Deep Learning Based Sentiment Analysis On Twitter Data

Xiangyu Hu, Xin Hu, Qian Yi, Jingyu Zhang

Abstract

In this paper, we utilized several deep learning methodologies to complete sentiment analysis of Twitter data. The purpose of this project is to automatically predict the sentiment for each tweet by reading in a large amount of tweets. The application is broad, such as understanding product reviews, managing customer support, and monitoring social media. We focused on coronavirus related tweets here to analyze individual emotions. A quick overview about what we have done so far to solve this problem. First, we apply NLP techniques to clean text data, including normalization, stop words and punctuation removal, stemming, lemmatization, and tokenization. Secondly, we build up the logistic regression as the baseline multiclass classification model. Finally, we introduce deep learning methodologies to improve the model performance, such as CNN, LSTM, Bi-LSTM, GRU, and BERT. Overall, the BERT model provides the highest accuracy of 0.91 on the test dataset.

Introduction

Text classification (TC) is one of the important tasks of machine learning and has been extensively used in the several areas of Natural Language Processing (NLP). It's objective to design appropriate algorithms to allow computers to extract features and classify texts automatically. In this project, we construct several deep learning models to understand people's attitudes towards Coronavirus by reading in a large amount of data. It's worth to mention that the deep learning models could also be used in other Natural language processing problems, including words semantic classification, question answering, and dialogue management. [1]

There are three reasons we decided on this topic. First, the coronavirus has created the biggest global crisis in generations, sending shock waves through health systems, economies, and societies around the world. Next, different people have different attitudes and would make different behaviors. The last one is that the government should make corresponding responses by understanding people. And sentiment analysis provides answers into it, and beyond this, it can be automated, decisions can be made based on a significant amount of data rather than plain intuition that isn't always right.

Dataset

The dataset is from Kaggle, which contains nearly 45 thousand tweets with sentiment taggings and split to train data and test data. The data[2] is text data, and contains 6 main features: Username, ScreenName, Location, Twitter at(time), Original twitter, Sentiment.

The target variable contains 5 categories: extremely negative, negative, neutral, positive, and extremely positive, these are the examples for positive twitter and negative twitter:

Positive:
1. 'Based on a grocery store visit, it seems that Norwegians are dealing with #Covid_19 by stockpiling tomato soup and eggs. Which makes perfect sense, since Norwegians put an egg in their tomato soup.'

2. 'How things shift in a few hrs! Things get real in T.O. as schools to shut down, big events cancelled. and my brother just called from the local grocery store where there are massive line ups & frayed tempers Take care all, and let's do what we can to help each other! #Covid_19'

Negative:
1.'If 1,000,000 get infected with a 2% death rate for the #coronavirus then 20,000 will die in USA.If you dont have mo mney to stock up food, then find a fishing hole and supplies to hunt and fish!'

2. 'All Americans should be concerned w/ stock markets. LIQUIDITY & CREDIT. Lack of liquidity can cause a credit crisis & everything will freeze. -No medicine -No food -No supplies -Empty stores. #CoronaVirusUpdate #coronavirus @realDonaldTrump @SpeakerPelosi'

So that we are going to build a model to analyze and bucket each tweet. To be more practical, we transfer the tag into three categories, positive. neural and negative.

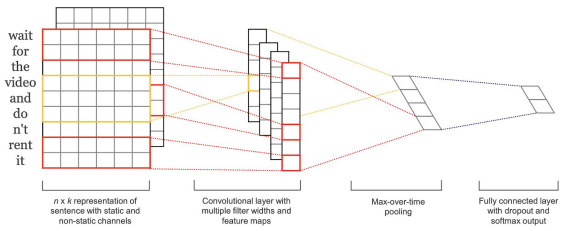
Modeling

Logistic Regression

Our baseline model is logistic regression, which is not a deep learning model. We build this baseline model so that we can compare its results with other deep learning models to understand the advantages or disadvantages of CNN, LSTM, Bi-LSTM, GRU and BERT approaches. We use "CountVectorizer" to perform word-level tokenization while removing stop words and representing twitter posts as a sparse matrix. With 80% of training data and 20% of testing data, surprisingly, the logistic regression model did a good job. The accuracy of logistic regression reached 82.20% with F1 score of 0.82.

CNN

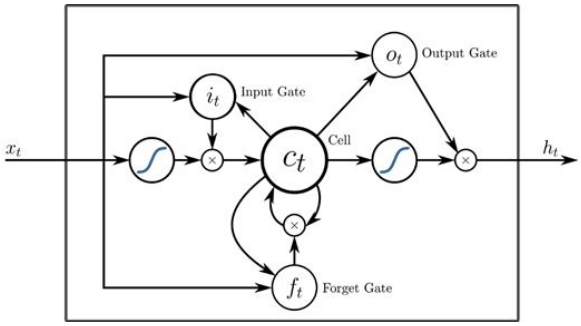
Convolutional Neural Network (CNN) is a class of deep neural networks, most commonly applied to image classification. The biggest reason to adopt CNN in image analysis and classification is because CNN can extract an area of features from global information, and it is able to consider the relationship among these features.[3] Recently it has been utilized on NLP tasks as well. An example of CNN model structure is shown below.



