Sentiment Classification Using Hybrid TextBlob Bi-LSTM Deep Learning Model

Abstract:

Sentiment analysis techniques are used to classify tweets as positive or negative. While traditional machine learning methods often struggle with low performance, deep learning models Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM) have shown significant improvements. In this method a novel hybrid TextBlob-Bi-LSTM method is proposed for sentiment analysis and the tweets are classified here based on sentiments extracted by TextBlob using a deep learning Bi-LSTM classifier. TextBlob is a lexicon based tool which is interpretable and easy to use but lack adaptability. Bi-LSTM is used to improve the context learning through bidirectional layers but struggles with parallel feature extraction. Combining these two in a hybrid model allows us to exploit the semantic understanding of TextBlob and the contextual learning power of Bi-LSTM. This proposed hybrid model achieves the accuracy of 89.3% and this hybrid model typically performs 2–5% better in accuracy compared to LSTM depending on the common dataset.

Keywords: Bi-LSTM; TextBlob; Sentiment Analysis; Classification;

1. Introduction:

Sentiment analysis, a subset of natural language processing, utilizes machine learning algorithms to examine text. This approach has gathered significant interest from developers and researchers, who have successfully used it to identify text polarity with accuracy. Sentiment analysis has been applied to various text sources, with Twitter being a popular platform for sentiment analysis due to its abundance of user-generated content [1]. Sentiment analysis applies text analysis and computational linguistics to identify, extract, and classify subjective information from unstructured text [2]. Its main goal is to determine the polarity of sentences, often using word clues extracted from sentence context.

Sentiment analysis plays a crucial role in generating valuable insights from unstructured data sources like tweets and reviews. In the business sector, companies use sentiment analysis to better understand customer feedback on products or services [3]. With the rise of social media platforms like Twitter, Facebook, Instagram, blogs, and news websites, people have unprecedented opportunities to share their opinions, making sentiment analysis an essential tool for capturing and analyzing public sentiment at scale [4].

As for sentiment analysis of tweets, the key task is to classify the divergence between opinion-bearing tweets as either negative or positive. Sentiment analysis of tweets comes with its challenges. Although sentiment analysis has many uses and applications across many domains, it also presents a number of problems and difficulties with regard to Natural Language Processing (NLP). Technical and theoretical complexity continues to affect emerging sentiment analysis investigation, limiting its inclusive accuracy in sentiment identification [5].

Objectives: In NLP, sentiment analysis plays a vital role in understanding human opinions from text. Since deep learning models like Bi-LSTM are powerful in learning complex patterns from sequence data, rule-based models like TextBlob offer quick and interpretable sentiment features. A hybrid model is proposed here that combines the linguistic strength of TextBlob with the deep contextual learning of a Bidirectional Long Short-Term Memory (Bi-LSTM) network to enhance sentiment classification performance

2. Related Works:

There is a wide scope of analyzing sentiments in the field of text classification. Social media platforms enable users to create and share a wide variety of content, which can be distributed to others in real-time from any location. A MS-BSLI multistep method was developed [6] for building bilingual sentiment lexicons by integrating lexical constraints into word vectors. It uses lexical resources from a dominant language and a deep feed-forward network to transfer sentiment information to low-resource languages.

Two primary methods have been used to analyze sentiment: the machine learning approach and the lexicon approach [7]. To evaluate and ascertain the polarity of an opinion, the lexicon technique makes use of a glossary of terms that are both positive and negative. Dictionary-based and corpus-based methods are two more categories for this. The ML algorithms are used for the classification of text data into predetermined classes using lexical and structural features [8]. In recent years, several studies have proposed deep learning based sentiment evaluations, with differing features and performance. In order to solve different sentiment analysis problems (such sentiment polarity and aspect-based sentiment), this study looks at recent study utilizing methods based on deep learning such as CNN, RNN and DNN [9]. Deep learning models and word embedding are used to apply the most sophisticated DL-based techniques for sentiment analysis to Twitter datasets [10].

Sentiment analysis of natural language text is a key task in NLP, and CNNs have recently gained attention for their effectiveness in this area. This method applies CNNs with various configurations to perform SA on Hindi movie reviews collected from online sources. The manually annotated dataset is evenly split for training and testing [11]. A novel Fuzzy Graph

Convolutional Network (FGCN) was proposed for SLSA by integrating fuzzy logic into GCNs to handle sentiment ambiguities. First, BERT+BiLSTM generate contextualized vectors, which are then fuzzified. A sentence adjacency matrix based on dependency trees is also fuzzified and later defuzzified for feature extraction. Finally, the fuzzy vectors and fuzzy adjacency matrix are combined and passed through GCN layers to capture high-level sentence features [12]. For improved performance, this model combines deep learning and machine learning methods. Compared to text alone, emoticons were found to be a significant factor in shaping the polarity of sentiments. However, the results were not entirely representative because this model was limited to one domain and one language (English). For sentiment classification, Artificial Neural Networks (ANNs) is used due to their adaptive learning capabilities [13].

