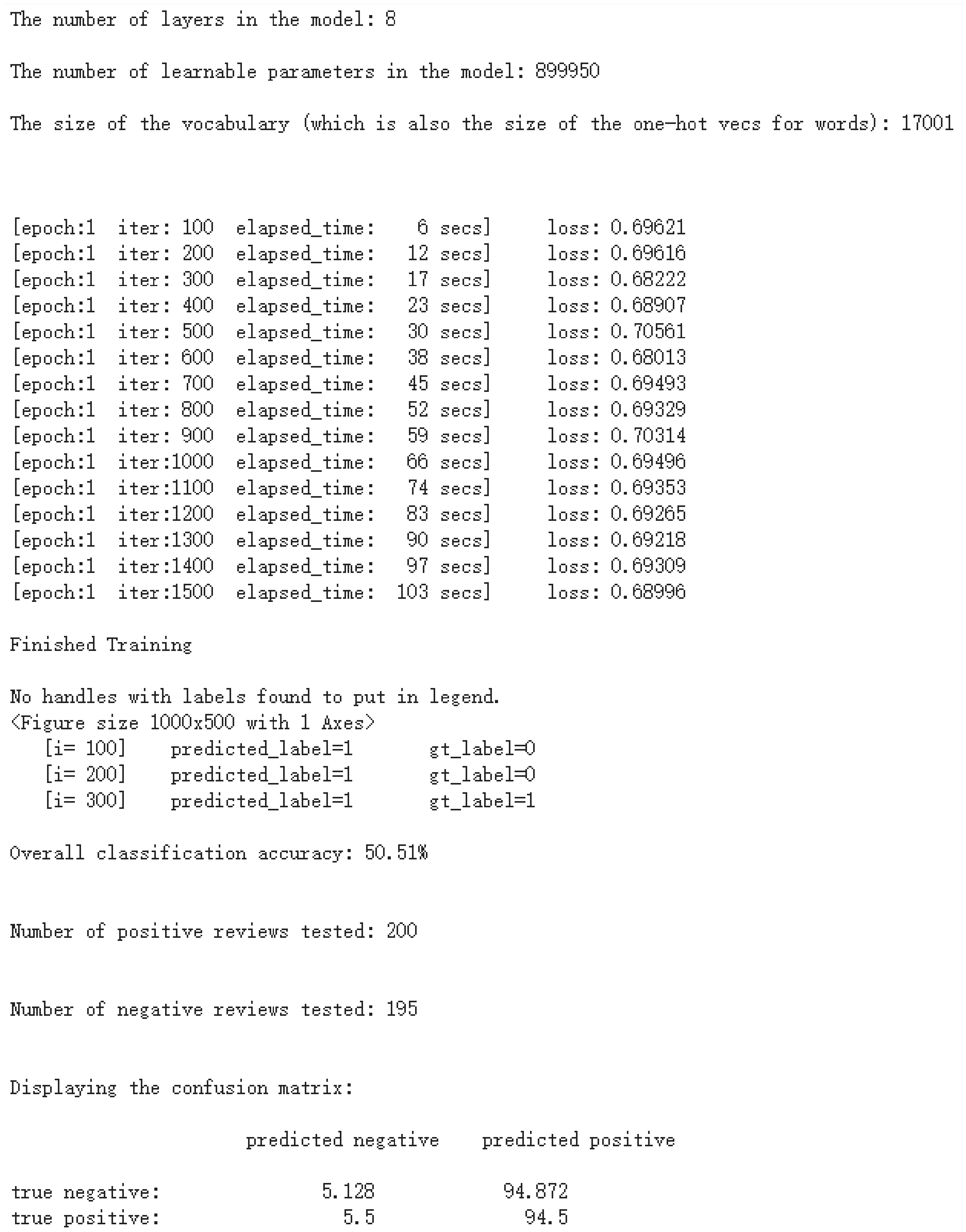
The results screenshots show that, after using binary encoding, the number of learnable parameters decreased from 10718770 to 323950, training time decreased and the accuracy increased about 2%. Though after using the binary encoding the TNN and TPN are no longer zeros, the TNP is still high.

