NLP Track : Sequence Modelling

Siddharth Anil

June 2025

1 My Approach

During the week 2 of CSOC, I was tasked first with selecting a learning trackamong Reinforcement Learning (RL), Natural Language Processing (NLP), and Computer Vision (CV). The choice for me was actually between CV and NLP since I was sure i ain't ready for what's inside that pdf of RL.

Initially, I chose CV first,thinking it would mainly involve implementing a CNN from scratch something I felt reasonably comfortable with. However, after completing the entire CV code, I quickly realized that the dataset was extremely large. The model took hours to train, with no clear end in sight. so i had to swap to NLP track, when i asked some of the others who took CV, They also felt like we should switch the track to NLP since it seems way more easy with a less troublesome dataset.(However i found issues of the dataset being too large, problematic to run even on the NLP task).

And so it was decided: NLP it is. And so i began watching campusx again. This time on naive bayes, RNN, Bag of words and so on.

Since i was given the Liberty to use any library, i first experimented on the dataset with just pandas,numpy,re libraries. This was my first attempt at solving the problem. I thought this cound strengthen my fundamentals and core concepts. In the end it took a lot of training time and the training was going on from morning to noon. The model was slow and i knew i had wasted a lot more time. Since now i found out i had to go for a better performing model i searched which library is best at performance on NLP tasks and Google showed me keras library to work on. I had very less time and so i went through github repositories and kaggle notebooks to understand how to use keras and to implement it on amazon reviews dataset. My final model in keras took only 50 lines of code and gets everything done with an accuracy of almost 0.9/1 and about the same f1 score. I am submitting both my models - first one hindered by low performance and second one optimised to work smoothly

I want to sincerely apologize to the seniors if my project felt rushed or incomplete. It wasnt due to a lack of interest or effort, but rather due to the constraints of time and computational resources. I genuinely enjoyed the learning experience and hope to deliver better results in the future with more preparation and time.

2 Cleaning data

Even in my second attempt with tensorflow too, my model was taking a significantly large time to train and it was on the submission date that is today that i realized maybe changing the size of training data could help in reducing training time and it did. So when i loaded the dataset i added a command to take a sample of 100000 from the training dataset to train the model. That worked. I strongly believe if a stronger performing pc is given the model with all the data from the dataframe, it could show more accuracy and hence show better results. The title and text columns as i thought needed to be merged together so that it would be easier to do the cleaning, padding works. Either i could have deleted the title column but i show some where title proved to be more relevant as the text could be said in a sarcastic tone.

In the last week task, A lot of problems were caused by imbalanced data so I had done an analysis on the number of positive and negative reviews using seaborn and matplotlib. They were almost equal in this case

I used the re library. In my earlier model, i tried to implement nltk but it caused performance issues and so i halted from using it in this model. By defining a function for cleaning text which converted all text to lowercase alphabets, remove punctuation marks, special characters, digits and replaces extra space with a single space. I also thought of the possibility of emojis in reviews but i had no clue on how to use them to predict the sentiment. I applied loop for all text in the column apply the function and was able to clean the review text. While running the loop i encountered an error which was caused by missing values in the training data and I resolved it by drop duplicates function. Since the dataset was too large i thought deleting the duplicate or missing value columns was the best option. The output column also needed a little tuning as it was 1 for negative and 2 for positive and so i converted it to binary format.

After that i was able to clearly define the X train, y train and X Text, y test.

3 Padding and Truncating

I imported tokenizer and pad Sequences from keras utilities since it is necessary to turn the raw text into sequences of integers and pad/truncate them to equal length. I initialized vocabulary size to 10000 which results in keeping of the 10000 most frequent words in the tokenizer. I kept it to 10k to limit memory usage. In keras documentation of tokenizer it needed mapping of rest of the words to ¡00V¿.

The next step was to convert text to integers based on the vocabulary index defined. This is acheived by tokenizer.text to sequences.

Padding and truncation were done which ensures all sequences are of equal length,in this case 100. Longer reviews truncated to avoid errors in model.

4 Model Training

Then i made a Keras model with a SimpleRNN layer and a Dense layer which is the hidden layer of the model with relu activation function. The final output layer activated by sigmoid function predicts the sentiment of the review. Sigmoid function is used since it is best to implement in binary classification problems as was seen in the previous task. I used same loss function since it was the most desirable for text based classification problems and i experimented the optimizer with adam and SDG, adam gave lower losses and better accuracy. And i trained the model with the sample dataset i created from the large dataset to create a model of almost 90 percent accuracy.