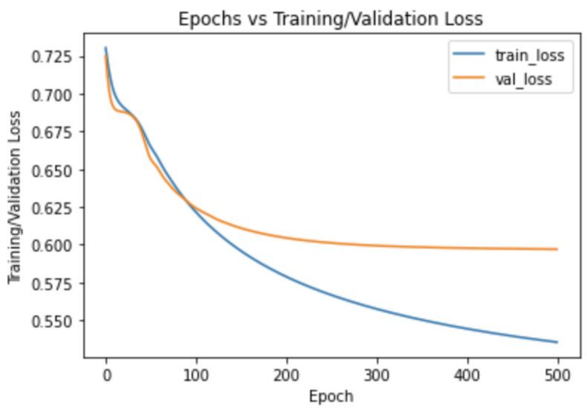
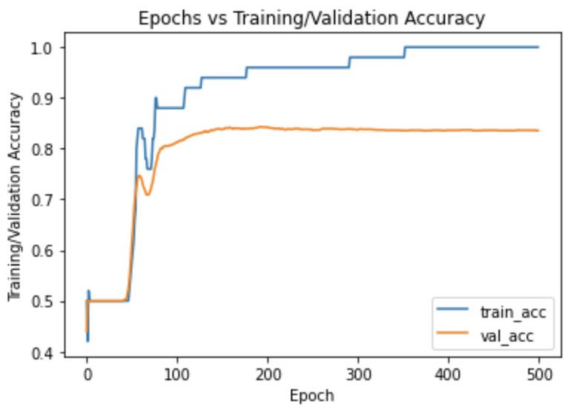
Project Report

Train/Validation/Test splits

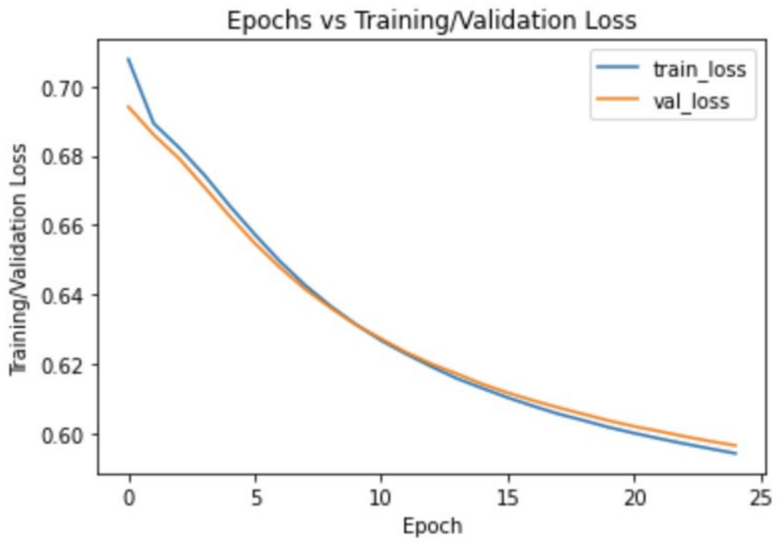
```
train.tsv:
        Objective:3200; Subjective:3200

validation.tsv:
        Objective:800; Subjective:800

test.tsv:
        Objective:1000; Subjective:1000

overfit.tsv:
        Objective:25; Subjective:25
```

