# Project Overview

The problem of classifying sentiment in Twitter documents (tweets) presents an interesting and difficult challenge because of the very short length of the documents in the corpus. Each document in the corpus is limited to 140 characters. Such a small number of tokens in a document may reduce the accuracy of any attempt at sentiment classification.

Various types of neural networks - sequential and non-sequential - were tested. The performance of the models was measured using the standard metrics of accuracy and F1 score.

The following Colab notebook contains the code which performed this analysis”

<https://colab.research.google.com/drive/1Spspi8rktRHVpaP7wVrhsrrin0XXnHcM> .

# Literature Review

Previous attempts to classify sentiment in Twitter documents have typically used traditional machine learning techniques. Go, Bhayani, & Huang (2009) used Naïve Bayes (NB) and support vector machine (SVM) classifiers, as did Hasan, Moin, Karim, & Shamshirband (2018). Rahman, Sarma, Sinha, Sinha, & Pradhan (2018) employed similar techniques and also included linear regression among their attempts at classification. The research of Shah (2018) included random forest along with NB and SVM.

By contrast, Saif, He, Fernandez, & Alani (2016) used a lexicon-based approached that did not require training data at all. Instead, they relied upon the pre-labelled sentiments of a lexicon of words to classify the documents.

Research involving the use of deep neural networks to classify a Twitter corpus appears to be limited or unpublished.

# Data

The corpus used for this project was taken from a Kaggle dataset that had been created specifically for Twitter sentiment analysis (<https://www.kaggle.com/kazanova/sentiment140> ). It consists of 1.6 million tweets from that have been pre-labelled as either positive or negative in sentiment. The dataset was originally constructed for the Stanford NLP course CS224n (Go, Bhayani, & Huang, 2009).

# Methodology

Various models were tested from the open source Kashgari NLP framework (<https://kashgari.bmio.net/> ). The models tested were a bi-directional LSTM (BiLSTM), a bi-directional Gated Recurrent Unit (BiGRU), a convolutional neural network (CNN), a combined CNN/LSTM (CNN\_LSTM) and a combined CNN/GRU (CNN\_GRU).

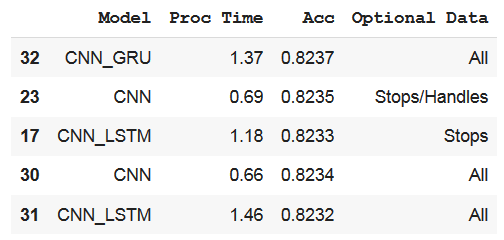
In addition, Kashgari allows for any of its models to be run with pre-trained BERT embeddings. These embeddings were created by the Google AI team, based upon the BERT model created by Devlin, Chang, Lee & Toutanova (2019). Thus, two more models using the BERT embeddings - the BiGRU and the CNN\_LSTM - were tested.

The corpus was randomly split into a training, validation and test set using the standard proportions of 60%/20%/20%. The test set was used to score the performance of the various models using the measures of accuracy and F1 score. Some minor cleaning of the text was performed by tokenizing it and removing most punctuation.

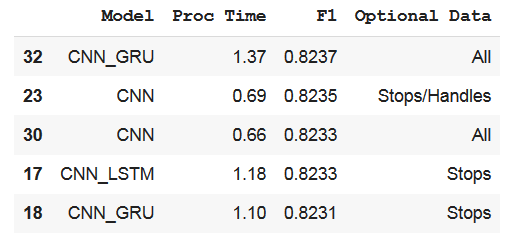
Each of the models was tested while including/excluding certain types of tokens. The models were tested with various combinations of standard English stop words, Twitter handles (which start with “@”), the standalone character “@” (which can used to represent the word “at”) and some basic emoticons that otherwise would have been considered punctuation and removed during preprocessing.

# Results

A summary of the five best performing models in terms of accuracy and F1 score is shown below in Figures 1 and 2. Among the best models, the accuracy and F1 scores were nearly identical. This is in part due to the fact that the two classes (positive sentiment and negative sentiment) were almost perfectly balanced. It also means, though, that the models correctly classified the two types of sentiment with nearly the same accuracy. None of the models did a better job of classifying one type over the other.

