Sentiment Analysis of Twitter Data for movie reviews using LSTM

Jay Shah, Harsh Muniwala, Keya Shah, Deep Patel
Students of CSE, University of Texas at Arlington

ABSTRACT

With advances in the technological age and easy access to the Internet, the modern world has vast amounts of digital data available on social media platforms such as Twitter and Facebook. In recent years, a lot of research has been done in the field of NLP, especially in the field of sentiment analysis of multiple datasets collected by social media platforms. Among them, the Twitter dataset is one of the most popular works. These publicly available datasets can provide many hidden insights into crowd perceptions on specific topics. This is technically supported by machine learning algorithms such as classifiers and deep learning models. One of these great technologies is LSTM, which stands for Long-Short Term Memory. In this paper we have observed and analyzed the effects of using real world data on the LSTM models while training on the static datasets which are available specifically for research work, this provides an overview of the actual implementation of this neural network on to real-world data.

1. INTRODUCTION

Tweets on Twitter are often generated in real time, making them a valuable source of live data for sentiment analysis. By analyzing tweets in real time, researchers can gain insight into how public opinion and sentiment are changing over time. This data is very important for sentiment analysis due to their richness, brevity, hashtag categorization, and real-time nature. By analyzing tweets, sentiment analysis can provide valuable insight into public opinion and sentiment on various topics [3]. Using tweets from actual user of tweets can also provide the insight of public opinion on a specific topic, which can be of a great importance to business owners and many other fields.

Analyzing this data is a very complex process and is executed by machine learning techniques such as classifiers, regressions, and other deep learning models. These models are very complex mathematical equations and are not so easy to implement as there must be certain pre-processing on the data as well. Advancement in the field of machine learning has enables us to achieve high accuracy on the predictions of the models that has been trained on certain data. Machine learning is important in sentiment analysis because it enables the development of accurate and efficient algorithms for classifying large amounts of text data [2]. Machine learning algorithms can learn from tagged data so they can recognize patterns and relationships in the data. It can also handle noisy and ambiguous data, considering the context and linguistic features of the text.

Sentiment analysis using classifiers have given the accuracy of about 70-80 % for a long time but for higher accuracy rate we have to use Deep learning modules such as LSTM, RNN, CNN etc. LSTM is one of the most promising types of neural network and can perform better than RNN's as LSTM can overcome the problems of vanishing or bursting gradients which can be an issue with RNN's [2]. LSTM is neural network which is a variant of RNN but with memory retention. Using LSTM researchers have achieved accuracy of more than 94% even on validation data. With huge amount of research work in the field of NLP, researchers are achieving even better results using transformers, which are even more efficient than LSTM.

A lot of research work on sentiment analysis has taken place in last decade with the rise in popularity of topics such as Artificial Intelligence and Machine Learning. Researchers have worked with datasets and achieved high accuracy rates using machine learning models, but there is a catch to it. Most of the research work is done using the datasets which are easily available and structured in a way that's easy for researchers to work on, but that's not the case when that research is to be applied on real world application because than the data is very ambiguous and in order to deal with it, we need to do research work using real time data. In order to work with real time data, we used Tweepy API to fetch the data from twitter. Tweepy API is very useful as it gives access to 1500 tweets to work on. Furthermore, we worked on observation of using live data for prediction.

Our research work is about Binary classification of sentiments based on the tweets of people on a certain topic. For doing this we have used deep learning techniques named LSTM [1]. We have used a mix of static and live data for training and validation data to observe the effects. We have explained details of technicalities further in the Description section of this paper. We will be also sharing analysis of the output of this model and it effectiveness.

2. DATA DESCRIPTION

For our research work we used two different binary classified data sources for sentiment analysis:

1. IMDB movie review dataset: 1.6 million tweets classified as positive or negative are included in this dataset. We use a 50,000-tweet subset of this dataset, with 25,000 tweets used for training and 25,000 tweets used for testing. The average length of a tweet is 400 words, making them comparatively brief. The dataset contains a variety of tweets on movies.

2. Twitter Real time data: This dataset was fetched from live tweets with the specific movie tag (e.g., FastAndFurious) with the help of TWEEPY API and it was stored in CSV file as text format. It was further classified and labelled positive and negative manually; thus, the sentiment analysis of the data becomes easy to observe the effect of various features. The raw data having polarity is highly susceptible to inconsistency and redundancy.

The text data was pre-processed in order to eliminate extraneous information and uniformly format the content before training the LSTM networks.