Task 2

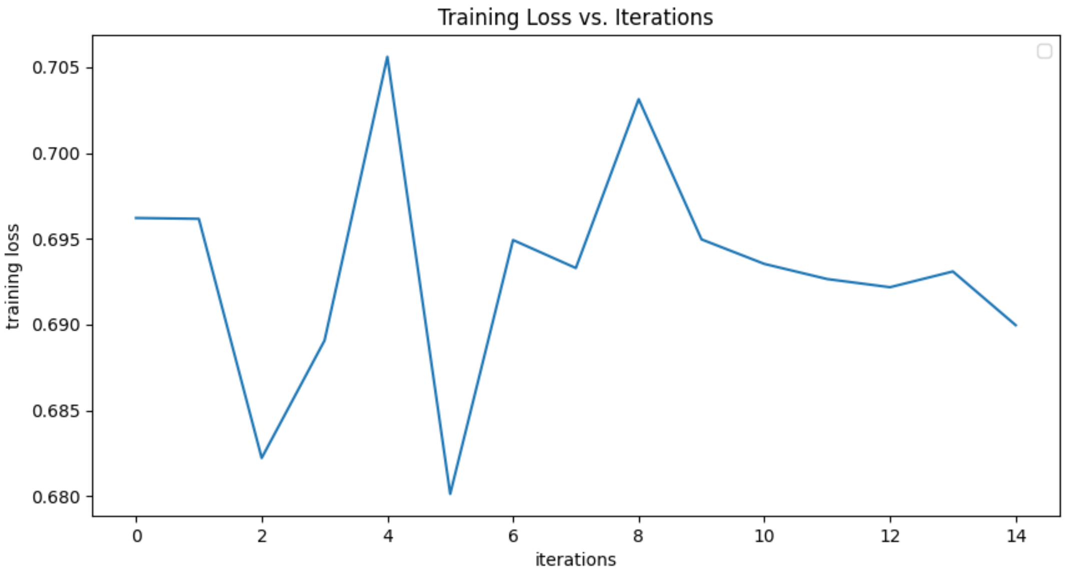
The goal of task 2 is including additional gating action in *TEXTnetOrder2* to further improve the performance. Given the architecture of *TEXTnetOrder2* in Week13 slides, the sigmoid function of cell state plays a switch role. Additionally, I added another switch for *hidden* as the additional gating action. These two switches determine how much from previous *hidden* will be used. For reference, check *task2.py*. Following screenshots show the results and loss.

For this task, I used epoch of 1, learning rate of 1e-5, momentum 0.9.

