Twitter Sentiment Analysis using CNN and LSTM

Pramod Acharya

University of South Dakota, MS Computer Science

Abstract

Twitter sentiment analysis involves automatically analyzing text data to identify the emotions and opinions expressed in public tweets from various fields, including marketing and politics. With a large number of tweets being posted with hashtags, it's challenging for researchers to accurately interpret the context in which specific tweet words are used, making it difficult to determine whether a tweet is negative or positive. A compromised system can greatly reduce user reliability. To address this challenge, our project performs sentiment analysis on 1.6 million Twitter data using a deep learning approach that combines feature work with tweet words, word2vec, and stop words integrated with CNN and LSTM models. These algorithms can identify stop word count patterns with unique strategies, enhancing the accuracy of the models. Our proposed approach includes two neural network models, CNN and LSTM, which are trained to classify the tweets as positive or negative.

Keywords: LSTM, CNN, Sentiment analysis

1. Introduction

Natural Language Processing (NLP) is a branch of artificial intelligence that involves using natural language to communicate with intelligent systems. Sentiment analysis is a popular use of NLP, particularly in social media platforms like Twitter and Facebook. [1]NLP is also used for chatbots, speech recognition, machine translation, spell checking, keyword searching, information extraction, and advertisement matching. Sentiment analysis involves using NLP and tweet analysis to automatically determine the personal feelings expressed in tweet data. Deep learning, a subset of machine learning that utilizes artificial neural networks, has become increasingly popular for sentiment analysis. In this paper, we propose a sentiment analysis approach on Twitter data using deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). We preprocessed the data using various preprocessing techniques and prepared the deep learning model by performing tokenization and padding on the training data and created an embedded matrix using pre-trained embeddings such as Glove. [2]The proposed approach can be beneficial for sentiment analysis in various fields, including marketing and politics, and provides a new and efficient way to analyze large volumes of data on social media platforms to determine the personal feelings expressed in tweet data automatically. Our proposed LSTM and CNN models achieved an accuracy of approximately 78.23% and 77.36%, respectively, on sentiment analysis for Twitter data. This indicates that the LSTM model was able to better capture the context and long-term dependencies of the text, resulting in more accurate predictions. These results demonstrate the effectiveness of our approach, which integrates feature work with tweet words, word2vec, and stop words into the deep learning models for sentiment analysis.

2. Motivation

Twitter sentiment analysis has become an important area of research due to the massive volume of user-generated content on the platform. Social media platforms such as Twitter have become a popular source of data for sentiment analysis, as they provide a large amount of real-time data on how people feel about different topics. The ability to accurately classify tweets as positive, negative, or neutral has a wide range of applications in fields such as business intelligence, political analysis, and social science research. In recent years, deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) have shown promising results in sentiment analysis tasks. CNNs can effectively capture local patterns and correlations in text data, while LSTMs can model longer-term dependencies in the data, making them well-suited for time-series data such as tweets. As a result, combining CNN and LSTM models can improve the accuracy and effectiveness of sentiment analysis.

In this research paper, we aim to explore the use of CNN and LSTM models for Twitter sentiment analysis. Our research will focus on the impact of different model architectures, hyperparameters, and pre-processing techniques on the accuracy and efficiency of sentiment analysis.

3. Related work and Literature Review

Social media platforms, particularly Social Network Services (SNS), generate massive amounts of data that can be useful for prediction and sentiment analysis. Twitter is an SNS that generates a vast amount of data through user posts, making it an ideal source for sentiment analysis and text mining research. However, managing unstructured data of this scale is challenging, and machine learning techniques, such as deep learning, can be useful. Deep learning algorithms are more efficient and offer improved performance compared to traditional methods like the surface approach, which relies on manual feature extraction. This report proposes a deep learning technique that can be combined with feature extraction and presents three stages of research: developing sentiment classifiers, ensemble techniques, and information merging to obtain the final set of sources, and analyzing the performance of various models.