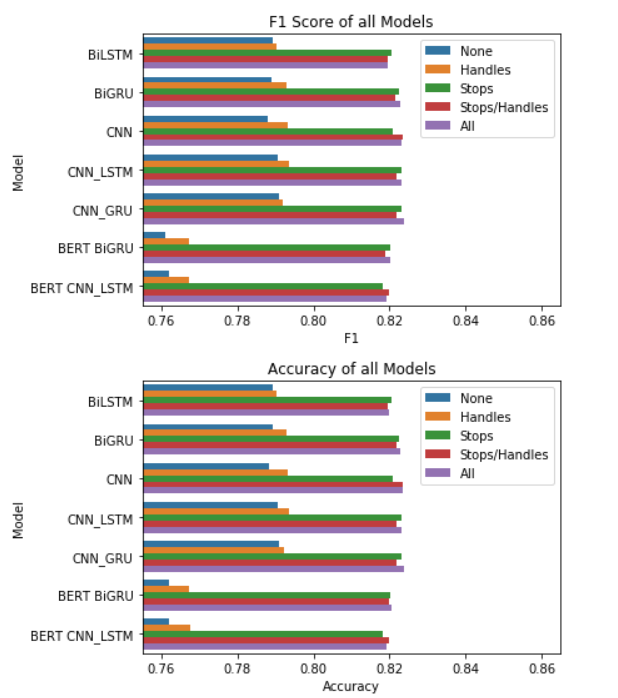


**Figure 1. Best performing models by accuracy.**



**Figure 2. Best performing models by F1 score.**

Performance of all of the models by F1 score and by accuracy is shown in graphical form in Figure 3 below.



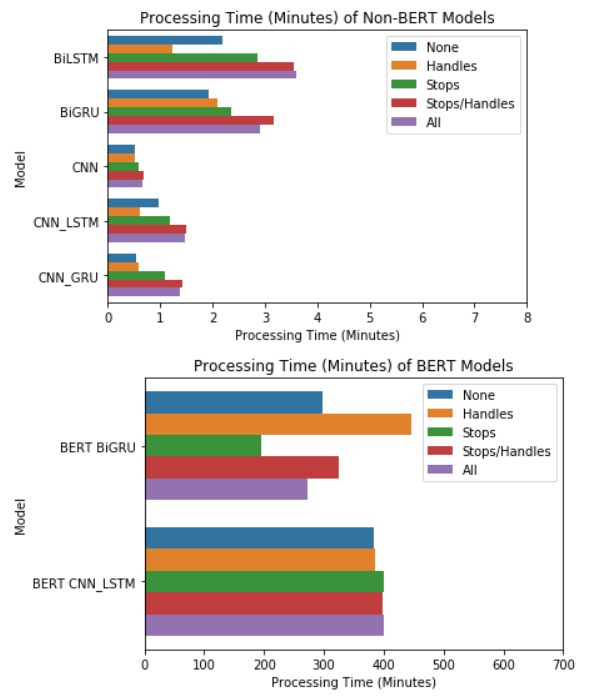
**Figure 3. Graphical representation of the performance results of all models.**

The top performer was the combined CNN\_GRU model. All of the best models contained a CNN at least as one component. The standalone CNN model, in fact, was very nearly the equal of the CNN\_GRU. This clearly shows that a CNN is very effective, and in fact was the single most effective of the standalone models, at classifying sentiment in this Twitter corpus. The CNN does a good job of discovering features from these very small Twitter “documents”. That small size may also explain why memory in terms of LSTM or GRU does not add much if anything in terms of the performance on top of the CNN. Sequential models may indeed have some difficulty when the sequences are very short.

It is also interesting to note that the models that used pre-trained BERT embeddings actually underperformed those that did not. It may be that these tweets are simply too short for BERT to be of use. Another possibility is that the corpus that the BERT embeddings were not well suited for this task because the corpus that they were trained on is dissimilar from this corpus.

Shown below in Figure 4 are graphs of processing time (in minutes) of all models. Note that the graphs for the BERT models and the graph for the non-BERT models are on vastly different scales.

