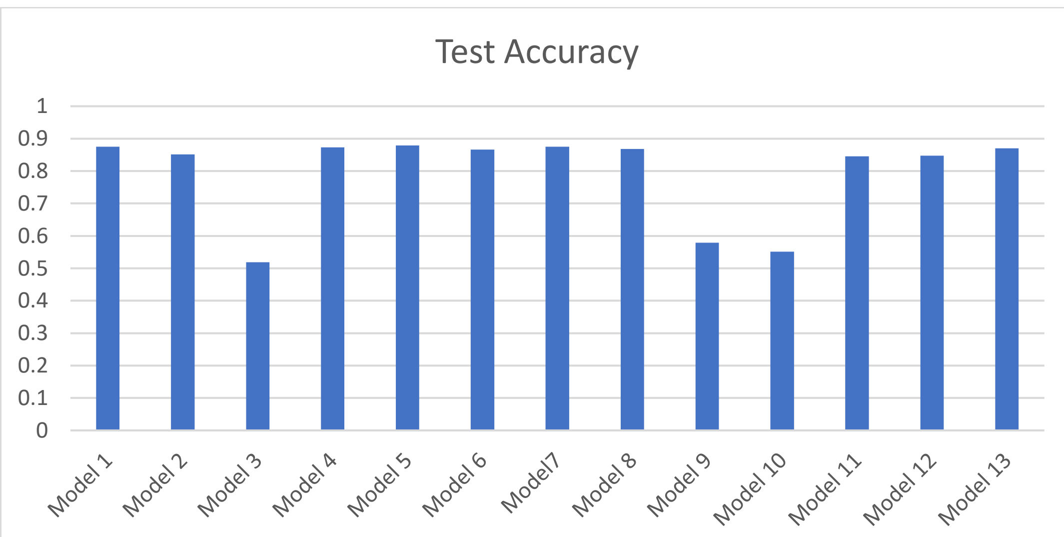
In our fourth assignment, we experimented with 13 different models to improve network performance, particularly when dealing with limited data. Techniques such as word reduction, limiting training and validation samples, and using various embedding layers were explored.

	Training sample	Validation sample	Words cutoff	Top words	Dropout	Embedding Layer	Embedding with masking	Pretrained Embedding Layer	Test Accuracy
Model 1	20000	5000	600	20000	No	No	No	No	0.875
Model 2	20000	5000	150	20000	No	No	No	No	0.851
Model 3	100	5000	600	20000	No	No	No	No	0.519
Model 4	25000	10000	600	20000	No	No	No	No	0.873
Model 5	20000	5000	600	10000	No	No	No	No	0.879
Model 6	20000	5000	600	20000	Yes	Yes	No	No	0.866
Model7	20000	5000	600	20000	Yes	Yes	Yes	No	0.875
Model 8	20000	5000	600	20000	Yes	No	No	Yes	0.868
Model 9	100	5000	600	20000	Yes	Yes	No	No	0.579
Model 10	100	5000	600	20000	Yes	No	No	Yes	0.551
Model 11	22500	2500	600	20000	No	No	No	No	0.846
Model 12	22500	2500	600	20000	Yes	Yes	No	No	0.848
Model 13	22500	2500	600	20000	Yes	No	No	Yes	0.87

Initially, utilizing a model from the book with 20,000 training samples, 5,000 validation samples, and 25,000 test samples yielded a quite high accuracy of 87.5%. When we reduced the word count to 150, as expected, accuracy dropped to 85.1%. Surprisingly, using the top 10,000 words resulted in an increased test accuracy of 87.9%, suggesting that these words contain more relevant information for sentiment analysis and excluding less frequent words might help mitigate noise in the model. When we decreased the training sample to 100 as instructed in the question, our test accuracy significantly fell to 51.9%. Introducing an embedding layer with 100 training samples improved accuracy slightly to 57.9%, and with a pretrained embedding layer, it further fell to 55.1%. One probable reason is that the pretrained embeddings might not align well with the limited and specific context of the smaller training sample.



With 20,000 training samples and various embedding techniques, test accuracies were 86.6% (embedding layer), 87.5% (embedding layer with masking), and 86.8% (pretrained embedding layer), respectively. Increasing the training sample to 22,500 saw test accuracies decrease: 86.1% for the base model, 84.7% with an embedding layer, and 87.4% with a pretrained embedding layer. The observed decline in accuracy with a larger training sample might be attributed to overfitting, evident from the decreasing validation accuracy.

Therefore, we can conclude, finding the right balance of training samples and optimal word count is crucial for optimal model performance, helping to avoid chances of overfitting and noise and focus on informative features.

PS: I took help from AI tools to code the splitting part of the training and validation data set.