IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SENTMENTAL ANALYSIS USING LSTM NETWORKS AND TEXT PREPROCESSING

G.K.Venkata Narashimha Reeddy
dept.of .Computer science and artificial intelligence
St.Johns college of engineering and technology
Yemmiganur,

Polishetti Aparna
dept.of.Computer science and artificial intelligence
St.Johns college of engineering and technology
Yemmiganur,

Kunday Malika Muskan
dept.of.Computer science and artificial intelligence
St.Johns college of engineering and technology
Yemmiganur,

Darji Zakiya Simran
dept.of.Computer science and artificial intelligence
St.Johns college of engineering and technology
Yemmiganur,

Alur Swathi
dept.of.Computer science and artificial intelligence
St.Johns college of engineering and technology
Yemmiganur,

Abstract

With the aid of Long Short-Term Memory (LSTM) networks, this project seeks to explore and introduce a deep learning approach to sentiment analysis of film reviews. The proposed model utilizes an LSTM structure with embedding and dropout layers following a text preprocessing pipeline involving tokenization, sequencing, and padding. The IMDB dataset, comprising out of 50,000 movie reviews, is employed to train the model. The model classifies reviews as either negative or positive with a 0.8999% accuracy. Binary cross-entropy loss and Adam optimizer are employed to train the LSTM model, and hyperparameters are tuned for improved performance. Experimental results indicate the performance of the proposed method in addressing sequential dependencies in text data and captures intricate emotion patterns. The study demonstrates how deep learning methods can be applied to sentiment analysis, offering.

***Keywords*—Sentiment Analysis, Text preprocessing, LSTM, Deep Learning, CNN, NLP**.

I. INTRODUCTION

Sentiment analysis, being the most prevalent application of Natural Language Processing (NLP), allows companies, researchers, and policymakers to analyze vast text data on social media, customer reviews, and online discussion forums. Since the digital content has been growing exponentially, deep learning techniques have played a pivotal role in sentiment classification, particularly through the application of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). LSTM, an architecture of Recurrent Neural Networks (RNN), is particularly good at identifying long-term dependencies in text, while CNN is very good at local pattern recognition and spatial hierarchies and hence, quite surprisingly, is powerful for text processing. Merging the strength of both the architectures, the hybrid CNN-LSTM model uses CNN for feature extraction and LSTM for sequential dependency, leading to improved sentiment classification accuracy.

For optimal performance, text preprocessing is an important task, and this includes important steps such as tokenization, stopword removal, stemming, lemmatization, and word embeddings (Word2Vec, GloVe, FastText). These operations improve noise suppression, improved feature representation, and model efficiency in general. In contrast to individual models, the CNN-LSTM hybrid model has been discovered to be more robust withstood improved feature extraction, improved noise immunity with increased classification accuracy.

This article explains and contrasts the performance of CNN, LSTM, and their hybrids using standard datasets like Amazon reviews and IMDB. This article finally illustrates how hybrid deep learning techniques can efficiently enhance the performance of text classification tasks as well as contribute more to the innovation of sentiment analysis.

II. LITERATURE SURVEY

Alm et al. [1] introduced machine learning solutions for *emotion prediction based on text* and highlighted the significance of the contribution of supervised learning and feature engineering to emotional tagging of children's text. With the use of WordNet glosses for sentiment-bearing word extraction, Andreev Kaia and Bergler. [2] introduced a *fuzzy sentiment scoring mechanism*, which is an automated solution for

manually developed sentiment lexicons. With their *neural probabilistic language model*, Bengio et al. [3] removed the *curse of dimensionality* by finding word distributed representations, opening the door to later embedding techniques like word2vec and opening the door to deep learning in NLP. To identify hidden topics in opinionated texts, Blei et al. [4] introduced *Latent Dirichlet Allocation (LDA)*, a probabilistic model that has been used extensively in sentiment research. To further improve cross-lingual sentiment analysis, Boyd-Graber and Resnik .[5] extended LDA with a supervised multilingual LDA model with sentiment labels. Strapp Arava and Mihalcea. [6], attempting to compare machine learning models, proposed an emotion-annotated dataset, but emphasized the importance of affective lexical resources. Pang and Lee.[7], in their first survey on opinion mining and sentiment analysis, surveyed supervised and unsupervised approaches. Recursive Neural Tensor Networks (RNTN) introduced by Socher et al.[8] provide phrase-level sentiment classification with enhanced sense of context.Word2vec, one of the early word embedding models to identify semantic relationships and assist sentiment analysis, was originally suggested by Mikolov et al. [9]. Peters et al. [10] subsequently enhanced contextualized embeddings through the use of ELMo, which changes word meanings dynamically according to their context.