3. Sentiment Analysis using Hybrid TextBlob-Bi-LSTM Model:

This proposed work aims to develop a novel approach for improvising the sentiment classification of tweets by using hybrid approach TextBlob-Bi-LSTM. This proposed model involves Data loading and processing; word embeddings; Model building and concatenation; Classification and evaluation.

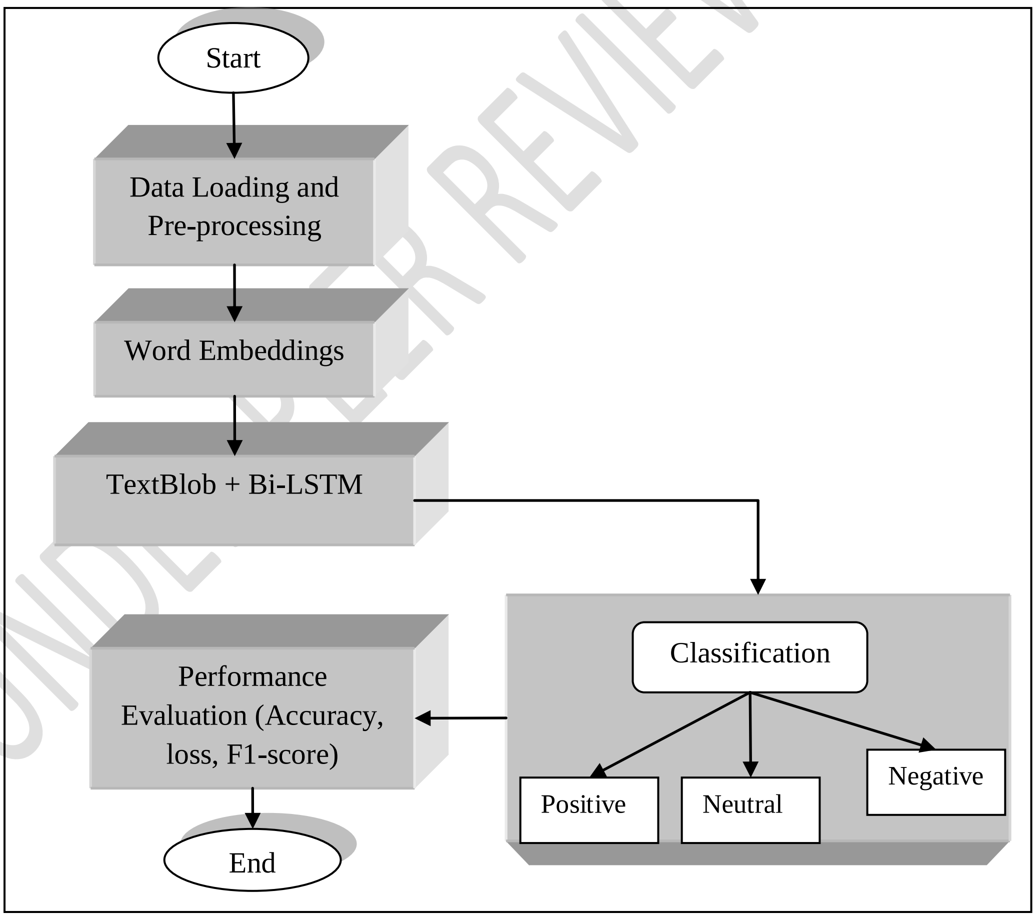

Figure 1: Work flow diagram of Proposed Model

1. **Input layer:** Word Embedding + TextBlob Features. TextBlob, a lightweight Python library, provides sentiment scores based on polarity and subjectivity using a pre-defined lexicon and rule-based grammar. The polarity might ranges from -1 (negative) to +1 (positive) and the subjectivity might ranges from 0 to 1.

2. **Bi-LSTM Layer:** Analyzes the sequential dependencies in the feature-rich representations provided by TextBlob, capturing contextual information from both directions.

