Subjective/Objective Sentence Classification Using Word Vectors and NLP

Section 3 – Preparing the Data

Section 3.1 – Create train/validation/test splits

Table 1: Split values for each dataset

Label	1	0	Total
Train	3200	3200	6400 (64%)
Validation	800	800	1600 (16%)
Test	1000	1000	2000 (20%)

Section 7 – Grading Experimental and Conceptual Questions

1.

Table 2: Train/Val/Test Acc/Loss for Baseline/RNN/CNN

	Baseline	RNN	CNN
Train Acc	87.5%	99.9%	100%
Train Loss	0.409	0.003	0.006
Validation Acc	86.5%	91.9%	90.8%
Validation Loss	0.409	0.379	0.276
Test Acc	86.8%	92.0%	91.3%
Test Loss	0.405	0.414	0.269

RNN model performed best (marginally) however, these were the values obtained for one run. Analyzing that of many runs, it can be seen that models RNN and CNN both had very similar test accuracy values at around 92%. There was very little to no difference between both the validation and training set for both accuracy and loss when comparing them against each other on the same model (as depicted in Figures 1-3). This is because normally, when training the model, we train the model on both the training set and *indirectly* on the validation set. We aim to achieve as high of an accuracy/minimal loss ass possible by "testing" the model on the validation set. We then adjust the hyperparameters accordingly such that the validation accuracy increase. Since there was no such modification in hyperparameters, both the validation and test set were considered "never before seen" samples for the model. There was no refinement to make the validation set better, thus the results should be very similar, which they are.