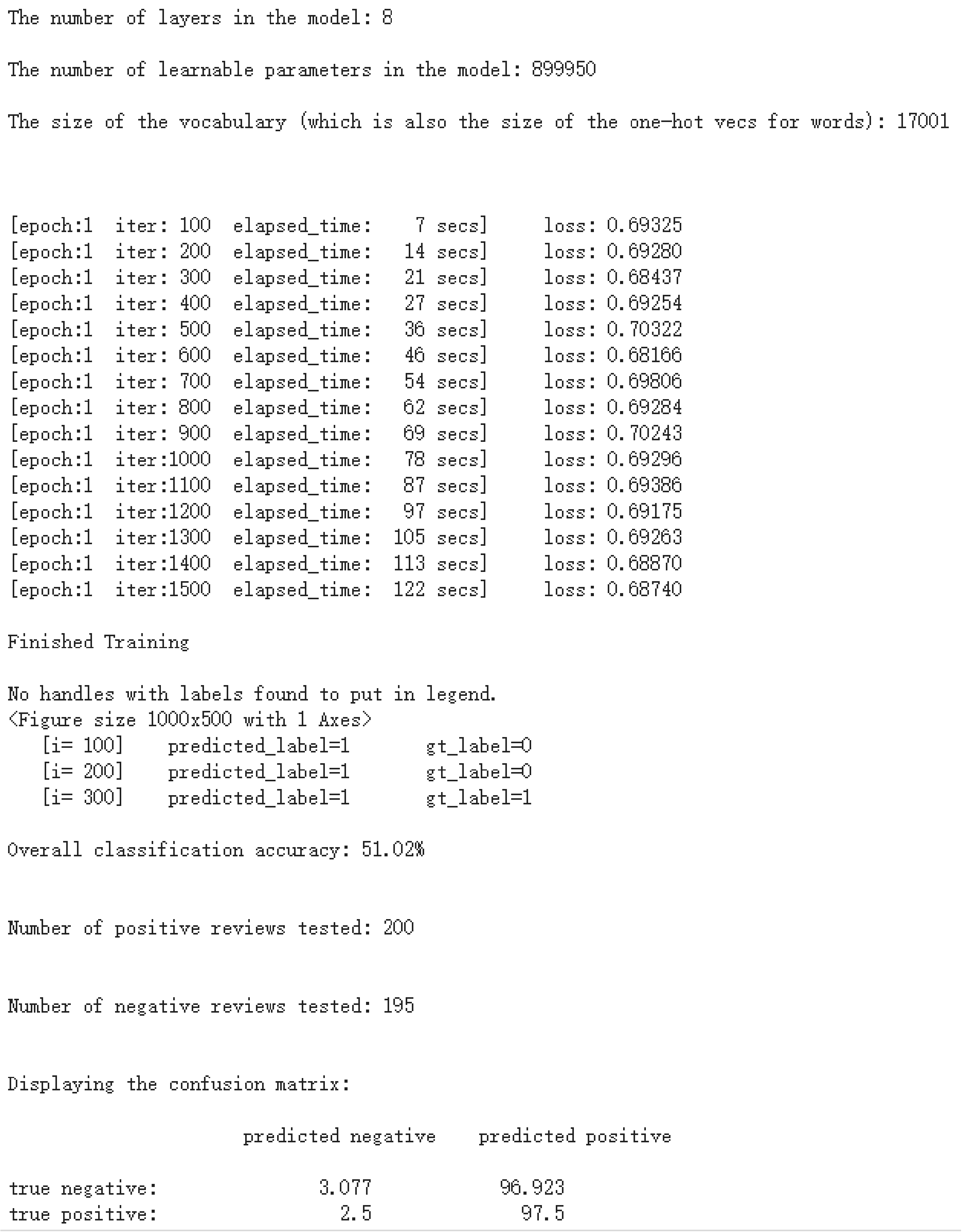
TEXTnetOrder2 original setting with binary encoding