3. **Concatenation Layer:** This layer combines Bi-LSTM output with TextBlob sentiment scores**.**

4. **Dense and Output layers:** Dense layer combines the feature vector and Outputs the final sentiment classification based on the combined feature representations.

a. Input layer:

Word embeddings and emotion scores from TextBlob are the two feature types that are taken in by the input layer. Each word in the text is converted into a dense vector representation which represents semantic relationships by word embeddings model called GloVe. GloVe learns word embeddings by factorizing a co-occurrence matrix. This model optimizes word vectors to reflect co-occurrence statistics globally and captures subtle global relationships like analogies better. In addition, TextBlob offers numerical measures of sentiment at the phrase level, such as subjectivity and polarity. Concatenating these characteristics with the embeddings enables the model to integrate emotional and contextual inputs immediately.

Each input sequence is tokenized into words and each word wi is converted into an embedding vector $e_i \in R^d$ where'd' is the embedding dimension. Let the sentence contain T tokens and given in equation 1,

$$E = [e_1, e_2, \dots e_T] \quad (1)$$

TextBlob provides both polarity score $p \in [-1, 1]$ and subjectivity score $s \in [0, 1]$.

b. Bi-LSTM Layer:

The Bi-LSTM extends the basic LSTM by processing input in two directions, both forward and backward, allowing it to capture both past and future dependencies. This is particularly useful when the sequence of the data is important for understanding its context.

The model may take into account both preceding and succeeding context thanks to Bi-LSTM. This is essential when performing sentiment analysis because the meaning of a word can be significantly impacted by the words that surround it (e.g., "not bad" vs. "bad"). Compared to a normal LSTM, Bi-LSTM captures a deeper comprehension of the sentence structure and meaning by combining forward and backward hidden states. For each time step 't', the forward

LSTM hidden state is given in equation 2 and the backward LSTM hidden state is given in equation 3,

$$\overrightarrow{h_t} = LSTM\ (x_t, \overrightarrow{h_{t-1}}) \tag{2}$$

$$\overleftarrow{h_t} = LSTM\ (x_t, \overleftarrow{h_{t-1}}) \tag{3}$$

Therefore the combined hidden state is given in equation 4,

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{4}$$

c. TextBlob+Bi-LSTM Layer:

The Bi-LSTM output is concatenated with the subjectivity and polarity sentiment ratings from TextBlob. By combining rule-based sentiment features with learnt sequential patterns, the network is able to provide a more informative representation. It enhances the network's capacity to detect sentiment, particularly in equivocal or mixed-tone inputs, by bridging the gap between data-driven and rule-based learning. The final Bi-LSTM hidden representation with the global sentiment features from TextBlob is given in the equation 5,

$$z = [h_{final}\, ; p; s] \tag{5}$$

The combined feature vector from the preceding stage is sent to the dense (fully connected) layer. To identify high-level abstract patterns in the data, it applies transformations applying learnt weights and non-linear activation functions such as ReLU. These layers serve as classifiers, gradually reducing to the most predictive features for the final classification after learning the various feature combinations correspond with sentiment labels.

$$\sigma(x) = \max(0, x) \tag{6}$$

The output layer is responsible for generating the final sentiment prediction. For binary classification (e.g., positive/negative), a sigmoid activation function is used to output a probability between 0 and 1. For multi-class classification (e.g., positive, neutral, negative), a softmax function is used to distribute probabilities across classes. This layer interprets the processed features from earlier layers and outputs the final sentiment class prediction.

For **binary classification** (positive/negative sentiment), a **sigmoid** activation is used and it is given as

$$\hat{y} = \frac{1}{1+e^{-(w_2^T a_1 + b_2)}} \tag{7}$$

4. Results and Discussion:

The proposed model is implemented using python 3.12.0 and the dataset used is Amazon product reviews. It begins by loading many key libraries for working with text, processing data, displaying it, and creating machine learning models. Several essential Python libraries are used in the sentiment analysis model's implementation. The dataset is loaded and manipulated using Pandas and NumPy, and data distributions and trends are visualized with the help of Matplotlib and Seaborn. Essential natural language processing activities are carried out using the Natural Language Toolkit (NLTK).

Model evaluation and data preprocessing are supported by Scikit-learn (sklearn). Word embeddings that represent the semantic links between words are created using Gensim. Lastly, the neural network models for sentiment categorization are constructed and trained using TensorFlow and Keras.