Pre-processing of tweet include following points,
• Tokenization: Remove all URLs (e.g., www.xyz.com), hash tags (e.g., #topic), targets (@username); Using a tokenizer, the text input was divided into discrete words or tokens. We utilized the Tweet Tokenizer from the NLTK package to handle the special format of tweets, which includes hashtags, mentions, and emoticons, for the Twitter dataset.
• Stemming/Lemmatization: The text data was stemmed or lemmatized to lower the dimensionality of the feature space and increase the model's effectiveness. Lemmatization entails stripping words of their suffixes to return them to their canonical form, whereas stemming entails returning them to their base form.
• Correct the spellings; sequence of repeated characters is to be handled
• Replace all the emoticons with their sentiment.
• Remove all punctuations, symbols, numbers
• Remove Stop Words: To decrease noise in the text data, stop words, which are popular words with little to no meaning, were eliminated.

The text data was divided into training and testing sets for each dataset after pre-processing. The testing data was used to assess the model's performance while the training data was utilized to train the LSTM network.

Here we will be using static data to train the model whereas for validation set for our model will be a dataset which will be tagged by a human itself. In order to make our dataset compatible with our model we have also transformed our dataset from (.txt) to (.csv). this way it is much easier to read and perform actions on the dataset.

3. PROJECT DESCRIPTION

3.1 Description

We have applied Long Short-Term Memory (LSTM) algorithm to a Twitter dataset for the purpose of binary sentiment analysis. Our research work is about observing the result change and also to show the effects of use of live dataset instead of using a pre-built static dataset. That way a better understanding of the real-world application of the model can be studied and curated by making improvement where our knowledge lacks. We have architected a LSTM model for binary classification of data in a “positive” and “negative” label. We have worked with a mix type of data to train our model with industry ready standards.

Our LSTM model has a structure containing embedding layer, LSTM layer, Global average pooling layer, Dropout layer and a dense layer. Below given is out model architecture refer to it for the structure layout.

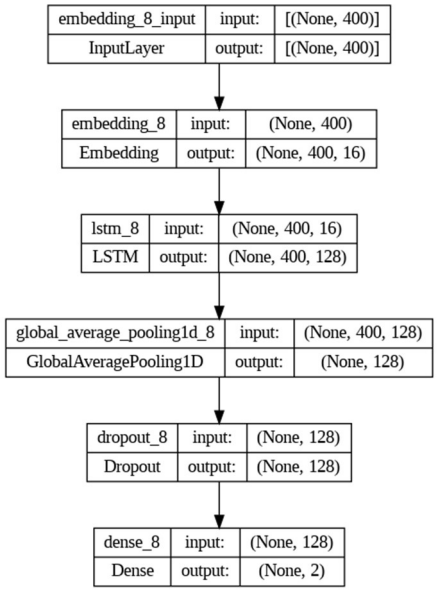


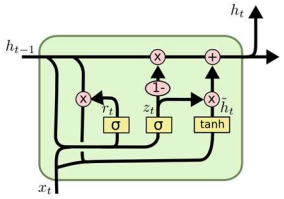
Fig. 1. Model Architecture

Using the above model, we managed to get accuracy rate of more than 94% and that too with a limited amount of computational and time resources. You can also find the model summary below.

```
(50000, 2)
Count Of Labels : {'negative': 25000, 'positive': 25000}
Total number of Labels : 2
Vocab length: 112718
Max sequence length: 400
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_7 (Embedding)     (None, 400, 16)           1803488

 lstm_7 (LSTM)               (None, 400, 128)          74240

 global_average_pooling1d_7   (None, 128)              0
 (GlobalAveragePooling1D)

 dropout_7 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 2)                 258

=================================================================
Total params: 1,877,986
Trainable params: 1,877,986
Non-trainable params: 0
```

Fig. 2. Model summary

LSTM works in a similar way to RNN and it work in a below given manner. You can also find the equation that LSTM uses for calculations below.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Fig. 3. LSTM Formulation

We have used a static dataset for training our model which is fetched from a dataset named as IMDB movie review dataset. We have used 50,000 instances of tweets in which 25,000 were labelled as “positive” and other half was labeled as “negative”. And for validation set we have used the data which we collected from Tweepy API and then manually labeled the data

3.2 Main References

- Sentiment analysis for movies reviews dataset using deep learning models
- Sentiment Analysis on Twitter Data Using Deep Learning approach
- SENTIMENT ANALYSIS USING DEEP LEARNING

3.3 Difference in approach/method

Articles referenced used static data for testing purposes, but we ran an analysis based on the live feed data and observed the changes in the result as live data can vary wildly from a pre-processed dataset. Our model was trained on the data set but validated on the real data representing the real-world scenario. While reference papers used pull test splitting of data. The low accuracy of the initial models necessitated the implementation of more advanced deep learning models such as recurrent neural networks (RNNs). RNNs have a storage unit that allows the inclusion of previous inputs in the prediction of the current output. The Long Short-Term Memory (LSTM) network is typically used to process large temporal data sets. The model worked satisfactorily on a large movie tweet dataset. LSTM along with proper adjustment of model and dataset complexity helped us achieve higher accuracy and less dataset loss. Reference papers had between 70 and 84% accuracy, but we achieved accuracy rates in excess of 93% by using the right data filtering and sanitization techniques. Our model was trained on datasets but validated on real data representing the real-world scenario. While reference papers used pull test splitting of data.[1]