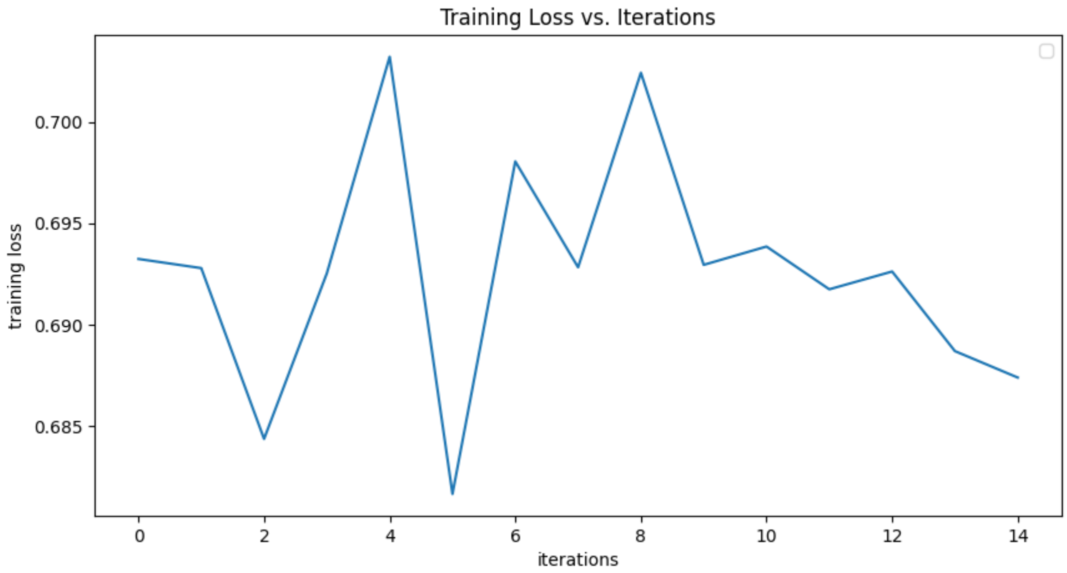
```
The number of layers in the model: 8

The number of learnable parameters in the model: 899950

The size of the vocabulary (which is also the size of the one-hot vecs for words): 17001



[epoch:1  iter: 100  elapsed_time:    6 secs]     loss: 0.69621
[epoch:1  iter: 200  elapsed_time:   12 secs]     loss: 0.69616
[epoch:1  iter: 300  elapsed_time:   17 secs]     loss: 0.68222
[epoch:1  iter: 400  elapsed_time:   23 secs]     loss: 0.68907
[epoch:1  iter: 500  elapsed_time:   30 secs]     loss: 0.70561
[epoch:1  iter: 600  elapsed_time:   38 secs]     loss: 0.68013
[epoch:1  iter: 700  elapsed_time:   45 secs]     loss: 0.69493
[epoch:1  iter: 800  elapsed_time:   52 secs]     loss: 0.69329
[epoch:1  iter: 900  elapsed_time:   59 secs]     loss: 0.70314
[epoch:1  iter:1000  elapsed_time:   66 secs]     loss: 0.69496
[epoch:1  iter:1100  elapsed_time:   74 secs]     loss: 0.69353
[epoch:1  iter:1200  elapsed_time:   83 secs]     loss: 0.69265
[epoch:1  iter:1300  elapsed_time:   90 secs]     loss: 0.69218
[epoch:1  iter:1400  elapsed_time:   97 secs]     loss: 0.69309
[epoch:1  iter:1500  elapsed_time:  103 secs]     loss: 0.68996

Finished Training

No handles with labels found to put in legend.
<Figure size 1000x500 with 1 Axes>
   [i= 100]    predicted_label=1       gt_label=0
   [i= 200]    predicted_label=1       gt_label=0
   [i= 300]    predicted_label=1       gt_label=1

Overall classification accuracy: 50.51%


Number of positive reviews tested: 200


Number of negative reviews tested: 195


Displaying the confusion matrix:

                    predicted negative    predicted positive

true negative:            5.128            94.872
true positive:              5.5              94.5
```



TEXTnetOrder2 gating setting with binary encoding



```
The number of layers in the model: 8

The number of learnable parameters in the model: 899950

The size of the vocabulary (which is also the size of the one-hot vecs for words): 17001



[epoch:1  iter: 100  elapsed_time:    7 secs]     loss: 0.69325
[epoch:1  iter: 200  elapsed_time:   14 secs]     loss: 0.69280
[epoch:1  iter: 300  elapsed_time:   21 secs]     loss: 0.68437
[epoch:1  iter: 400  elapsed_time:   27 secs]     loss: 0.69254
[epoch:1  iter: 500  elapsed_time:   36 secs]     loss: 0.70322
[epoch:1  iter: 600  elapsed_time:   46 secs]     loss: 0.68166
[epoch:1  iter: 700  elapsed_time:   54 secs]     loss: 0.69806
[epoch:1  iter: 800  elapsed_time:   62 secs]     loss: 0.69284
[epoch:1  iter: 900  elapsed_time:   69 secs]     loss: 0.70243
[epoch:1  iter:1000  elapsed_time:   78 secs]     loss: 0.69296
[epoch:1  iter:1100  elapsed_time:   87 secs]     loss: 0.69386
[epoch:1  iter:1200  elapsed_time:   97 secs]     loss: 0.69175
[epoch:1  iter:1300  elapsed_time:  105 secs]     loss: 0.69263
[epoch:1  iter:1400  elapsed_time:  113 secs]     loss: 0.68870
[epoch:1  iter:1500  elapsed_time:  122 secs]     loss: 0.68740

Finished Training

No handles with labels found to put in legend.
<Figure size 1000x500 with 1 Axes>
   [i= 100]    predicted_label=1       gt_label=0
   [i= 200]    predicted_label=1       gt_label=0
   [i= 300]    predicted_label=1       gt_label=1

Overall classification accuracy: 51.02%


Number of positive reviews tested: 200


Number of negative reviews tested: 195


Displaying the confusion matrix:

                    predicted negative    predicted positive

true negative:            3.077              96.923
true positive:              2.5                97.5
```