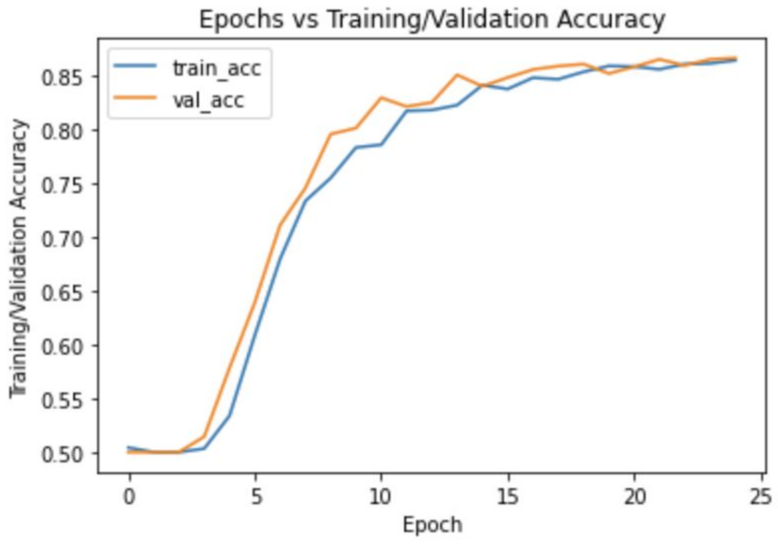
Overfitting to debug





Full Training Data



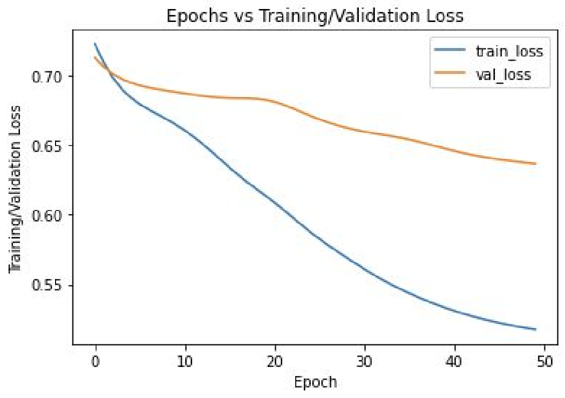


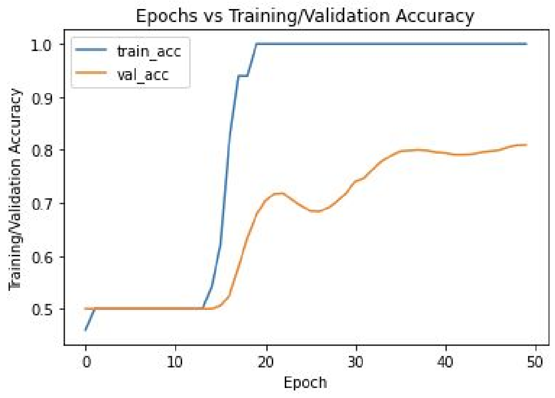
Final Testing accuracy: 0.857

Final Testing loss: 0.601

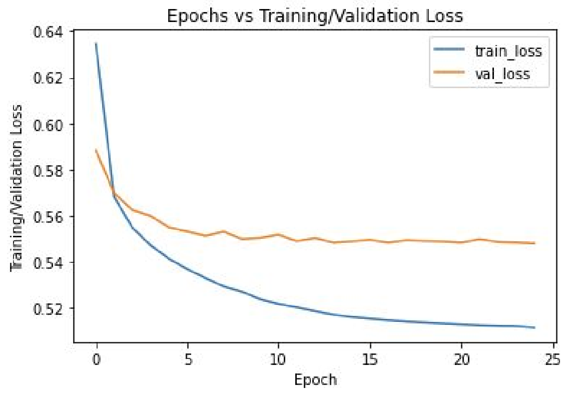
Overfit, Training and Test (CNN)