Table 1: Obtained values of Accuracy, Loss and F1-Score values

Model	**Precision, Loss and F1-score values**		
	Accuracy	***Loss***	***F1-score***
LSTM	84.2%	0.38	0.84
Bi-LSTM	86.5%	0.33	0.86
TextBlob+Bi-LSTM	89.3%	0.29	0.89

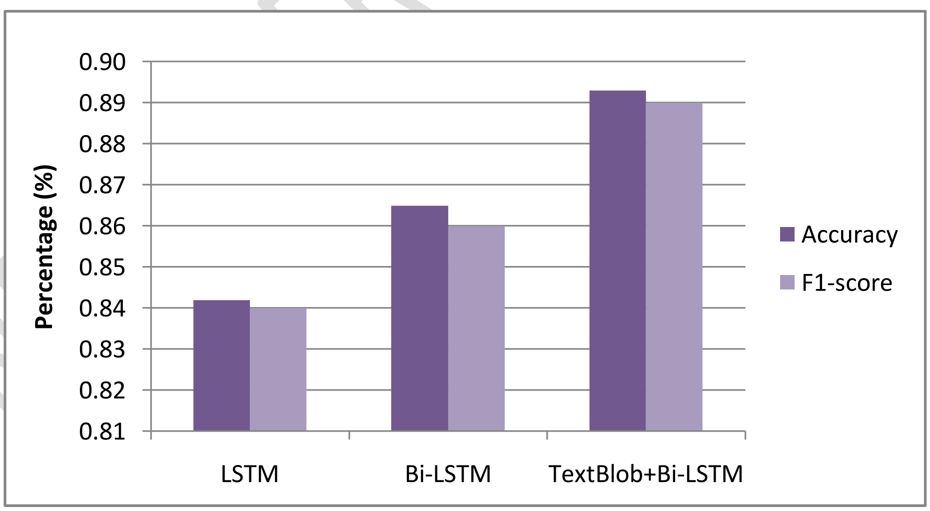

Figure 2: Accuracy and F1-Score for LSTM, Bi-LSTM and TextBlob-Bi-LSTM Models

Figure 2 shows the graphical representation of obtained values of accuracy and F1-score for the existing LSTM, Bi-LSTM and the proposed TextBlob-Bi-LSTM Models. Table 1 provides the obtained values of Accuracy, Loss and F1-Score values for both conventional and proposed models.

5. Conclusion

The proposed hybrid TextBlob-Bi-LSTM model effectively combines the interpretability of lexicon-based sentiment scoring with the deep contextual learning capabilities of Bi-LSTM networks. By using TextBlob's sentiment scores as additional features, the model improves sentiment prediction accuracy, particularly for tweets that contain fine or mixed emotions. Experimental results demonstrate that this hybrid approach outperforms traditional LSTM models by 5%, achieving an accuracy of 89.3% on benchmark datasets. Thus, the integration of rule-based TextBlob-Bi-LSTM model presents a robust and efficient solution for sentiment analysis tasks in social media contexts.

References:

1. Rasool, G., & Pathania, A. (2021). Reading between the lines: untwining online user-generated content using sentiment analysis. Journal of Research in Interactive Marketing, 15(3), 401-418.
2. Coletta, L. F., da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014, October). Combining classification and clustering for tweet sentiment analysis. In 2014 Brazilian conference on intelligent systems (pp. 210-215). IEEE.
3. Bharadwaj, L. (2023). Sentiment analysis in online product reviews: mining customer opinions for sentiment classification. Int J Multidiscip Res, 5(5).
4. Xu, Q. A., Chang, V., & Jayne, C. (2022). A systematic review of social media-based sentiment analysis: Emerging trends and challenges. Decision Analytics Journal, 3, 100073.
5. Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). Affective computing and sentiment analysis. A practical guide to sentiment analysis, 1-10.
6. Chang, C. H., Hwang, S. Y., & Wu, M. L. (2021). Learning bilingual sentiment lexicon for online reviews. Electronic Commerce Research and Applications, 47, 101037.
7. Nigam, N., & Yadav, D. (2018, April). Lexicon-based approach to sentiment analysis of tweets using R language. In International Conference on Advances in Computing and Data Sciences (pp. 154-164). Singapore: Springer Singapore.
8. Mitra, A., & Mohanty, S. (2021). Sentiment analysis using machine learning approaches. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2 (pp. 63-68). Singapore: Springer Nature Singapore.
9. Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. Electronics, 9(3), 483.
10. Abdullah, T., & Ahmet, A. (2022). Deep learning in sentiment analysis: Recent architectures. ACM Computing Surveys, 55(8), 1-37.
11. Rani, S., & Kumar, P. (2019). Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering*, *44*(4), 3305-3314.
12. Phan, H. T., & Nguyen, N. T. (2024). A fuzzy graph convolutional network model for sentence-level sentiment analysis. IEEE Transactions on Fuzzy Systems, 32(5), 2953-2965.
13. Thakkar, A., Mungra, D., Agrawal, A., & Chaudhari, K. (2022). Improving the performance of sentiment analysis using enhanced preprocessing technique and artificial neural network. IEEE transactions on affective computing, 13(4), 1771-1782.