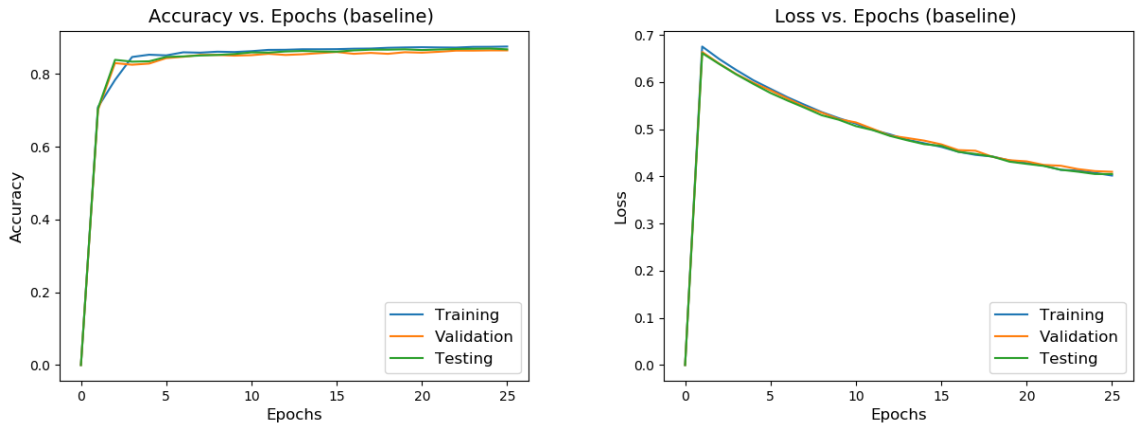


Figure 1: Accuracy and Loss vs. Epoch for model Baseline

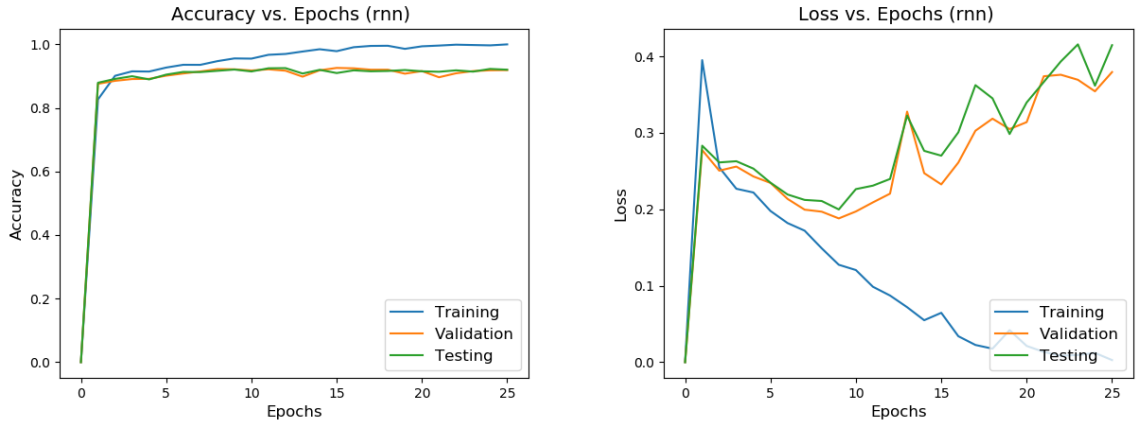


Figure 2: Accuracy and Loss vs. Epoch for model RNN

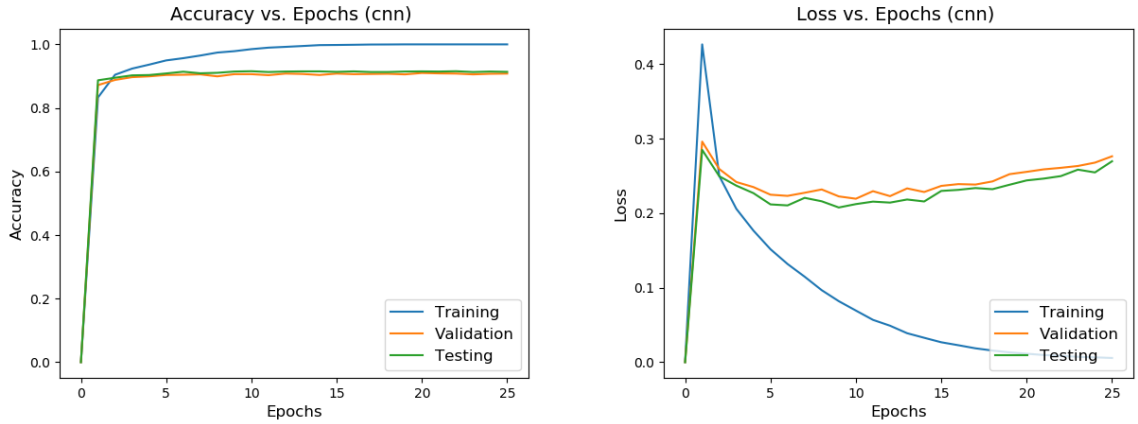
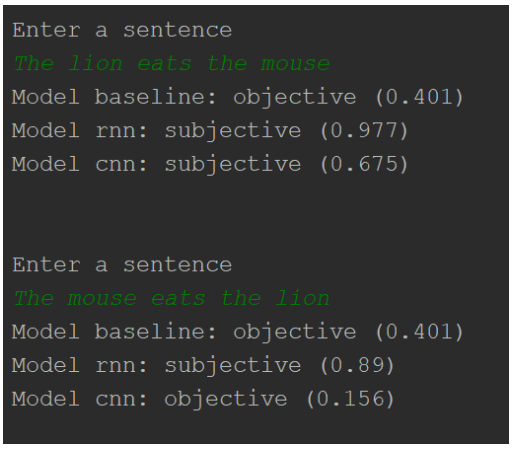


Figure 3: Accuracy and Loss vs. Epoch for model CNN

2.

In the baseline model, the values of each word were summed up and averaged out. This causes the loss of the meaning of words that are beside each other. A sentence such as “The lion eats the mouse” and “The mouse eats the lion” would essentially the same meaning/value since the same words appear in both sentences, which it does as seen below in Figure 4 (a). Both sentences although containing the same words mean somewhat different things, but the baseline model sees it as the same. Also, descriptors (i.e. adjectives) would not necessarily lose its meaning but the context in which it was used in will be lost. Sentences like “The hot coffee was spilt on the man” vs. “The coffee was spilt on the hot man” would have the same score from the baseline model but one describes the temperature of coffee whereas the other describes the attractiveness of the man. Since in both cases, the baseline scores are and will be identical, it shows that such details like how words are oriented in a sentence, adjective placement, etc. matter in the meaning of the sentence itself.