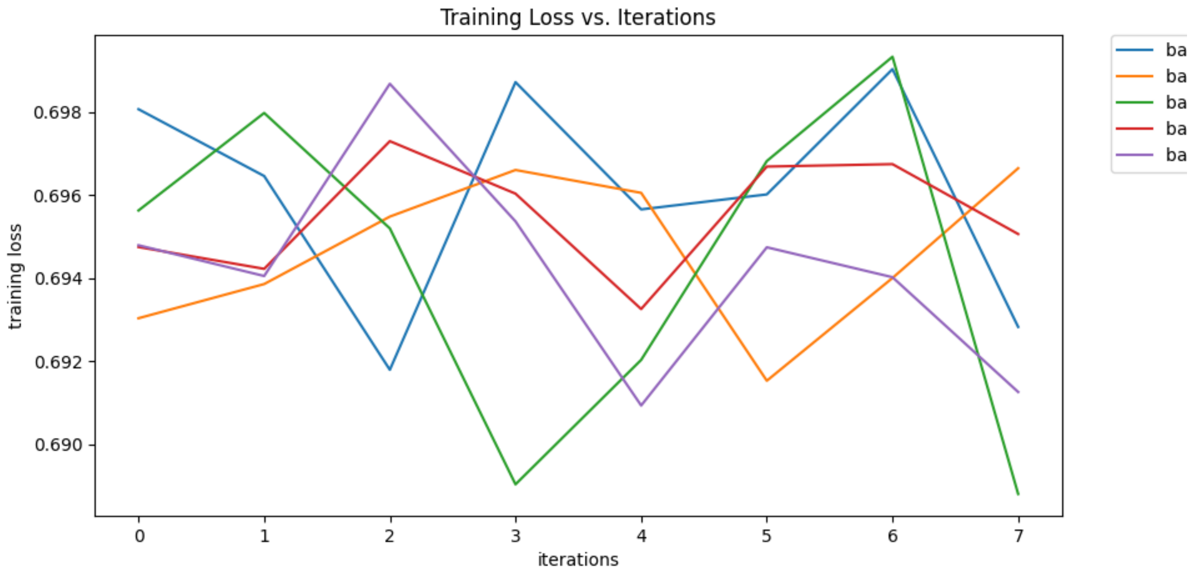
As the results showed, after using the additional gating action, accuracies increased.

Task 3

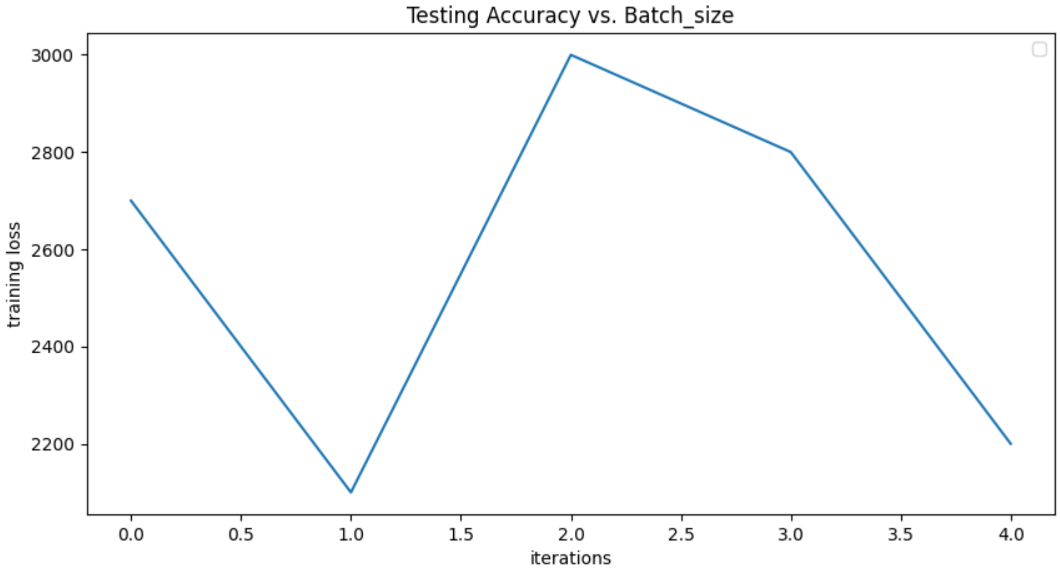
The goal of task3 is finding out how batch-size affect the quality of results. For this task, I decided to pad each review with 0's so that they come up to the length of the longest review. To do this, I created *TextClassification_extend* and *SentimentAnalysisDataset_extend* inherited from original classes. While creating the training and testing data, *train_max_len* and *test_max_len* track the longest review size and feed into *review_to_tensor* to create equal size review tensor. In the training and testing part, I used a for loop to iterate from batch-size of 1 to 3 and plot the corresponding loss. For reference, check *task3.py*. Following screenshots show the loss results and accuracies.