Some related research in the field of sentiment analysis are Rachana Bandana (2018), Materials Engineering and Nano-Technology (IEMENTech) IEEE, proposed the sentiment analysis with document level for the movie review classification applied by heterogeneous features, for supporting the supervised machine learning algorithms as Naive Bayes and Linear Support Vector Machine. M. Ali Fauzi (2018), IEEE Access, shows the ensemble technique with Naive Bayes for sentiment analysis of movies on Indonesian Twitter data. Vasilija Uzunova (2018) Personal and ubiquitous computing, proposed sentiment analysis of film reviews in Macedonian using Naive Bayes [3].

4. Proposed Methodology

4.1 Data Collection

The dataset used in this project is taken from Kaggle, which contains 1.6 million tweets (238.8 MB in size) in which the positive values are “800k”, and negative values are “800k”. The tweets have been annotated (0 = negative, 1 = positive) and can be used to detect the sentiment.
According to the creators of the dataset: "Our approach was unique because our training data was automatically created instead of having humans manually annotate tweets. In our approach, we assume that any tweet with positive emoticons, like :), were positive, and tweets with negative emotions, like :(, were negative. We used the Twitter Search API to collect these tweets by using keyword search".

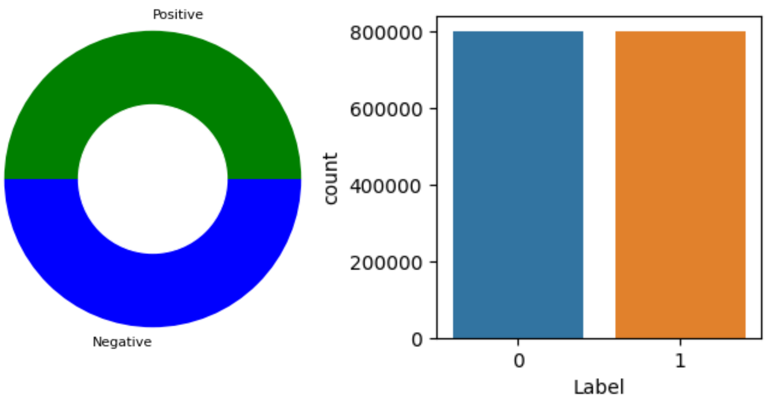

Fig 1. Equal number of Positive and negative tweets

4.2 Implementation

4.2.1 Data Preprocessing:

To prepare the Twitter data obtained from Kaggle for sentiment analysis, we performed several preprocessing steps as preprocessing is an essential step in any pattern recognition and machine learning problem, and for Twitter sentiment analysis, it's particularly important as the text data is often unstructured and noisy [4].

- Removed null values as no models can handle NULL values on their own.
- Performed tokenization to split the raw text data into a list of tokens. This process helps us better understand the data and develop the model.

- Lemmatized the tokenized data, which allows mapping of multiple words to common root words, treating the words similarly.
- Removed stop words, which repeatedly appear in the text, but do not add much value. Removing them shifts the focus to more unique data that holds significant information.
- Removed all special characters and punctuation in the preprocessing step, generating a cleaner tweet with lowercase characters.

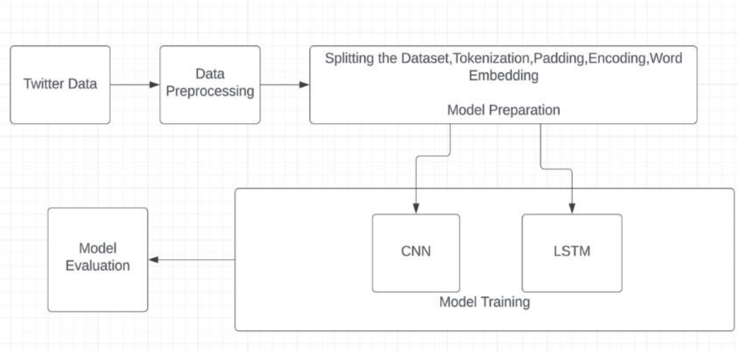

Fig 2. Proposed Methodology

4.2.2 Model Preparation:

The model preparation phase is an essential step in the deep learning pipeline, and it involves several critical steps that are essential for building an effective and accurate model. The first step in model preparation is splitting the dataset into training and testing sets. This step is necessary to ensure that the model is trained on a portion of the dataset and tested on the remaining portion. This helps to prevent overfitting and ensures that the model generalizes well to unseen data.