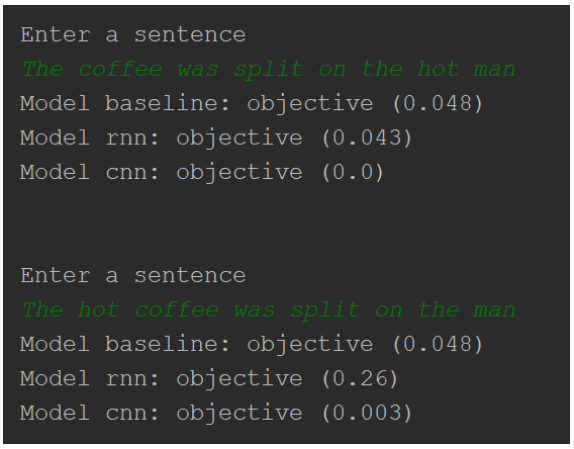


Figure 4: Rearrangement of a sentence, same baseline score. (a) Lion and Mouse (b) hot coffee and hot man

3.

Table 3: Train/Val/Test Acc/Loss for Cases A, B, C on RNN

	(a) Default	(b) No padding	(c) b and Iterator
Train Acc	99.9%	100%	97.4%
Train Loss	0.003	0.001	0.081
Validation Acc	91.9%	91.2%	88.5%
Validation Loss	0.379	0.484	0.332
Test Acc	92.0%	91.2%	89.7%
Test Loss	0.414	0.508	0.317

Given the results, it seems that there is a slight decrease in accuracy from the default case to case (b), where no padding was applied and another slight drop between case (b) and case (c), where the BucketIterator was swapped with Iterator. Below, in Figure 5 outlines the lengths of sentences of a batch using case (c). It is noted that many attempts of running case (c) were required to obtain the results seen in the table above. Normally, the results of all cases, a through c, were very similar, accuracies of around 1.0, 0.91-0.92, and 0.91-0.92 for training, validation and testing respectively. Although normally the results are what I statement prior to this, it should not be the case. It is because in case (b), when no padding is applied, sentences that are shorter than others in the same batch will have an extra GRU layer applied. Referring back to the assignment handout, specifically in Figure 4, without the padding, it would cause the final hidden state to be wrong since some sentences end earlier than others. The padding allows this issue to be fixed and make the RNN stop at the last word on the statement rather than the last word of the longest statement in that batch. The caveat to this however is that since the lengths of the sentences are very similar (due to BucketIterator), there is not too much of a difference between the two cases a and b (as seen above in the results and as stated before), albeit there should be one. This changes when case (c) is mentioned. Since BucketIterator groups sentences of similar lengths, the hidden states of each sentence would be around the same mark, maybe +/- 2 words (from what I had observed) from max length to min length in each batch. In case c, Iterator is used, meaning sentences of different lengths are being fed as batches, as outlined in Figure 5. This is a more severe case of case b since now sentences that are much shorter than the longest will be exposed to more zeroes, and the hidden layer would be wrong, which in turn justifies why case (c) should have the lowest accuracy, which it tends to have, albeit very little difference.

```
tensor([55, 53, 53, 50, 46, 40, 40, 38, 37, 36,
 35, 35, 35, 35, 33, 32, 31, 31,
       31, 31, 30, 28, 27, 27, 27, 27, 27, 26,
 25, 25, 25, 24, 24, 24, 22, 22,
       22, 22, 22, 22, 21, 21, 20, 19, 18, 18,
 17, 17, 16, 16, 16, 16, 16, 15,
       15, 15, 15, 13, 13, 12, 12, 11, 11, 10])
```

Figure 5: Lengths of sentences of a batch of RNN using no padding, Iterator (case (c))

4.

