CS779 Competition: Sentiment Analysis

Shivangi Singh
200943
{shivangis20}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

16th April 2023

Abstract

Sentiment analysis is a potent tool with varied applications across industries. It is helpful for social media and brand monitoring, customer support and feedback analysis, market research, etc.Sentiment Analysis is an NLP application that identifies a text corpus's emotional or sentimental tone or opinion. Usually, emotions or attitudes towards a topic can be positive, negative, or neutral.This makes sentiment analysis a text classification task. Different models were tried and tested such as LSTM, Bidirectional LSTM , but finally RNN was employed to do the sentimental analysis in our case.

1 Competition Result

Codalab Username: S_200943
Final leaderboard rank on the test set: 39
F1 Score wrt to the best rank: 0.63

2 Problem Description

Given a sentence, we were required to automatically predict the sentiment expressed in the sentence, i.e., assign one of the three labels (positive, neutral and negative) to the sentence.We have studied different types of neural models, e.g.,RNN based models, sequence-to-sequence models, transformer models.We were supposed to use these to implement a system for predicting the sentiment of a given sentence. There 3 main sentiment classes that need to be assigned to a sentence: Positive, Neutral and Negative. It requires us to implement a deep learning based sentiment prediction system by implementing the architectures like RNNs, Transformers etc..

.

3 Data Analysis

3.1 Train Data

The train dataset was provided to us in a CSV File that contained three columns of text id , sentence and gold label(sentiment). The dataset includes 92,228 English sentence. We would not want the stopwords to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that we consider to stop words. NLTK(Natural Language Toolkit) in python has a list of stopwords[1] stored in 16 different languages.The next property of the sentence is the sentence length. It was observed that most of the sentences have length less than 15.

[1]https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

3.2 Test Data

We were also provided with the test data for the actual translation. It consisted of 5110 English sentences. We applied the same procedure to preprocess it and then used it for our analysis.

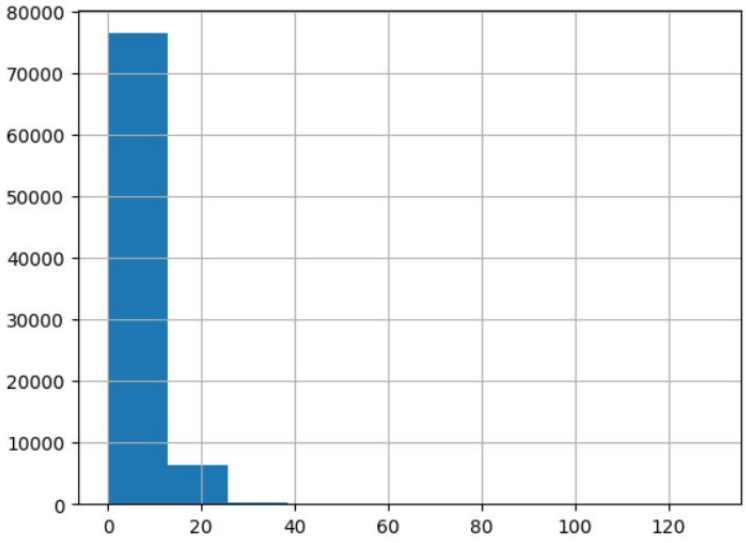

Figure 1: Histogram showing the length of sentences in train data

4 Model Description

For sentiment analysis,the fundamental architectures that come to our minds are different RNN Architecture like Bidirectional RNN, Multi layer Architecture and LSTMs.Like many deep learning models, LSTM models are susceptible to overfitting if they are not properly regularized. This was one of the problem that was observed when we used this model.The RNN_Text Model that has been used is a simpler and effective model which is easier to train and works well for the short sentence length data like in our training dataset as it is able to capture the contextual information and temporal dependencies between words in a sentence more effectively than simpler models like bag-of-words approaches.

5 Experiments

1. **Data Pre-processing :** As we observed in the data analysis, given data contains some unnecessary stopwords, so we cleaned the data tockenised it , built a dictionary of as many as 33755 words containing the mostly occured words and also apply some basic pre-processing on the data to make it ready for further processing. First of all, we have filtered the data on the sentence length. We select a specific value of Max sentence length and only processed these sentences whose length was less than the Max length as they were padded with zeroes.
2. **Training procedure:** The Optimizer used is Adam optimzer. SGD updates all parameters with the same learning rate and choosing this learning rate can be tricky. Adam adapts the learning rate for each parameter, giving parameters that are updated more frequently lower learning rates and parameters that are updated infrequently higher learning rates.Another change that we have made here is the loss function(also known as the criterion).We learnt that CrossEntropyLoss is used when our examples exclusively belong to one of C classes, but BCEWithLogitsLoss is used when our examples exclusively belong to only 2 classes (0 and 1) and is also used in the case where our examples belong to between 0 and C classes (aka multilabel classification). Thus, the latter was used.
3. **Hyperparameters:** The hyperparameters for the test phase are give in the table below

Hyperparameter	Test phase
Embedding size	256
Hidden size	1024
Number of layers	2
Output dimension	3
Batch size	50

6 Results

1. The submission made in the dev data in the devlopment phase fetched 0.116 F1 score which was using the LSTM model. Although the LSTM model is regarded better in such scenarios,due to some unresolved errors and such low F1 score, I decided to switch my model to something simpler.
2. It was during the test phase when the RNN_Text model was used we got the F1 score to be 0.304 and further tuning the hyperparameters was not producing any significant results.

7 Error Analysis

Although the RNN_text model can be computationally expensive to train, particularly if we are working with a large dataset.The RNN_Text model is able to capture the contextual information and temporal dependencies between words in a sentence more effectively. This allows the model to generate more accurate predictions and produce more meaningful representations of the underlying data.Due to the technical constraints we could not run more than 5 epochs which has affected on our score and gives us the scope that it could have been better.However the model does converge signifying that the parameters were tuned effectively in the given case.

8 Conclusion

Overall, the RNN_Text model is a powerful and versatile tool for processing and analyzing sequential data, particularly in the domain of NLP. By carefully considering the strengths and limitations of the model, we can effectively leverage its capabilities to address even the multi-class sentimental analysis tasks.

References