The next step is tokenization, where each word in the text is converted into a unique integer value or token which is fitted on the training data's tweet column to create mappings of words to their integer values stored in its internal vocabulary. Tokenization is an important step that converts textual data into a format that can be processed by the neural network. After tokenization, the input sequences are padded to a fixed length which ensures that all input sequences are of the same length, which is necessary for feeding them into the neural network.

Furthermore, we created word embeddings which are a type of dense vector representation that maps each unique word in the vocabulary to a vector of fixed length. This step is crucial as it helps to capture the semantic meaning of words and the relationships between them. A pre-trained GloVe embedding model is used to create an embedding matrix, which is initialized with the pre-trained word embeddings. capture the semantic meaning of words and their relationships.

4.2.3 Model Training:

A. Convolutional Neural Network (CNN):

A Convolutional Neural Network (CNN) is an advanced machine learning algorithm designed to excel at image processing and recognition. The network is composed of multiple layers, including

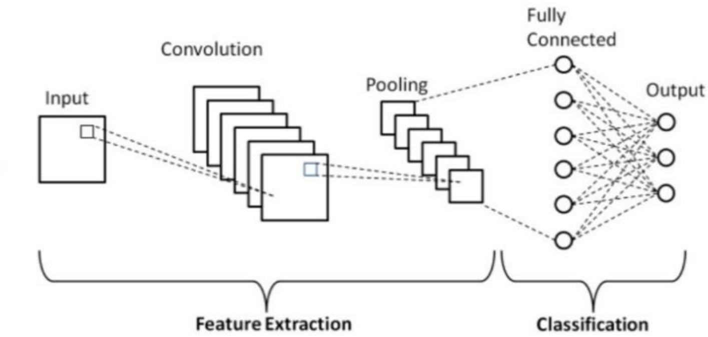

Fig 3. Schematic diagram of a basic convolutional neural network

Our CNN model architecture consists of multiple convolutional layers, batch normalization, and dropout layers for regularization. The model starts with an input layer and an embedding layer, followed by a spatial dropout layer with a 0.2 dropout rate. The model then contains three convolutional layers with 64 filters, kernel size of 5, and batch normalization. A max-pooling layer is added after the first two convolutional layers to down sample the feature maps. The output is flattened and passed through two dense layers with 512 and 256 neurons, both with batch normalization. A dropout layer with a rate of 0.5 is added after the first dense layer. The model ends with a dense output layer with a sigmoid activation function. The model is then compiled, trained, and evaluated, with a test loss of 0.470 and a test accuracy of 77.36%

B. Long Short-Term Memory (LSTM):

LSTM is a type of recurrent neural network that addresses the issue of long-term dependencies in RNN. RNN feeds output from the last step as input in the current step [5]. Hochreiter & Schmidhuber developed LSTM to overcome this problem. RNN struggles to predict words stored in long-term memory as the gap between them increases. However, LSTM can retain information for a longer time period, making it useful for processing, predicting, and classifying time-series data.

Our model architecture consists of an LSTM layer, dense layers, and dropout layers for regularization. The model starts with an input layer followed by an embedding layer. Then, an LSTM layer with 128 units is added, incorporating a 0.2 dropout rate and a 0.2 recurrent dropout rate. Subsequently, two dense layers with 128 and 64 neurons are included, with Leaky ReLU activation functions (alpha=0.1). After each dense layer, a dropout layer with a 0.5 rate is added to prevent overfitting. The model concludes with a dense output layer using a sigmoid activation function. The LSTM model is then compiled, trained, and evaluated with the Test loss of 0.455 and Test accuracy of: 78.23%.