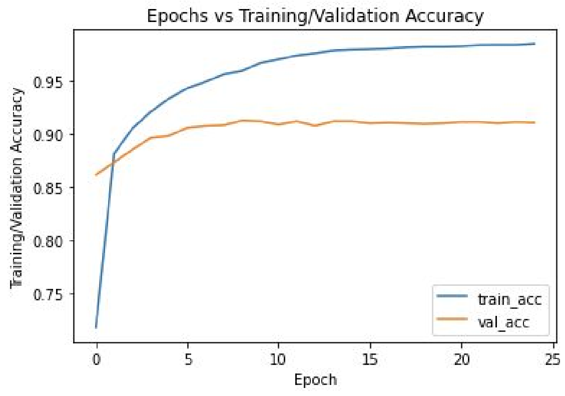
Overfit model





Full training model



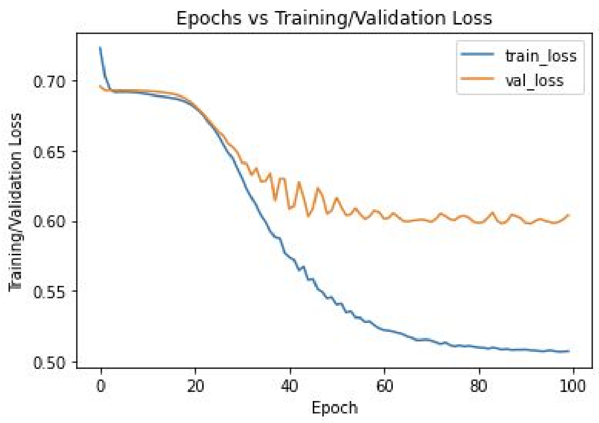


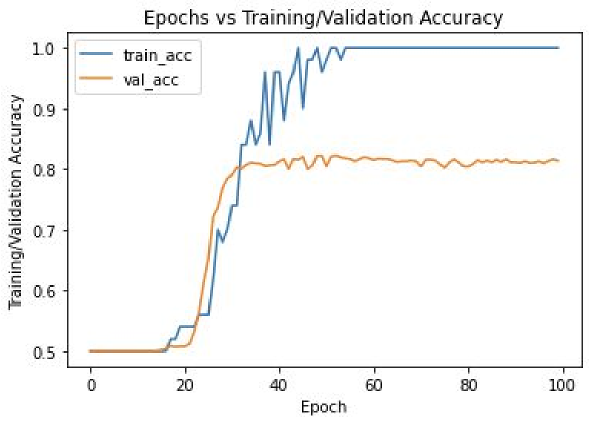
Final Testing accuracy: 0.917

Final Testing loss: 0.550

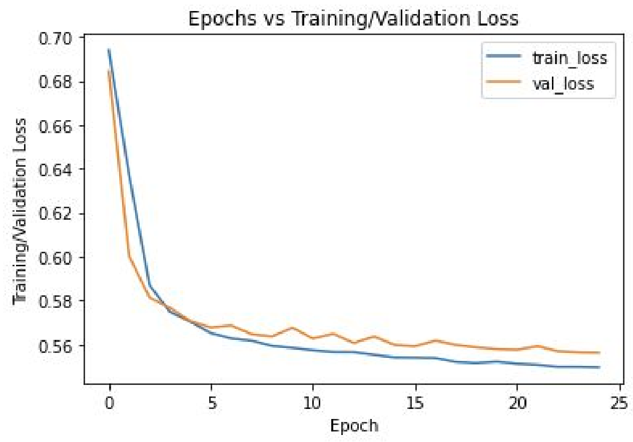
Overfit, Training and Test (RNN)