The NRC Emotion Intensity Lexicon by Mohammad et al. [12] enables in-depth emotion analysis compared to sentiment classification between two classes. More advanced AI-based sentiment analysis was made possible after Brown et al.[13] released GPT-3 with sentiment-aware text generation. More contextually oriented accurate models for opinion mining, emotion classification, and sentiment detection have been the result of improvements in the above methods. More recent transformer architectures improve significantly both accuracy and explainability of emotion and mood analysis through advances in deep learning.

III. PROPOSED WORK

We introduce in this paper a sentiment analysis approach employing an LSTM classifier with BERT-based text preprocessing. Word2Vec and GloVe are instances of word embeddings used in conventional sentiment analysis methods, though more recent Natural Language Processing (NLP) advancements have shown the capability of Transformer-based architectures such as BERT to recover fine-grained contextual semantics. Because the BERT tokenizer maintains word semantic context between words when doing text-to-subword token splitting, we make use of it in place of standard tokenizing algorithms. We then take a pre-trained BERT to generate contextual word embeddings from text that has already been tokenized. As BERT embeddings define words contextually dynamically depending on their contextual situation, they are far more superior to fixed embeddings for use in sentiment analysis. Once the embeddings are achieved, they are fed into an optimal Long Short-Term Memory (LSTM) network best suited for sequential data. As it gets longer in terms of time, the LSTM model develops a capability to learn to recognize emotion patterns in emotion expressions by being sensitive to long-range dependencies of the text. Our method effectively classifies text into positive and negative sentiments by taking advantage of the capability of LSTM in sequential data processing and the strength of BERT in building strong word representations

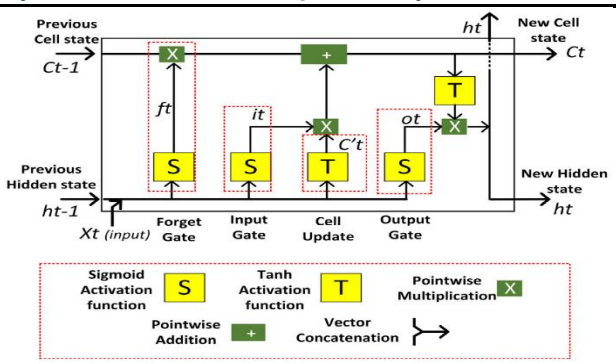

Fig1. LSTM Network Flow

The sentiment class probability distribution is offered by a softmax-activated dense layer for output classification. With an integration of the optimum features of recurrent and Transformer models, this hybrid model ensures performance over the traditional approach. We enhance precision in sentiment prediction by using LSTM for classification and BERT for feature extraction, which increases the capacity of the model to process complex and subtle textual input. This research could be extended further by exploring the manner in which the performance of many datasets is boosted by fine-tuning BERT in the LSTM model. We enhance the precision of sentiment predictions and improve the robustness of the model against complex and contextualized textual data through the utilization of LSTM as the classifier and BERT for extracting features. In a bid to achieve optimal performance on varied datasets, additional research would build upon these findings by exploring how BERT could be tuned within the LSTM model.

IV. ARCHITECTURE OF PROPOSED NETWORK USED

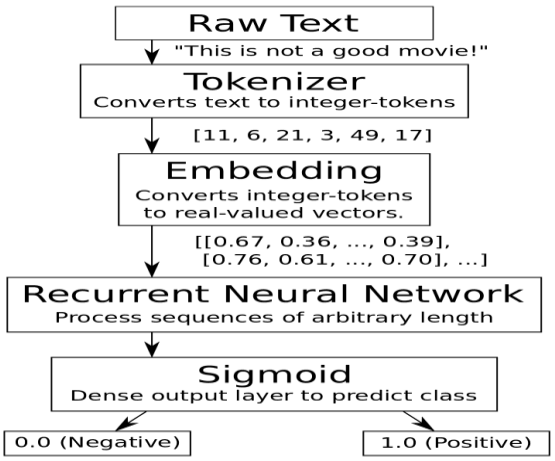

Fig 2. LSTM ARCHITECTURE

The architecture shown in the Fig.2 is a LSTM architecture from this architecture Raw Text, Tokenizer, Embedding, RNN and sigmoid are discussed.