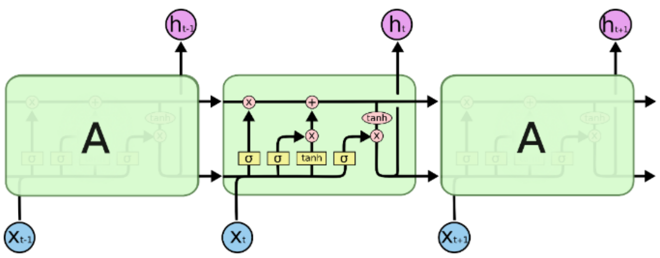

Fig 4. The repeating module in an LSTM contains four interacting layers

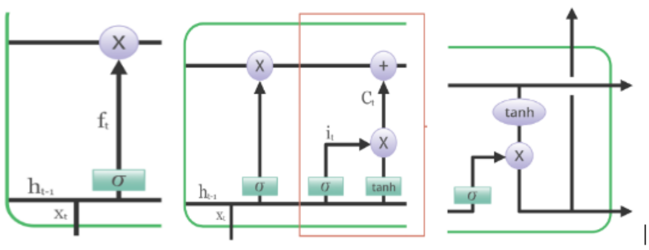

Fig 5. Shown are the Left (Forget Gate), Center (Input Gate), and Right (Output Gate)

4.2.4 Model Evaluation:

Training Accuracy & Validation Accuracy:

We observed that for both of the models, both the training and validation accuracy progressively increased, following a similar pattern as the training accuracy. The convergence of these accuracies towards the end of the training process indicates that the model has achieved a good level of performance in estimating the target variable.

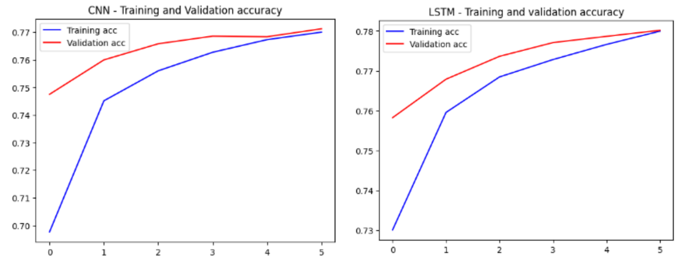

Fig 6. Comparison between training and validation accuracy using CNN (left) and LSTM model(right)

Training Loss and Validation Loss:

We can observe that for both of the models training and validation loss are gradually decreasing. suggesting that the model is not overfitting. The validation loss is lower than the training loss, which suggests that the model is not overfitting the training data. In addition, the difference between the validation loss and the training loss is not very large, which suggests that the model is not suffering from high bias.

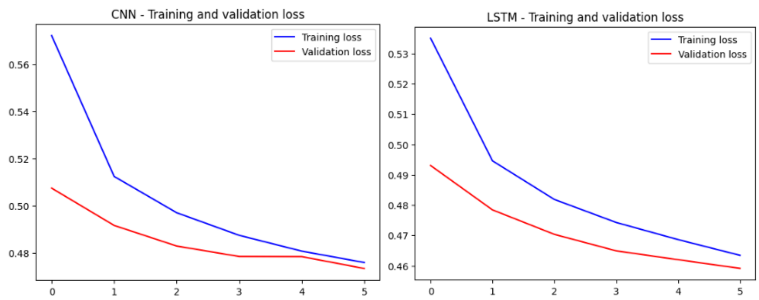

Fig 7. Fig: Shown is the comparison between training loss and validation loss using CNN(left) and LSTM model (right)

5. Results

5.1 Classification Report:

We have presented the results obtained from the CNN and deep learning model in terms of Accuracy, precision, recall, and F-measure (F1). The macro average value for those parameters; accuracy, precision, recall, and F1 have 0.77 in the CNN model where each parameter has higher values for positive tweet annotation except for precision. This pattern is similar in the case of the LSTM model with those parameters having 0.78 in an average with a little higher for positive annotation except for precision.