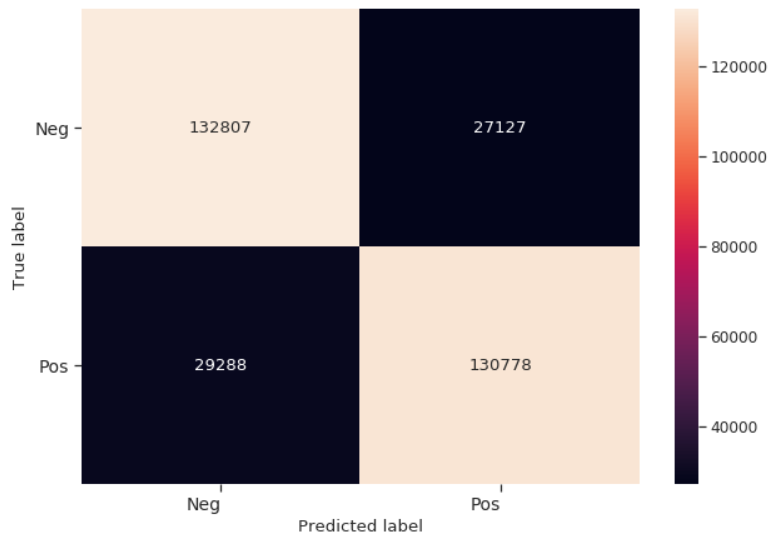
Not only did the BERT models underperform, they were also much slower. Part of this discrepancy is due to the number of epochs required for each model to converge. The BERT models generally needed more epochs. Mostly, though, this is a result of each epoch of the BERT models simply being much slower to train. In short, for this particular task, the use of the BERT embeddings was counterproductive.



**Figure 4 Processing time (in minutes) of all models. Note the different scales for the BERT models vs. the non-BERT models.**

# Evaluation of Results

The results of the best-performing model (CNN\_GRU using all available tokens) were examined. Figure 5 below shows the confusion matrix of its classifications. The model performed only slightly better on positive than negative tweets.



**Figure 5. Confusion matrix of the results of the best performing model.**

To investigate these results further, for each token in the training corpus, a sentiment value was created. In addition to single word terms, word bigrams, trigrams and quadrigrams were also included in the analysis.

This value is a modified document frequency. It is simply a count of how many positive-labelled tweets in the training set contain the term, less the number of negative-labelled tweets that contain it. The value can be positive, negative or zero. Multiple occurrences in a tweet simply count as one occurrence. This is not a term frequency. The counts are adjusted to deal with the small class imbalance (there are slightly more positive tweets in the training set than negative ones). Thus, the scores are not integers.

Table 1 below shows the 20 highest scoring (most positive) and the 20 lowest scoring (most negative) terms in the training corpus.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Term** | **Score** |  |  | **Term** | **Score** |
| **1** | **you** | **16159.52** |  | **1** | **i** | **-15715.51** |
| **2** | **good** | **7363.60** |  | **2** | **my** | **-7771.94** |
| **3** | **love** | **6559.13** |  | **3** | **not** | **-6831.74** |
| **4** | **the** | **6513.44** |  | **4** | **but** | **-5587.48** |
| **5** | **a** | **6500.72** |  | **5** | **no** | **-5058.37** |
| **6** | **thanks** | **6059.11** |  | **6** | **sad** | **-4800.26** |
| **7** | **for** | **5804.88** |  | **7** | **miss** | **-4439.21** |
| **8** | **http** | **5276.73** |  | **8** | **work** | **-4200.54** |
| **9** | **your** | **5217.53** |  | **9** | **wish** | **-2986.56** |
| **10** | **with** | **4210.55** |  | **10** | **bad** | **-2884.91** |
| **11** | **com** | **4156.49** |  | **11** | **want** | **-2861.24** |
| **12** | **great** | **3668.07** |  | **12** | **can't** | **-2838.77** |
| **13** | **i love** | **3450.16** |  | **13** | **i miss** | **-2777.41** |
| **14** | **lol** | **2963.70** |  | **14** | **sorry** | **-2714.02** |
| **15** | **happy** | **2953.14** |  | **15** | **hate** | **-2640.15** |
| **16** | **thank** | **2815.53** |  | **16** | **i have** | **-2583.97** |
| **17** | **and** | **2777.45** |  | **17** | **don't** | **-2441.41** |
| **18** | **new** | **2663.92** |  | **18** | **still** | **-2420.92** |
| **19** | **are** | **2614.37** |  | **19** | **go** | **-2405.14** |
| **20** | **haha** | **2601.88** |  | **20** | **sick** | **-2352.95** |

**Table 1. The 20 highest and lowest sentiment scoring terms in Twitter corpus.**

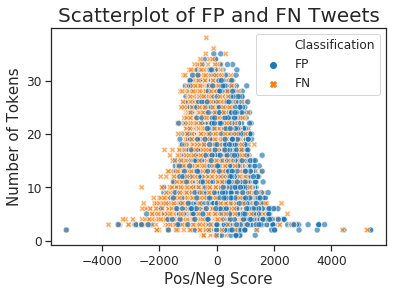
The list of top scoring positive terms includes some that are to be expected, such as “good”, “love”, “thanks”, “happy”, etc. Likewise, the list of most negative terms unsurprisingly includes “not”, “but”, “no”, “sad”, “hate”, etc. What may not have been expected, though, is that the most positive term is “you” and the most negative are “i” and “my”. This is very interesting because it suggests that negative sentiment on Twitter is strongly associated with narcissism. Positive sentiment on the other hand may perhaps be associated with generosity or empathy, indicated by the presence of the word “you” as most positive in the corpus.

