Allison Aprile
Assignment 4
April 30, 2021

Document Classification

Preprocessing Steps

For each of the three books, I first filtered the text to keep only that between the ***START OF PROJECT GUTENBERG … EBOOK *** and ***END OF PROJECT GUTENBERG … EBOOK *** tags. Then, I split the text into paragraphs to get individual samples. To remove the chapter headers, I removed any paragraphs that contained less than ten words. I then removed any newline and punctation characters, as well as normalized the text by lowercasing it. Finally, I tokenized the text. Each sequence was also padded to a length of 50, numerically encoded based on the vocabulary, and the corresponding label vectors were one-hot encoded.

I split the data into 80% training (12,774 samples), 10% validation (1,597 samples), and 10% testing (1,598). The testing set is imbalanced, with 949 paragraphs belonging to 'Austen', 473 belonging to 'Dostoyevsky', and 176 belonging to 'Doyle'. Overall, this led to a lower precision, recall, and F1-scores for the 'Doyle' texts in the classification reports.

I initialized the Embedding layer using the GLoVe 50d pretrained embeddings. These were set to trainable during training.

CNN

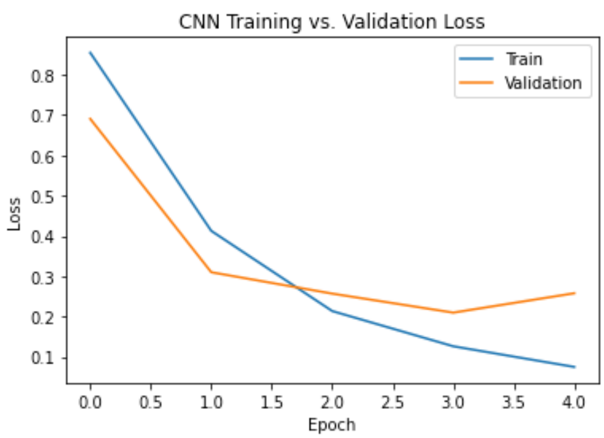
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_14 (Embedding)     (None, 50, 50)            944550    
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 50, 128)           19328     
_________________________________________________________________
max_pooling1d_9 (MaxPooling1 (None, 25, 128)           0         
_________________________________________________________________
flatten_9 (Flatten)          (None, 3200)              0         
_________________________________________________________________
dense_19 (Dense)             (None, 100)               320100    
_________________________________________________________________
dense_20 (Dense)             (None, 3)                 303       
=================================================================
Total params: 1,284,281
Trainable params: 1,284,281
Non-trainable params: 0
_________________________________________________________________
```

Allison Aprile
Assignment 4
April 30, 2021

Hyperparameters: Number of filters = 128, kernel size = 3x3, units in secondmost Dense layer = 100

> For the CNN, I tried 32, 64, and 128 filters, as well as 10 and 100 neurons in the first MLP Dense layers. 128 filters performed significantly better than the other filter sizes, which only reached a maximum of 64% validation accuracy. In effort with unfreezing the Embedding layer parameters, the CNN was able to reach approximately 90% validation accuracy.

Loss function: Categorical cross-entropy
Learning rate: 1e-3
Optimizer: Adam



Results:

	precision	recall	f1-score	support
austen	0.98	0.91	0.95	949
dostoyevsky	0.94	0.93	0.94	473
doyle	0.60	0.85	0.71	176
accuracy			0.91	1598
macro avg	0.84	0.90	0.86	1598
weighted avg	0.93	0.91	0.92	1598

Allison Aprile
Assignment 4
April 30, 2021

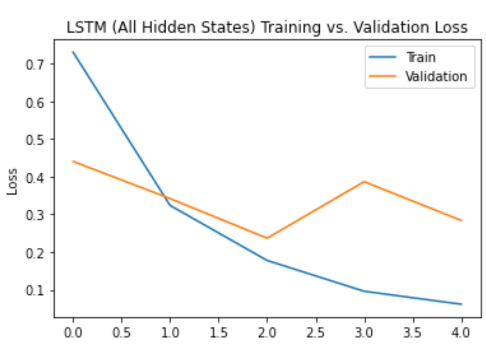
LSTM (All Hidden States)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 50, 50)            944550
_________________________________________________________________
bidirectional_7 (Bidirection (None, 50, 128)           58880
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 25, 128)           0
_________________________________________________________________
flatten_6 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_13 (Dense)             (None, 3)                 9603
=================================================================
Total params: 1,013,033
Trainable params: 1,013,033
Non-trainable params: 0
_________________________________________________________________
```