Our main approach is to show the difference in using static dataset for training and testing and using live data to see the validations. There is a huge difference in using a split of a dataset and using a real time data because dataset is well prepared for training purpose whereas, live data is not so easy to work with. For e.g., use of sarcastic language has been on rise in recent trends as it reflects of humor, however the use of sarcasm in tweets hinders our program’s accuracy as our model classifies based on words used in a tweet. To tackle with this problem, we used lexicon-based approach in LSTM and we were able to overcome this issue.

3.4 Difference in accuracy/performance

Below given table shows the accuracy of model.

Reference Paper model	70-84 %
Long-Short Term Memory Unit (LSTM)	95%

Table 1. Accuracy comparison

As you can see our model was able to achieve higher accuracy than our reference paper. We achieved this by properly manipulating data and also by using different architecture of LSTM model which was given in the reference paper.

We even plotted the accuracy and loss for our model and even performed analysis on it. Further analysis is shown in the Analysis section of our paper. Our analysis is very important to our paper as it is where we discuss how change in data used to validate is prone to have much lower accuracy if not pre-processed in the right manner. This increased accuracy is the result of better built model.

4. ANALYSIS

We took the data from IMD movie review dataset and trained our model on it, thereafter we used the data from Tweepy API and manually labeled it and then used it to feed into our model as validation set. Doing so we were able to plot the accuracy and loss of our model as shown below.

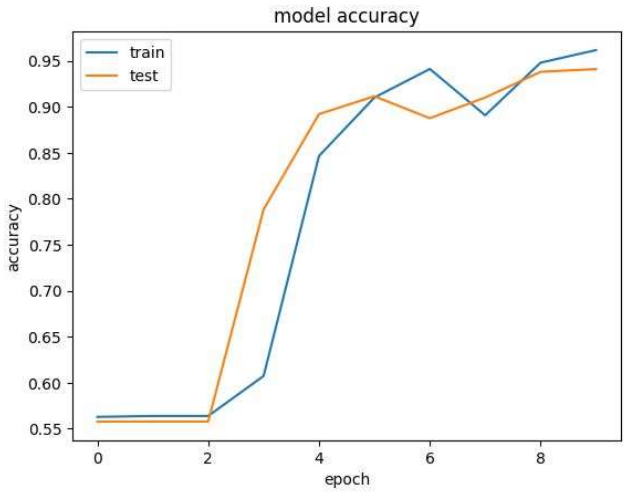


Fig 5. Plotting Accuracy

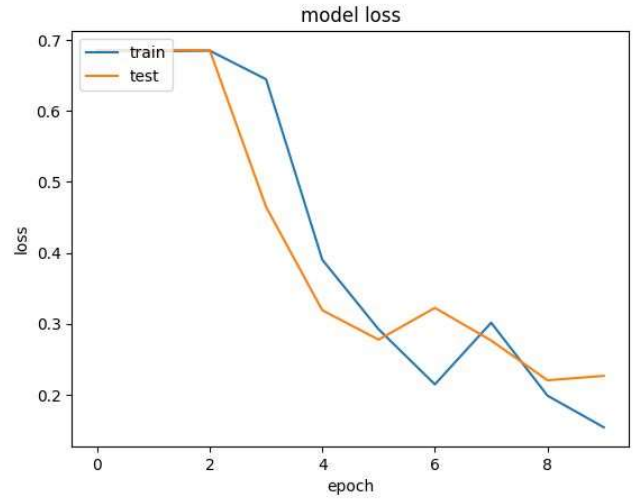


Fig 6. Plotting loss

As you can observe in the above given plots, we were able to achieve an accuracy of more than 90% and reduce the loss under 20% for validation set as well. If our validation dataset was to be a split from training data the learning curve would be much smoother whereas, here the crooked lines means that our data is not very well tuned. This also means that our model sometimes starts to just throw predictions without a reasoning to it. These issues can be overcome by having more computational power and better preprocessing steps.

4.1 What did I do well?

We performed analysis on the data which was cross matched between static and live feed data. This enables us to observe the irregularities of training on static data but then applying same model in real life application. We also fine tuned the data which helped us to improve the results for accuracy and loss. We even explored multiple papers to find out the right architecture for fitting such data. On top of that we even tried to change the model structure and that helped us to better understand how LSTM and other RNN's work, this led us to understand how minute changes in data can have vast impact on the classifications. We even used very accurate and effective pre-processing methods for feature extraction without any data loss, this helped us to better train our model as the data was fine tuned with the model and eventually helped us in the backpropagation.