As previously mentioned, the inclusion of common stop words improves model performance considerably. This is reflected in the positively-labelled words in Table 1. The terms “the”, “a”, “and”, “for” and “are” are all among the top 20 terms in positive tweets. Perhaps this suggests that negative tweets are written with less regard for formal sentence structure than are positive ones.

One other curiosity in the list of positive terms in the presence of “http” and “com”. They may show an unexpected relation of the presence of a URL and positive sentiment in a tweet. The scores for both of those terms are too high to be coincidental.

The next question to be considered is why more than 17% of the tweets were misclassified, even by the best model. To provide some insight, two characteristics of the misclassified tweets were examined: the length of the tweet (number of tokens) and the total positive/negative score of the tweet. The latter metric was created simply by summing the individual scores of each term in the tweet. This includes the scores for all word bigrams, trigrams and quadrigrams. The score was then normalized by dividing by the total number of terms in the tweet so that longer tweets would not naturally have higher or lower scores.

Grouped by false positive classifications and false negative ones, a scatterplot of total positive/negative score versus its length was created and is shown in Figure 6.



**Figure 6. Scatterplot of false positive and false negative tweets.**

This plot does not show any apparent trend with length in the relative number of false positives and negatives. Along the axis of positive/negative score, however, the pattern is clear. A sizeable portion of tweets labelled as negative are misclassified because of a preponderance of positive terms. Likewise, many positive-labelled tweets are misclassified because of negative terms in the tweet. The model is being trained to look for positive terms in positive tweets and likewise negative terms in negative tweets. That explains much of the misclassification and it is perhaps the CNN at work, extracting features without much regard for sequence.

# Summary

The problem of classifying sentiment in Twitter documents was researched. Various types of neural networks were tested. This included both sequential and non-sequential models. The models were tested with various subsets of the tokens in tweets, and two models were tested using pre-trained BERT embeddings. The performance of the models was measured with the standard metrics of accuracy and F1 score.

The results showed that the very best performance was produced by a combined CNN\_GRU model, but that all of the best models contained a CNN element. The standalone CNN model itself very nearly equaled the performance of the top model. In the case of this corpus, it would seem that a CNN is very effective at discerning features within the tweets and that sequential models do not add much if anything to the performance of classification, possibly due to the very small sequences in each tweet.

An examination of the terms in tweets labelled positive versus negative reveals some items of interest. Negative tweets appear to be strongly correlated with narcissism and conversely, positive tweets with empathy or generosity. Also, many of the common English stop words are associated with positive sentiment.

Among those tweets that were misclassified, evidently a large portion of them were classified on the bases of the positive/negative score of their tokens. This also suggests that in such short documents, context or sequence is less important than the tokens themselves.

# Future Directions

In the future, the classification of tweets could be attempted with Word2Vec embeddings but it is unclear if this would result in any increase in accuracy. The number of tokens in any of the tweets is so small that each token in a tweet would share similar context and thus similar embeddings. It is likely that the methods that were used here would produce results comparable or superior to those using Word2Vec.

Another area to explore would be the pre-training of BERT embeddings. The embeddings used for this project were trained by the Google AI team on Wikipedia and a corpus of books. If they had been pre-trained on a corpus of Twitter documents and then re-trained and tested on the same corpus used in this project, the results would have been interesting to observe.References

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from: arXiv:1810.04805v2

Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine Learning-Based Sentiment Analysis for Twitter Accounts. *Mathematical and Computational Applications*, 23(1), 11.

Go, A., Bhayani, R. and Huang, L. (2009). Twitter Sentiment Classification Using Distant Supervision*. CS224N Project Report, Stanford*, 1(12).

Rahman, E., Sarma, R., Sinha, R., Sinha, P., & Pradhan, A. (2018). A Survey on Twitter Sentiment Analysis. *International Journal of Computer Sciences and Engineering*, 6(11), 644-648.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing and Management*, 52(1), 5-19.

Shah, H. (2018). Twitter Sentiment Analysis. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(12), 15.