Hyperparameters: Hidden state size = 64

For the LSTM, I first built a baseline with a hidden state size of 32. Using this, I observed that taking the element-wise maximum (versus average) of all the states after the Bidirectional LSTM layer yielded better performance. I also observed that unfreezing the Embedding lyaer parameters led to better performance. After that, I tried increasing and decreasing the hidden state size. The baseline validation accuracy was 91% after 10 epochs. I tried sizes of 16, 32, 64, and 128. Decreasing the hidden state size to 16 reduced performance by approximately 5%, and increasing it to 64 increased the performance by 2%. Further increasing it to 128 kept the validation performance around 93% (equivalent to 64), but significantly increased computational cost. Therefore, I decided on using a hidden state size of 64.

Loss function: Categorical cross-entropy
Learning rate: 1e-3
Optimizer: Adam



Allison Aprile
Assignment 4
April 30, 2021

Results:

	precision	recall	f1-score	support
austen	0.94	0.99	0.96	949
dostoyevsky	0.98	0.88	0.93	473
doyle	0.78	0.77	0.77	176
accuracy			0.93	1598
macro avg	0.90	0.88	0.89	1598
weighted avg	0.93	0.93	0.93	1598

LSTM (Final Hidden State)

Layer (type)	Output Shape	Param #
embedding_11 (Embedding)	(None, 50, 50)	944550
bidirectional_8 (Bidirection	(None, 128)	58880
dense_14 (Dense)	(None, 3)	387

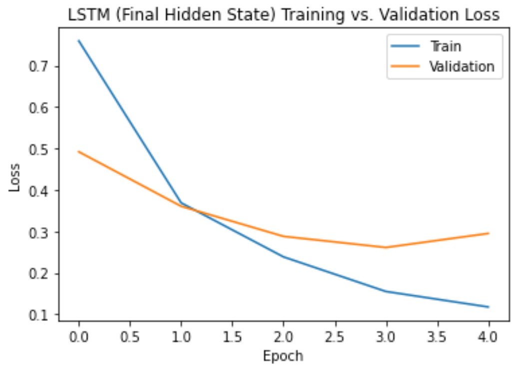
Total params: 1,003,817
Trainable params: 1,003,817
Non-trainable params: 0

Hyperparameters: Hidden state size = 64

For this LSTM, I followed the same tuning protocol as with the first LSTM. I observed the same optimal parameters as well - using a hidden state size of 64 and unfreezing the Embedding layers. This is expected because the last hidden state contains information about all the inputs and previous states; it is essentially a summary of the prior hidden states. The overall performance itself is similar as well, although this LSTM trained faster and has a few thousand less parameters.

Loss function: Categorical cross-entropy
Learning rate: 1e-3
Optimizer: Adam

Allison Aprile
Assignment 4
April 30, 2021



Results:

	precision	recall	f1-score	support
austen	0.91	0.99	0.95	949
dostoyevsky	0.94	0.93	0.93	473
doyle	0.91	0.48	0.63	176
accuracy			0.92	1598
macro avg	0.92	0.80	0.84	1598
weighted avg	0.92	0.92	0.91	1598

Allison Aprile
Assignment 4
April 30, 2021

Model Comparison

Based on the below table, it is clear that the CNN and the MLP had the best overall performance among the five models (based on F1-score, which is considerably an average/summary of precision and recall). As expected, the metrics for the 'Doyle' class are relatively lower than those of the other two classes because of the imbalanced dataset. Nonetheless, with all metrics significantly better than 'random guess', all of the models performed significantly well – despite the feature vectors being based on TD-IDF or the GLoVe pretrained word embeddings.

		Value		
	Metric	**F1-Score**	**Precision**	**Recall**
Model	**Class**			
CNN	**Austen**	0.95	0.98	0.91
	Dostoyevsky	0.94	0.94	0.93
	Doyle	0.71	0.60	0.85
LSTM (All Hidden States)	**Austen**	0.96	0.94	0.99
	Dostoyevsky	0.93	0.98	0.88
	Doyle	0.77	0.78	0.77
LSTM (Final Hidden State)	**Austen**	0.95	0.91	0.99
	Dostoyevsky	0.93	0.94	0.93
	Doyle	0.63	0.91	0.48
Logistic Regression	**Austen**	0.96	0.98	0.95
	Dostoyevsky	0.94	0.94	0.94
	Doyle	0.82	0.76	0.88
MLP	**Austen**	0.97	0.96	0.99
	Dostoyevsky	0.96	0.97	0.95
	Doyle	0.86	0.88	0.83