For this task, I used epoch of 8, batch-size from 1 to 5, learning rate of 1e-5, momentum 0.9,

TEXTnetOrder2 gating setting with binary encoding Loss at different Batch-size

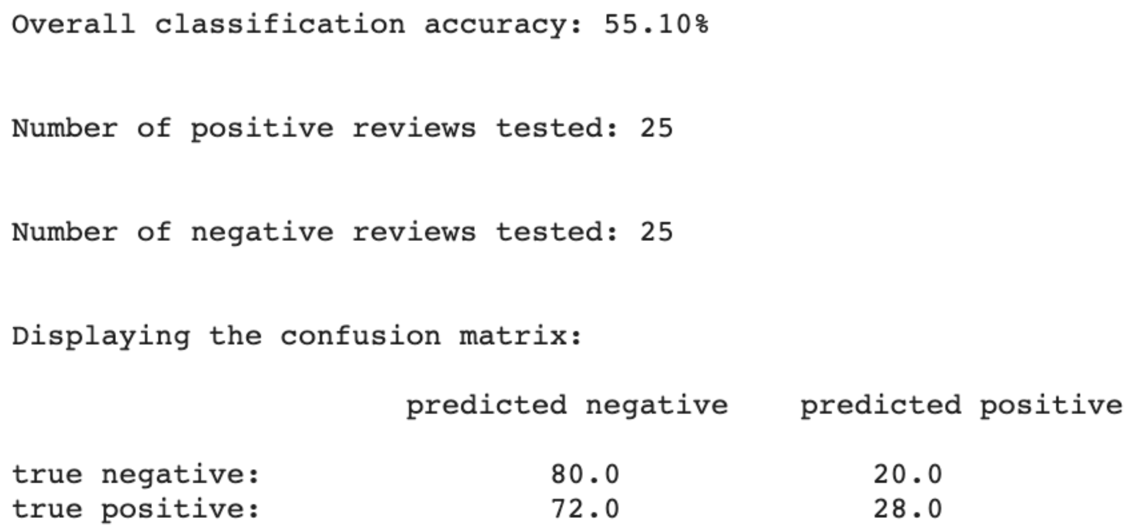


TEXTnetOrder2 gating setting with binary encoding Accuracies at different Batch-size