A. Raw Text:

We use datasets of Amazon products and IMDB movie reviews [6] to train and test our models. In total, these. Tweets labeled as positive or negative are available in datasets. We simply load it if you have saved it on your computer in the form of a text file. Then we remove the punctuation and convert the text to lower case. Everything is in one long string. We now have to isolate individual reviews and keep them in separate list members. review_1, review_2, review_3……. review n, for instance.

B. Tokenizer:

Text tokenization is a technique of splitting text into very minute units or tokens, i.e., words or sentences. Tokenization is a very crucial task of natural language processing, particularly sentiment analysis, where very small issues like punctuation marks make a huge difference in meaning. Tokenization-based text data preprocessing makes text data easily readable and analyzable for machine learning algorithms. Steps to Tokenization and Encoding.

- Vocab to int mapping: To get word frequencies and assign high-frequency words to smaller indexes, use Python's collections. Counter. Processing is quicker, and memory usage decreases.
- Words to Integer Mapping: Substituting the equivalent integer value of words in reviews with vocab mapping. Thus, every review is a list of numbers in an orderly manner.
- Pad and Normalize Sequences: Pad all the reviews to the same length by padding the shorter reviews with zeros. This is a requirement for deep learning models to be processed in batches.
- Label the Encodings: The positive sentiment should be marked as 1 and negative sentiment as 0. This will enable the model to differentiate between various sentiment groups.

C. Embedding

A Natural Language Processing (NLP) technique, word embedding finds its basis on the cross-fertilization between computational linguistics, computer science, artificial intelligence, and machine learning. Founded on the distributional hypothesis where it is hypothesized that co-occurring words are semantically equivalent, it constructs the relationships among the words of the text data by their syntactic and semantic worth. Count-based co-occurrence statistics-based embeddings and prediction-based neural network-based embeddings for learning correlations are the two basic types. Dense vector representations with language relationships are referred to as embeddings. The input dimension defines the vocabulary size, the output dimension defines the vector space, and the input length is the longest sequence. The embedding layer maps integer indices to dense 128-length vectors. The direction and strength of such dense vectors indicate the word relationships, and they have a reduced dimensionality with word meaning. Similar words are clustered in this vector space. This process is performed by an embedding layer in deep learning libraries such as TensorFlow and Keras, which transforms words to their vector forms through a lookup table.

D. Embedding Layer :

A learned word embedding trained together with a neural network model on a particular linguistic communication processing task, like document classification or language modeling, may be termed an embedding layer for lack of a better term. In order to make sure that each word is one-hot encoded, the text of the document must be prepared and cleaned. The scale of the vector space—maybe 50, 100, or 300 dimensions—is model-determined. The vectors are initialized from a few random numbers. In the front end of a neural network, the embedding layer is supervised-trained. In spite of its possibility of slowness and high training data needs, this embedding layer learning will produce an embedding that is optimized to the NLP task and the particular text data.

E. Sigmoid

Such a common activation function used in machine learning, especially in binary classification problems, is the sigmoid function. It is useful for probability estimation as it normalizes input values into the range of 0 to 1. The Mathematical function $\sigma(x) = 1 / 1+e^{-x1}$ The S-shaped character of the function in gradient-based optimization offers smooth and differentiable properties. Its ability to handle probability-based outputs is one of its distinguishing features, making it popular among logistic regression and neural networks. Its weakness is the vanishing gradient problem, which decelerates deep network learning by generating gradients that are nearly zero when x is very high or very low.

V. WORKING OF LSTM NETWORK

- Take the latest input, the current internal state of the cell, and the previous hidden state as inputs.
- Determine the value of each of the four gates by doing the following actions:

For each gate, perform element-wise multiplication between the corresponding vector and the respective weights to obtain the parameterized vectors for the input of the present moment and the hidden state of the previous moment. Apply the respective activation function to each gate component across the parameterized vectors. The gates and the activation functions that must be applied on every gate are as follows:

Gates and activation functions.