In CNN, the kernels are trying to learn the meaning behind the grouping of words, i.e. what hot means when it is beside man vs. when it is beside coffee (from the example above). It is trying measure the importance of sentence word order and the reasoning behind why a word is place beside another word. When max-pooling is applied, it drops the extra meanings that are less significant. If a sentence were to be “I like to make and eat pepperoni pizza” The kernels would detect that “I” and “like” have a high dependence between each other since it’s a subject with a verb or “pepperoni” and “pizza” would be a high importance since “pepperoni” is like a descriptor of the pizza or has relations to it. Things such as “make” and “and” would have less

of an importance since there is not much meaning to be interpreted from the conjunction of the two. Since the kernels find the meaning between all pairs, (or quadruplets for a kernel of size 4), there is a lot of meaning that have less significant value and thus will play no use in the meaning of the sentence. As such, the max-pooling will "pull" out the most important meanings and outputs it to be processed. This is different from the way the baseline model discards information. The baseline model (as described above) does not care about sentence order as words can be completely scrambled up and will still return the same meaning since it averages throughout the whole sentence whereas the CNN does care about the meaning, in sections of size *k* kernel size. It is somewhat similar however since in a way the CNN does depend on the kernel size. If the kernel size is too large, it would end up being very similar to the baseline model as it basically tries to find the average meaning in groups of words between each kernel.

5.

Subjective: I am the best.

Objective: MIE324 is a course at UofT.

Borderline: The MIE324 course is challenging.

Borderline: Jonathan Rose is a tall man.

```
Enter a sentence
I m the best.
Model baseline: subjective (0.601)
Model rnn: subjective (0.958)
Model cnn: subjective (0.994)


Enter a sentence
MIE324 is a course at UofT.
Model baseline: objective (0.326)
Model rnn: objective (0.126)
Model cnn: objective (0.031)
```

```
Enter a sentence
The MIE324 course is challenging.
Model baseline: subjective (0.929)
Model rnn: objective (0.499)
Model cnn: subjective (0.936)


Enter a sentence
Jonathan Rose is a tall man.
Model baseline: subjective (0.574)
Model rnn: objective (0.019)
Model cnn: objective (0.178)
```

Figure 6: Subjective/Objective Sentences. Subjective (top left). Objective (bottom left). Borderline (right)

For subjective, RNN and CNN had a very high prediction for subjective whereas baseline had a low prediction for subjective.

For objective, CNN had a very high prediction for objective, RNN a moderately high prediction and baseline had a low prediction.

For borderline, it was a toss up for all three models. In the first example, RNN had it very close to 50% whereas in the second example, baseline had it at 57%.

For subjective and objective sentences, the results behave as expected since in most subjective cases, there is opinion such as the words “I”, which was identified very well by the RNN and CNN. Unfortunately, the baseline model did not have as high of a confidence. This could be because the sentence being arranged in orientations such as “The best am I”, where something is saying that because it is a fact, or where it can be more borderline cased. The same can be said for the objective case. Since the sentence does not have opinion in the order it is presented, there is a low sense of subjectiveness predicted by the RNN and CNN. The baseline model once again was close to the borderline prediction, which makes sense since the sentence can be interpreted as “A course at UofT is MIE324”, which can be subjective if it was not true.

For the borderline cases, it is harder for the models to predict. In the first example, both baseline and CNN choose subjective with a high prediction value whereas RNN was 50%, which it should be. This is probably due to the nature of the sentence itself. Since the sentence is can be interpreted as someone’s opinion, it can be subjective. For the last example it was a toss up between both once again. This is probably due to the fact that the sentence can be objective since the person “Jonathan Rose” is being described, which can be objective in nature. The baseline model however was a bit more in the subjective side, which is also true since it can be one’s opinion.

Overall, all the results did make sense with some reasoning. Sentences of subjectivity were subjective and objective were objective. If one were to take a majority vote, it would lead to a correct answer since all cases, including the borderline cases, can be interpreted as the model suggests.

Scoring:

Baseline	2/2 for sub/obj 1/2 for borderline (should be close to 0.5)
RNN	2/2 for sub/obj 1/2 for borderline (should be close to 0.5)
CNN	2/2 for sub/obj 0/2 for borderline (should be close to 0.5)

RNN seems to have performed the best out of the three models, although baseline did perform quite well.