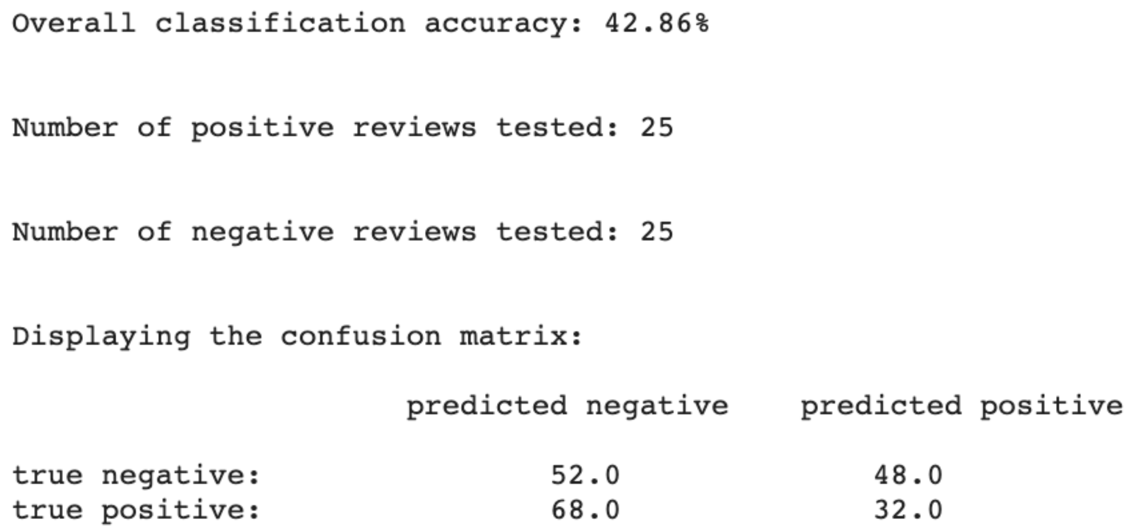
Confusion Matrices

Batch-size-1



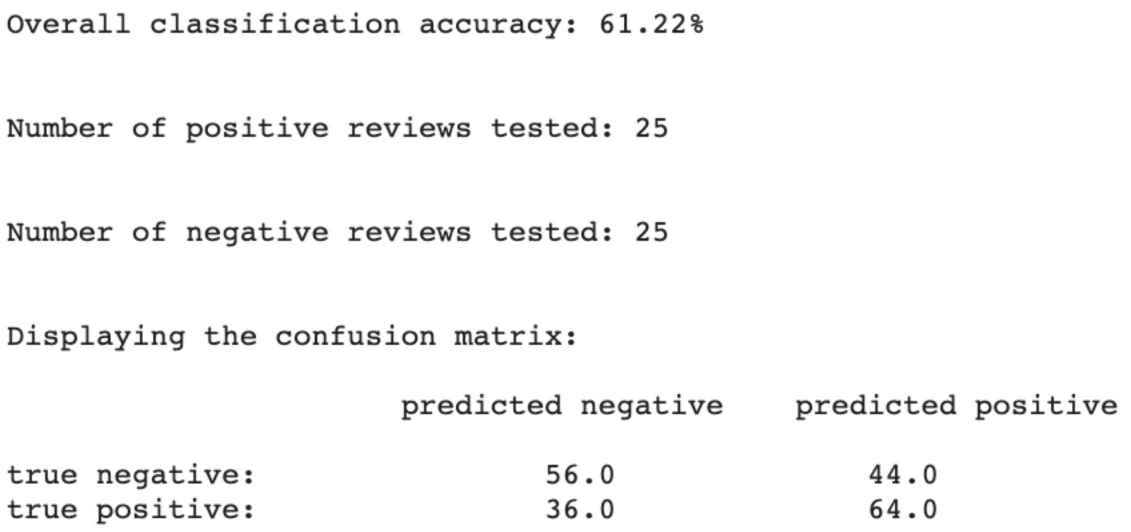
```
Overall classification accuracy: 55.10%


Number of positive reviews tested: 25


Number of negative reviews tested: 25


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                80.0                  20.0
true positive:                72.0                  28.0
```