- In order to obtain the current internal cell state, first calculate the input gate and input modulation gate element-wise multiplication vectors. Then, calculate the forget gate element-wise multiplication vector and the resultant internal cell state. Lastly, sum the two vectors.

$$C_t = (ft.C_{t-1}) + (I + C)$$

- To get the current hidden state, first get the element-wise hyperbolic tangent of the current internal cell state vector, and then apply the output gate to perform element-wise multiplication.

$$h_t = o_t \, . \, \tanh(C_t)$$

An LSTM network, as recurrent neural networks, produces an output at every time step, which is employed to train the network with gradient descent.

The sole inherent difference between Recurrent Neural Networks' Back-Propagation algorithms and Long Short Term Memory Networks is in the mathematics of the algorithm.

VI. RESULTS AND DISCUSSION

A. Summary of Dataset

Another very widely used sentiment analysis benchmark data set, and one on movie reviews, is the IMDB data set. The data set includes 50,000 reviews, 25,000 positive and 25,000 negative. The data set is used to train and test a deep learning model for sentiment classification in this experimental research. 50,000 out of available reviews are used for training, and 23,500 more reviews are reserved to test the performance of the model. Classifying a review as

polarizing—that is, whether the review is positive or negative in sentiment—is the primary goal of this research.

Table1: Summary of IMDB Dataset

Dataset	Total Sample	Train Sample	Test Sample	class
IMDB Data Set	50000	40000	10000	2

B. Layers of Model

Model: "CNN-LSTM Sentiment Analysis"

_____

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 100)	500000
conv1d (Conv1D)	(None, 196, 128)	64128
max_pooling1d (MaxPooling1D)	(None, 98, 128)	0
lstm (LSTM)	(None, 98, 100)	91600
lstm_1 (LSTM)	(None, 100)	80400
dense (Dense)	(None, 1)	101

Total params: **736,229**
Trainable params: **736,229**
Non-trainable params: **0**

_____

C. Model Architecture

Configuration of the model	Epochs	LSTM Units	Accuracy
Embedding Layer + LSTM - CNN Layer + Dense Layer	1	128	77.97
	2	128	90.93
	3	128	93.10
	4	128	95.10
	5	128	96.79

The accuracy was boosted step by step with the epochs from 77.97% in the first to 96.79% in the fifth.Both spatial and sequential dependencies were both acquired from text data by jointly using CNN and LSTM. Improved sentiment classification was achieved by utilizing LSTM to handle long-term dependencies and CNN to handle local word patterns. As per the findings, the learning capability of the model is enhanced by the number of epochs and misclassification decreases. The strength of the model in executing the task of sentiment analysis is shown by the high accuracy at the end. For further improvement in generalization across various datasets, other types of optimizations such as dropout regularization and hyperparameter tuning can be tried but the issues of overfitting also have to be addressed.

VII. CONCLUSION AND FUTURE SCOPE

Using the IMDB dataset, this research work efficiently employed the use of hybrid CNN-LSTM for sentiment classification that efficiently detected short-term as well as long-term text patterns. Its high accuracy reflects the superiority of deep learning when compared to the traditional machine learning approaches for sentiment classification. Experiments reflect the fact that hybrid deep models can be deployed in applications such as opinion mining, customer sentiments, and social media monitoring.

To enhance feature representation, future developments might include advanced embedding techniques such as BERT, GloVe, or FastText. Higher classification performance may also be found through model optimization using BiLSTMs and attention. Although real-time application in real-world usage like sentiment analysis of chatbots and customer feedback surveillance might enhance usage, increasing the dataset to more languages and subjects might enhance generalization. Also, scalability of the model will be achieved by deploying the model on cloud platforms to make it usable for sentiments analysis at large scales.

REFERENCES

[1] O. Alm, D. Roth, and R. Sproat. 2005. Emotions From text: machine learning for text-based emotion prediction. In Proceedings of HLT/EMNLP, pages 579–586.
[2] A. Andreev Kaia and S. Bergler. 2006. Mining WordNet for fuzzy sentiment: sentiment tag extraction From WordNet glosses. In Proceedings of the European ACL, pages 209–216.
[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. a neural probabilistic language model. Journal of Machine Learning Research, 3:1137–1155, August.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. Journal of Machine Learning Re-search, 3:993–1022, May.
[5] J. Boyd-Graber and P. Resnik. 2010. Holistic Sentiment analysis across languages: multilingual supervised latent Dirichlet allocation. In Proceedings of EMNLP, pages 45-55.