Yuan Hong (Bill) Sun
1003039838

Assignment 4: Subjective/Objective Sentence Classification Using Word Vectors & NLP

7. Grading Experimental and Conceptual Questions

1. Comparing the 3 basic models (25 epochs, default hyperparameters, no other modifications):

Model	Training Acc	Training Loss	Valid Acc	Valid Loss	Test Acc	Test Loss
Base	91.88%	0.2199	87.12%	0.3518	86.80%	0.3130
RNN	100%	0.00015	92.56%	0.1225	92.30%	0.2031
CNN	100%	0.00473	91.19%	0.1662	90.75%	0.3649

The RNN model has the highest validation and test accuracy, as well as the lowest validation loss. The CNN model is a close second, with similar but slightly lower validation and test accuracies. There is very little difference between the validation accuracy and test accuracy since the validation and test data came from the same source. The baseline model performed much more poorly because it omitted a lot of information by taking the averages of word vectors. All three models had some overfitting.

2. In the baseline model, since only the average index across each row along the word vectors is taken, it omitted "meanings" of individual words, as well as the order of words and grammatical structure in each sentence. It is important since this information is crucial for determining the overall meaning of a sentence; and removing the order and structure of words would only make the sentence meaning nonsense. The omission of this information is reflected by the lower validation and test accuracies given by the baseline model compared to the RNN and CNN models. Therefore, it is important to retain individual indices of the word vectors that make up the order and grammatical structure of the sentences.

3. Examining effects of *pack_padded_sequence* and using *BucketIterator* (default hyperparams)

Model	Training Acc	Training Loss	Valid Acc	Valid Loss	Test Acc	Test Loss
a).	100%	0.00015	92.56%	0.1225	92.30%	0.2031
b).	99.9%	0.00041	91.62%	0.1197	83.35%	0.5129
c).	95.2%	0.00338	90.38%	0.1549	92.25%	0.2137

a). With *pack_padded_sequence* and BucketIterator
b). Without *pack_padded_sequence* and *BucketIterator*
c). Without *pack_padded_sequence* and *Iterator*

Without pack padding, the model performs somewhat worse than the default settings since instead of taking in just the length of each sentence (after removing the blank spaces after each sentence), it also took in empty blocks/memory that the sentence did not occupy. This useless blank data corrupts the training, validation, and test data, which results in lower accuracies and longer training time to reach a desired accuracy.

The BucketIterator organizes sentences by length by grouping sentences of similar lengths together, so there would be less variance in sentence length in each batch. This would improve the model performance as there is fewer blank data in each batch. By using Iterator instead of BucketIterator, the validation accuracy is decreased, and the model takes longer to converge, as seen in case c).

Furthermore, in case b), when the models are tested on all the test data at once (or one very large test set), the test accuracy is poor since there is a much larger variance in sentence lengths, and without using pack padding, there is much more blank data, and it will therefore corrupt the test results. In case c) however, using a very large test set wouldn't make too much difference in the test accuracy since the variance of lengths of sentences in individual batches no longer matters when they are not sorted by length in the Iterator. Therefore, there is a lower test accuracy for b) and higher test accuracy for c).

4. In the CNN architecture, the kernels detect an average meaning within that kernel range. When performing max-pooling, the model marginalizes words that contribute less to the overall "meaning" while retaining the ones that have a larger impact to the overall meaning inside the word vectors by consolidating them and taking a "mean".

This is slightly different from the baseline model, but it still retains more information than the baseline model as the baseline model simply takes one average for an entire row of the stacked word vectors, while the CNN maxpool takes multiple "means" across different kernels of the stacked word vectors. Because the kernels iterate through the word vectors in a dynamic fashion during max pooling, it retains more information than the baseline model.

5. Manual test cases:
-Test case 1 (strongly objective): Roses are red, violets are blue
-Test case 2 (strongly subjective): Artificial intelligence is very important

-Test case 3 (borderline objective): The Raptors currently have a 11-1 record and they are having a great season so far
-Test case 4 (borderline subjective): The Raptors are having their best ever lineup in history

Results:

	Test case 1	**Test case 2**	**Test case 3**	**Test case 4**
Base prediction	Objective (0.293)	Subjective (0.967)	Objective (0.459)	Objective (0.480)
RNN prediction	Objective (0.131)	Subjective (0.785)	Subjective (0.692)	Objective (0.000)
CNN prediction	Objective (0.009)	Subjective (0.959)	Objective (0.024)	Objective (0.042)

On the first two cases, all three models made the correct prediction by a fair margin (i.e. values away from 0.5). In the third case, the RNN model made the incorrect prediction, while the base model made a marginally correct prediction. All 3 models failed to make the correct prediction in the last case, although the baseline model was close to being correct. This may be because the 4th test case can also be interpreted as an objective sentence (or a truth rather than an opinion) despite it being subjective.

Overall, judging by the values of each prediction, the CNN made the best predictions overall (not counting test case 4). The RNN model comes in second, where it performed well on the two obvious cases, but gave incorrect predictions on the borderline cases. It is expected that the CNN performs the best since it best retains the meanings of words that had the most impact on the overall meaning of each sentence, while the RNN just weights all the words equally, which can result in a lower performance. The baseline model is quite indecisive.

6. Feedback
a). Approximately 30 hours (including time setting up spacy which did not install on my laptop at first due to dependency issues)

b). Mainly programming the neural network models (CNN and RNN). It took a lot of research and time to figure out layer dimensions. Also, because there wasn't enough time to fully understand the assignment, a lot of it was done through trial-and-error, and teamwork.

c). It was enjoyable at the moment when the code starts working, and that the training and validation accuracies were quite high for the models, which was satisfying.

d). Some of the instructions were unclear, such as Sections 3.2 and 4.2. Also, there was a line in Section 3.2 that said, "This section needs to be improved", but it never was in subsequent

updates in the assignment. Furthermore, some of the documentations had little information and examples on how to use the function, so a lot of additional research had to be done. It was also confusing trying to understand the behavior of each model, but after understanding what each function does, it becomes more clear.

e). The diagrams were helpful in visualizing and understanding the structure of each model.