(figure 1)[4]

LSTM

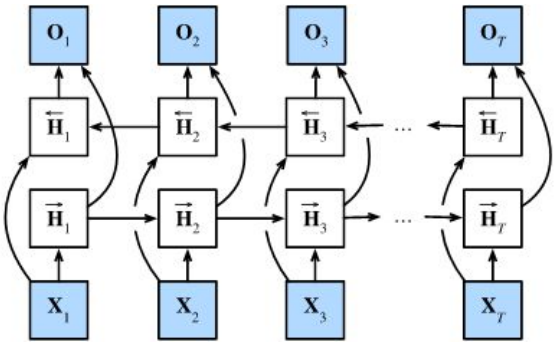
Long Short-Term Memory networks are a special kind of RNN model, unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). This model was first introduced by Hochreiter & Schmidhuber in 1997 [5], and has been improved much by many Scientists after many following works. And this figure shows how LSTM model work:



(figure 2)[6]

Bi-LSTM

A Bi-LSTM model is Bidirectional Long Short-term Memory Model, which is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. It can help the context become more available to the algorithm, and increase the available amount of information to the neural networks. The model is the combination of Bi-RNN and LSTM model. In 1997, Scheuster and Mike introduced the Bidirectional Recurrent Neural Networks[7], and the following researchers effectively combined it with the LSTM model in study of many areas. And this graph shows how the Bi-LSTM model works:

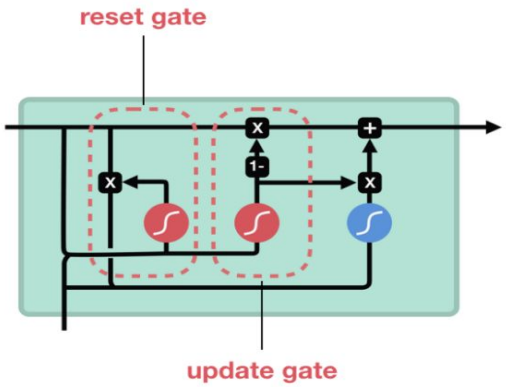


(figure 3)[8]

GRU

Gated recurrent units (**GRU**s) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho et al.[9] The GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. GRU's performance on certain tasks of polyphonic music modeling, speech signal modeling and natural language processing was found to be similar to that of LSTM. GRUs have been shown to exhibit better performance on certain smaller and less frequent datasets.

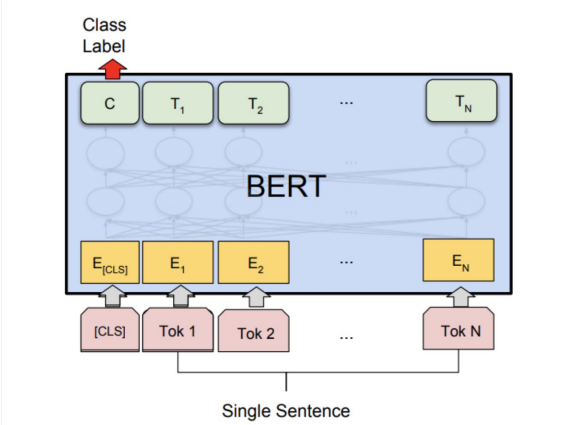
In the training process, due to the simplicity of this dataset we used, one layer GRU has been constructed.



(figure 4)[10]

BERT

The last model we try is the Bidirectional Encoder Representations from Transformers (BERT) model. The BERT model was developed by Google in 2018.[11] This state-of-art model could be fine-tuned with a small number of text data and achieve substantial accuracy in NLP tasks like sentiment analysis.[12]



(figure 5)[14]

The model is originally trained with English wikipedia and book corpus and it has 110 million parameters with powerful contextualized embedding capabilities [13] To fine-tune the model, we downloaded the BERT python package and chose a small learning rate of 1e-4 with an epsilon of 1e-7 for numerical stability. By fine-tuning BERT for sentiment analysis, we obtained more than 90% accuracy score, which is the best performance among all of our models. However, this model is computationally expensive and requires GPUs to run it. For student accounts, we are only able to run the model for limited epoches.

Results

Model	Accuracy	F-1 Score
Baseline: Logistic Regression	0.8183	0.8200
Word-level CNN	0.6630	0.6876
Character-level CNN	0.7578	0.7696
GRU	0.8465	0.8659
LSTM	0.8526	0.8700
Bi-LSTM	0.8530	0.8000
BERT	0.9065	0.9074

(table 1)

After fitting all these models, we compare them with accuracy and F-1 score.

The reason we use the F-1 score is that the F-1 score takes both false positive and false negative into account. Also, in most

real-life classification problems, imbalanced class distribution exists and thus F1-score is a better metric to evaluate our model on.
We applied word level and character level CNN models here, but both of the two had lower accuracy than the baseline model. The reason behind this is that CNN is more suitable for image classification. And this dataset may not be good for it to run CNN.