4.2 What could I have done better?

If we would have got better resources, we could have increased the complexity of our model and that way better fine accuracy could be achieved. Firstly, we lacked the live feed data which was only available with 1500 instances at max and that was due to the policy of twitter. Secondly, we lacked computational power to run high complex neural networks. This made us use our time to try and fix the dataset to run in shorter duration with low computation power. If we would have more time to spend after this research, we could have achieved even smoother learning curve.

If given access to better resources such as high-end GPU and other tools of machine learning methods, we could fine tune the data to reduce the outliers effect on the output classification. We could even try a hybrid technology of other powerful algorithm to see the effect on the final output.

4.3 What is left for future work?

Use and observe other Deep learning model instead of LSTM for the same functionalities to see if better results can be achieved in terms of time and space complexity. This can help to even find a new highly efficient algorithm for such problems. Explore other data manipulation techniques to better fit your data to models and filter data for real instances. This is because there is room for improvement in the data fine tuning. Data can be manipulated and can result in much better and higher learning curve.

5. CONCLUSION

Our paper gives an overview of using LSTM in neural model to do sentiment analysis of twitter data. Twitter data being available in vast quantity becomes the topic of great importance as it can analyzed in different forms. Furthermore, we decided to not traditionally use the splitting of training data to use as validation data and so we managed and accessed live twitter data using Tweepy API. We did this to explore the usage of live feed data instead of just using old and redundant training dataset. By using this data, we were not able to get high accuracy and even loss was high but than we applied certain pre processing methods on the data which helped us to achieve better accuracy and lower the model loss. Towards the end of our research, we managed to achieve the model accuracy of more than 93%. This research paper is also a proof of need to change the traditional manner of research in machine learning as the scenario of real-world problems is very different than the research environment.

6. REFERENCES

[1] Nehal Mohamed Ali, Marwa Mostafa Abd El Hamid and Aliaa Youssif. , "Sentiment analysis for movies reviews dataset using deep learning models", In Proceedings of the International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.9, No.2/3, May 2019.

[2] Vishu Tyagi, Ashwini Kumar, Sanjoy Das. , "Sentiment Analysis on Twitter Data Using Deep Learning approach", In Proceedings of the 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), December 2020.

[3] Shilpa PC, Rissa Shereen, Susmi Jacob, Vinod P. , "SENTIMENT ANALYSIS USING DEEP LEARNING", In Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), February 2021.

[4] Monisha Kanakaraj, Ram Mohana Reddy Guddeti. , "NLP based sentiment analysis on Twitter data using ensemble classifiers" In Proceedings of the 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), March 2015.

[5] Priyanka Tyagi, R.C. Tripathi, "A Review towards the Sentiment Analysis Techniques for the Analysis of Twitter Data" In Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), February 2019.

[6] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu & Gayathri Karthick. , "Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)" from Wireless Personal Communications, 2021.

[7] V.Uma Ramya, K. Thirupathi Rao, "Sentiment Analysis of Movie Review using Machine Learning Techniques", From International Journal of Engineering & Technology.

[8] Yogesh Chandra; Antoreep Jana, "Sentiment Analysis using Machine Learning and Deep Learning", In Proceedings of 7th International Conference on Computing for Sustainable Global Development (INDIACom), March 2020.

[9] Mohammed H. Abd El-Jawad, Rania Hodhod, Yasser M. K. Omar, "Sentiment Analysis of Social Media Networks Using Machine Learning", In Proceedings of 14th International Computer Engineering Conference (ICENCO), December 2018.

[10] Yaser Maher Wazery, Hager Saleh Mohammed, Essam Halim Houssein, "Twitter Sentiment Analysis using Deep Neural Network", In Proceedings of 14th International Computer Engineering Conference (ICENCO), December 2018.

[11] Sheresh Zahoor, Rajesh Rohilla, "Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study", In Proceedings International Conference on Advances in Computing, Communication & Materials (ICACCM), August 2020.

[12] R. Monika, S. Deivalakshmi, B. Janet, "Sentiment Analysis of US Airlines Tweets Using LSTM/RNN", In Proceedings of IEEE 9th International Conference on Advanced Computing (IACC), December 2019.

[13] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., &Stede, M.."Lexicon basedmethods for sentiment analysis". Computational linguistics, 2011:37(2), 267-307.

[14] Li, S., Xue, Y., Wang, Z., & Zhou, G.."Active learning for cross-domainsentiment classification". In Proceedings of the Twenty-Thirdinternational joint conference on Artificial Intelligence (pp. 2127-2133). AAAI Press,2013.

15] Bollegala, D., Weir, D., & Carroll, J.. Cross-Domain SentimentClassification using a Sentiment Sensitive Thesaurus. Knowledge andData Engineering, IEEE Transactions on, 25(8), 1719-1731,2013.