The higher value of those average statistics parameters in the LSTM model as compared to CNN by 1.29% means, the LSTM model has a higher ability to identify true positive cases, correctly classify all instances, and strike a balance between identifying true positive cases and avoiding false positive cases.

	precision	recall	f1-score	support
0	0.78	0.76	0.77	160156
1	0.77	0.79	0.78	159844
accuracy			0.77	320000
macro avg	0.77	0.77	0.77	320000
weighted avg	0.77	0.77	0.77	320000

Fig 8. Precision and Recall form CNN Model-annotated (0 = negative, 1 = positive) and can be used to detect the sentiment

	precision	recall	f1-score	support
0	0.79	0.77	0.78	160156
1	0.77	0.80	0.79	159844
accuracy			0.78	320000
macro avg	0.78	0.78	0.78	320000
weighted avg	0.78	0.78	0.78	320000

Fig 9. Precision and Recall form LSTM Model-annotated (0 = negative, 1 = positive) and can be used to detect the sentiment

5.2 Confusion Matrix:

A confusion matrix is a simple matrix that defines the model performance on the test data. It is the combination of predicted and the actual values. The basic terminologies in the confusion matrix are:

- True positives (TP)
- True negatives (TN)
- False positives (FP)
- False negatives (FN)

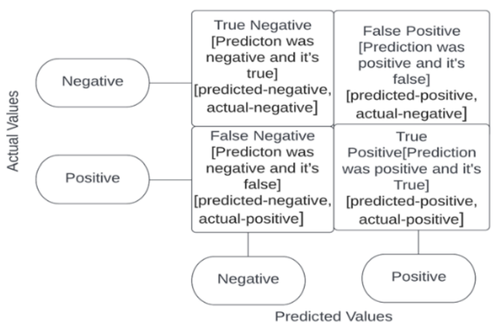

Fig 10. Confusion matrix display, which is used measuring the performance of model

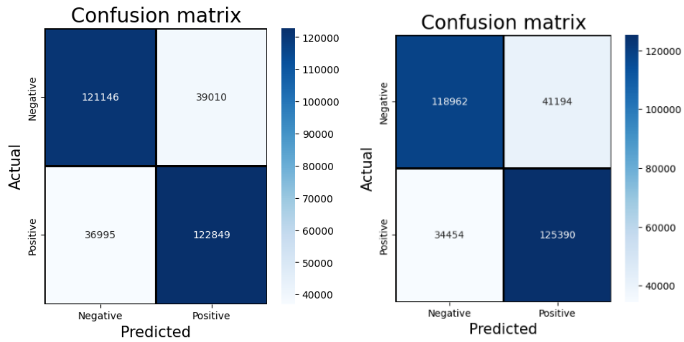

Fig 11. Confusion matrix using CNN model (left), Confusion matrix using LSTM model (right)

6. Conclusion

The functional model we developed includes features that significantly influence the evaluation and analysis of Twitter sentiment. We explored deep learning models that employ natural language processing techniques and used them to classify test data. Mathematical expressions were used to evaluate the performance of CNN and LSTM models, which produced excellent results for the feature work [6]. Moving forward, we plan to explore CNN models with various techniques and incorporate more complex features to enhance the accuracy of text data analysis. From the model training of both CNN and LSTM, it yields the test accuracy 77.36 % and 78.23% respectively.

References

[1] Ramadhani, Adyan Marendra, and Hong Soon Goo. "Twitter sentiment analysis using deep learning methods." *2017 7th International annual engineering seminar (InAES)*. IEEE, 2017.
[2] Shelar, Amrita, and Ching-Yu Huang. "Sentiment analysis of twitter data." *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018.
[3] Permatasari, Rosy Indah, et al. "Twitter sentiment analysis of movie reviews using ensemble features based Naïve Bayes." *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*. IEEE, 2018.
[4] Phung, Van Hiep, and Eun Joo Rhee. "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets." *Applied Sciences* 9.21 (2019): 4500.

[5] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks." *arXiv preprint arXiv:1909.09586* (2019)
[6] A Razia Sulthana, A K Jaithunbi, L Sai Ramesh P., 2018. Sentiment analysis in twitter data using data analytic techniques for predictive modelling. National Conference on Mathematical Techniques and its Applications (NCMTA 18) IOP Publishing. IOP Conf. Series: Journal of Physics: Conf. Series 1000 (2018) 012130 doi :10.1088/1742-6596/1000/1/012130