For the LSTM model, we get an 85% accuracy and 0.87 F-1 score, which is relatively high, and one of the advantages of the LSTM model is that it can have a good performance in text data. However, the drawback of this model is that it can be easily affected by the random initial weights, small weight initializations will be good for this model. Besides, although the LSTM model can help to solve the problem of vanishing gradients, it can not remove it completely. Thus, the cells will become more and more complex with more features.

Bi-LSTM model is the model we try to improve the LSTM model, the difference between it and LSTM is that it will run the input two ways, one in forward direction and one works back forward, it can help the model to understand the data better, while in this project,the result of this model has an 85.3% accuracy and 0.8 F-1 score, since the baseline model has already got a good performance, the improvement of this model is not much, and it cost more time since it doubles the layers.

By constructing one layer GRU model, the accuracy we got is 84.65% and the F-1 score is 0.8659. GRU is pretty similar to LSTM, but there are a few important differences. First, there is no hidden state. Second, the processes of determining what the cell states forgets and what part of the cell state is written to are consolidated into a single gate. The last one is the entire cell state is given as an output. All of these changes together provide a simpler design with less parameters than the LSTM. Less parameters, however, may come at the cost of decreased expressibility. The results from LSTM and GRU also reflect the similarity between these two models.

For the BERT model, we got a 90.65% accuracy and 0.9074 F-1 score, which is the best performance among all the models we tried. The BERT model did an excellent job in word embedding with the vector length of 768. This captured the most important contextual information from training data. However, while the model complexity brings high accuracy and F-1, the drawback is the computational expensiveness of the model. It took a very long time to run the model by using GPUs.

Discussion & Conclusion

For all the models we tried, the based line model is logistic regression, which is not a deep learning or neural network model. Surprisingly, the baseline model did a very good job in this sentiment analysis. The performance is even better than CNN model. The reason might be that the task itself is simple enough and the training data is large enough.

By comparing all the models, the BERT model has best performance with accuracy

and F-1 score higher than 90 percent. However, this model is computationally expensive. One major constraint we have is that Colab limited student accounts to use GPU. So we are unable to train more epoch and get higher accuracy for free.

We also generated some ideas for future works. For this project, we only include cleaned twitter text data as our input. Through our discussion, we agree that other information might also be important features of this sentiment analysis.

First, the time component is a great indicator. People's attitudes and responses towards Covid-19 might change overtime. For example, in January 2020, while not so many Covid cases were detected, people may not hold very negative sentiment on Twitter posts. However, in March 2020, the Covid cases increased exponentially, then more people may hold negative attitudes. Second, the hashtag (#) and at (@) signals may be directly related to specific topics and people.

Adding the topics into our model may also increase the robustness of the model. Third, the locations/countries of the Twitter users could also be included in the analysis. For example, for users who live in New York, where many Covid cases were detected, people may have greater possibilities to leave negative posts than people who live in other relatively safer cities. Based on these discussions, we will keep exploring in the future to improve the model.

Reference

[1] Zulqarnain, M., Ghazali, R., Hassim, Y. and Rehan, M., 2020. A comparative review on deep learning models for text classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(1), p.325.

[2] Kaggle.com. 2020. *Coronavirus Tweets NLP - Text Classification*. [online] Available at:<https://www.kaggle.com/datatattle/covid-19-nlp-text-classification>.

[3] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, *CNN for situations understanding based on sentiment analysis of twitter data*, *Redirecting*. [Online]. Available: https://doi.org/10.1016/j.procs.2017.06.037.

[4] Kim, Y., n.d. *Convolutional Neural Networks For Sentence Classification*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1408.5882.pdf>

[5] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.

[6] D2l.ai. 2020. *9.4. Bidirectional Recurrent Neural Networks — Dive Into Deep Learning 0.15.1 Documentation*. [online] Available at: <https://d2l.ai/chapter_recurrent-modern/bi-rnn.html>

[7] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681.2. Awni Hannun,

Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan

[8] D2l.ai. 2020. *9.4. Bidirectional Recurrent Neural Networks — Dive Into Deep Learning 0.15.1 Documentation*. [online] Available at: <https://d2l.ai/chapter_recurrent-modern/bi-rnn.html>.

[9] Zulqarnain, M., Ghazali, R., Hassim, Y. and Rehan, M., 2020. A comparative review on deep learning models for text classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(1), p.325.

[10] D2l.ai. 2020. *9.4. Bidirectional Recurrent Neural Networks — Dive Into Deep Learning 0.15.1 Documentation*. [online] Available at: <https://d2l.ai/chapter_recurrent-modern/bi-rnn.html>.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv.org, 24-May-2019. [Online]. Available: https://arxiv.org/abs/1810.04805.

[12] Processing, O., 2020. *Open Sourcing BERT: State-Of-The-Art Pre-Training For Natural Language Processing*. [online] Google AI Blog. Available at: <https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>.

[13] "BERT (language model)," *Wikipedia*, 12-Dec-2020. [Online]. Available: https://en.wikipedia.org/wiki/BERT_(language_model).

[14] "Sentiment Classification Using BERT," GeeksforGeeks, 02-Sep-2020. [Online]. Available: https://www.geeksforgeeks.org/sentiment-classification-using-bert/.