Batch-size-2



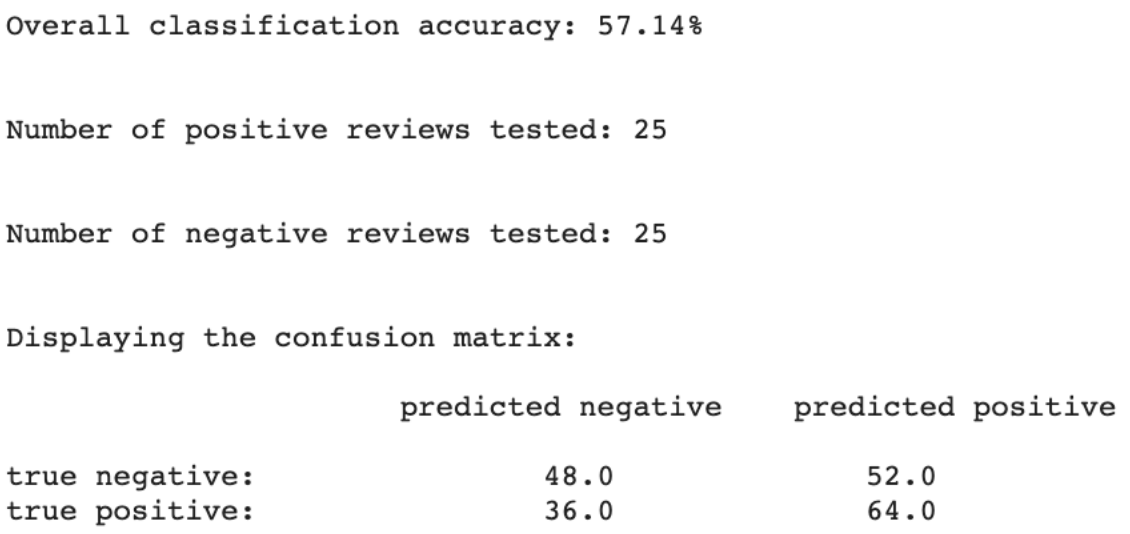
```
Overall classification accuracy: 42.86%


Number of positive reviews tested: 25


Number of negative reviews tested: 25


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                52.0                  48.0
true positive:                68.0                  32.0
```

Batch-size-3



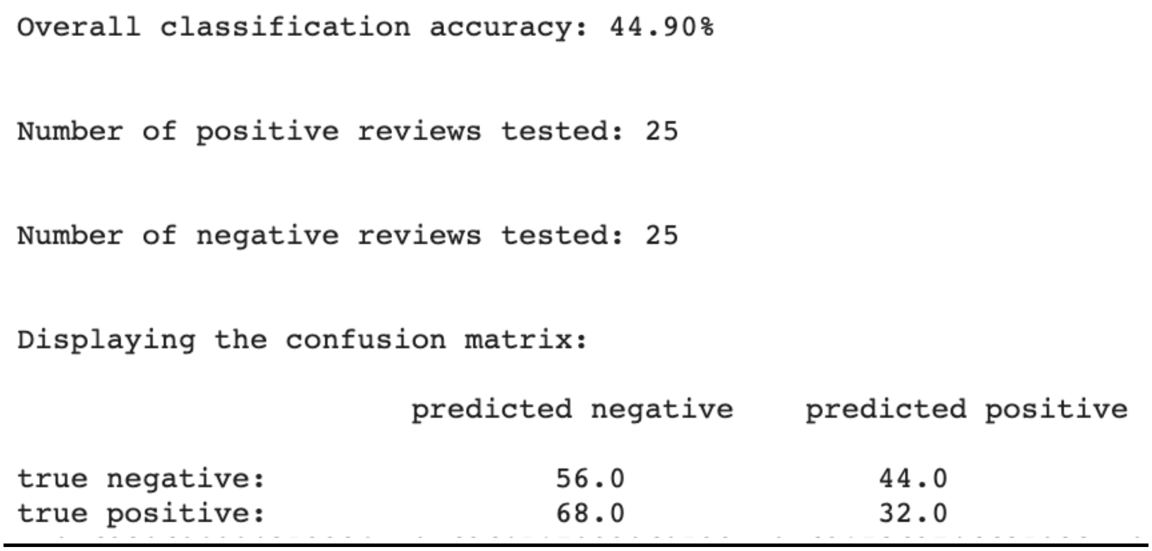
```
Overall classification accuracy: 61.22%


Number of positive reviews tested: 25


Number of negative reviews tested: 25


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                56.0                  44.0
true positive:                36.0                  64.0
```

Batch-size-4



```
Overall classification accuracy: 57.14%


Number of positive reviews tested: 25


Number of negative reviews tested: 25


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                48.0                  52.0
true positive:                36.0                  64.0
```

Batch-size-5



```
Overall classification accuracy: 44.90%


Number of positive reviews tested: 25


Number of negative reviews tested: 25


Displaying the confusion matrix:

                      predicted negative    predicted positive

true negative:                56.0                  44.0
true positive:                68.0                  32.0
```

Based on the results, batch-size seems has both positive and negative effects over the performance of the *TEXTnetOrder2* model. Model performed the best at batch-size of 3. The reason could be using smaller size dataset.