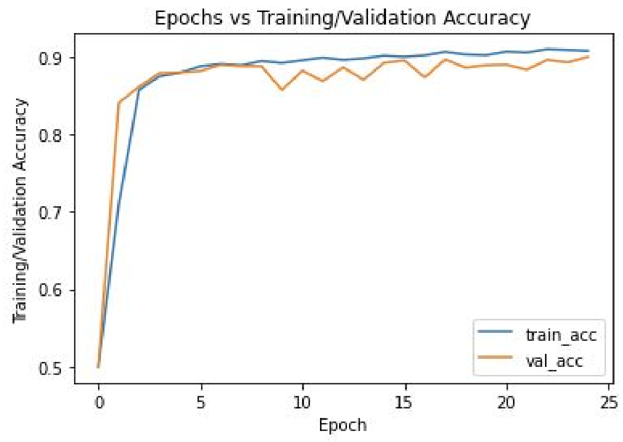
Overfit model





Full training model





Final Testing accuracy: 0.903

Final Testing loss: 0.558

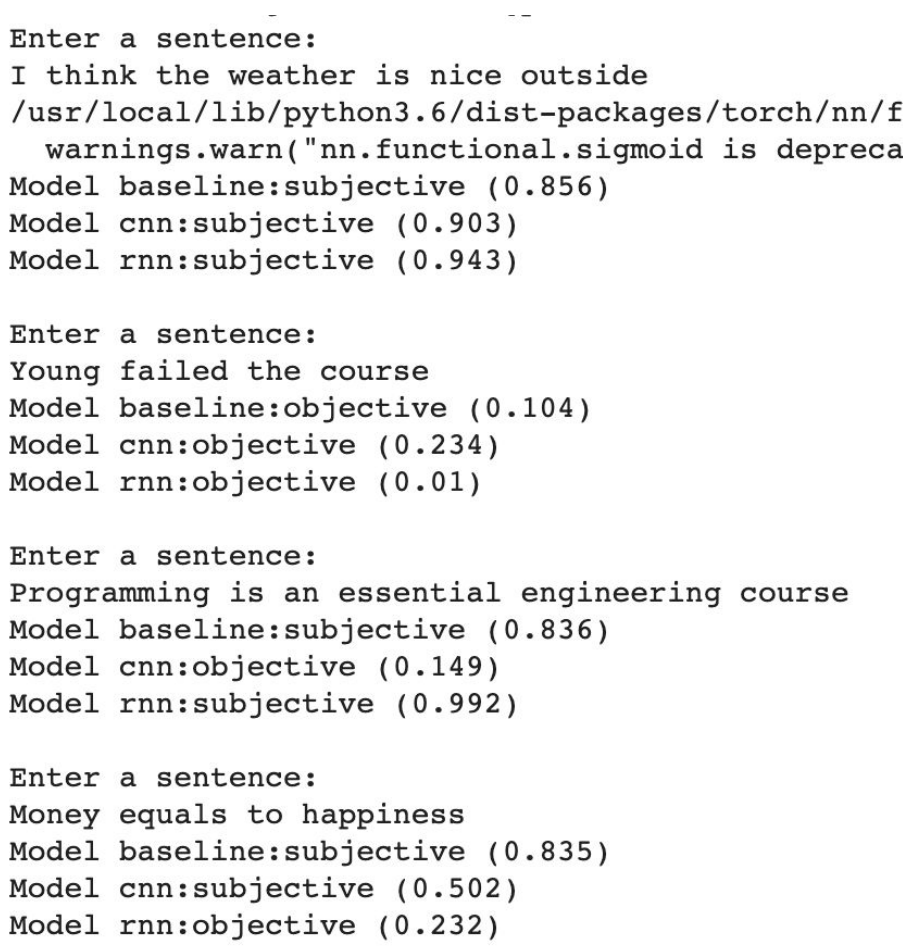
Conclusions

	Training Loss	Validation Loss	Testing Loss	Training Accuracy	Validation Accuracy	Testing Accuracy
Baseline	0.594	0.596	0.601	0.862	0.861	0.856
CNN	0.511	0.550	0.550	0.986	0.909	0.917
RNN	0.550	0.556	0.558	0.907	0.899	0.903

The model that performs the best is the CNN model. There isn't a significant difference between the validation and testing accuracy. This is because for all the 3 models, the hyperparameters are all predefined constants. Hence, both validation and testing datasets have no influence on the model (essentially can be both considered the "same"). Thus, the small difference makes sense, which simply reflects the randomness between the 2 datasets. However, if one were to tune hyperparameters based on validation accuracy, then there could be a bigger difference in the testing accuracy, as the model might overfit on the validation dataset.

In the CNN architecture, the kernels are learning to detect "elements of subjectivity and objectivity" in groups of 2 and groups of 4 (corresponding to kernel size 2 and 4). These pairs/quadruplets of words give hints to the subjectivity and objectivity nature of the sentences. During the max pooling operation, the CNN computes the one maximum value from every sentence for every kernel and uses that as the output feature/number. Hence, by only considering the one element that suggests maximum subjectivity/objectivity for the entire sentence, the model is discarding information regarding the context of the sentence. This is, combinations of words can have a stronger implication with regards to the nature of the sentence but they are not considered by the max pooling operation. This is similar to the baseline model's discarding of information, as it simply takes the numerical averages of the embedded word vectors and by doing so, contextual information will also be lost with regards to the sentence as a whole.

Creating new sentences



```
Enter a sentence:
I think the weather is nice outside
/usr/local/lib/python3.6/dist-packages/torch/nn/f
  warnings.warn("nn.functional.sigmoid is depreca
Model baseline:subjective (0.856)
Model cnn:subjective (0.903)
Model rnn:subjective (0.943)

Enter a sentence:
Young failed the course
Model baseline:objective (0.104)
Model cnn:objective (0.234)
Model rnn:objective (0.01)

Enter a sentence:
Programming is an essential engineering course
Model baseline:subjective (0.836)
Model cnn:objective (0.149)
Model rnn:subjective (0.992)

Enter a sentence:
Money equals to happiness
Model baseline:subjective (0.835)
Model cnn:subjective (0.502)
Model rnn:objective (0.232)
```

For the 2 definitive cases, all 3 models come to the same, correct conclusions. For sentence 3, “Programming is an essential engineering course”, personally I think this should be an objective fact. However, the majority of the models, specifically baseline and rnn, think that this is a subjective sentence. For sentence 4, “money equals to happiness”, which should be a subjective sentence, the majority of the models come to the correct conclusion (i.e, baseline and cnn). Overall, just based on these 4 testing sentences, the CNN model seems to perform the best, as it arrives at the “correct” conclusions for all 4 cases, whereas the other 2 models are at least wrong once.