EAI Endorsed Transactions Preprint Research Article/Editorial

Sentiment analysis using machine learning and deep learning algorithm

Pham Quang Tu[1]

[1]Faculty of Electrical and Electronic Engineering, Phenikaa University, Yen Nghia, Hanoi, 12116, Vietnam

Abstract

Sentiment analysis plays a crucial role in understanding public opinion, customer feedback, and user sentiments in various applications. This report details a project involving the implementation of three machine learning models (logistic regression, random forest, naive bayes) and two deep learning models (LSTM, CNN-LSTM Network) on two public datasets for sentiment analysis—IMDB 50k movie reviews and IMDB 5k Sentiment Analysis. The results highlight the performance variations among these models and their efficacy in predicting sentiment from text data.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning

doi:xx.yyy/trans.journalid.V.n.i

1. Introduction

Sentiment analysis, also referred to as opinion mining, plays a pivotal role in the field of natural language processing (NLP) by providing insights into the sentiments expressed in textual data. In the digital age, the ability to understand public opinion, customer sentiments, and social media reactions has become increasingly vital for businesses, researchers, and decision-makers. This report focuses on the application of diverse machine learning and deep learning models in sentiment analysis, utilizing two widely-used public datasets—the IMDB 50k movie reviews and IMDB 5k Sentiment Analysis.

Understanding sentiments in text data is a multi-faceted challenge due to the dynamic and nuanced nature of human language. Sentiments can range from positive and negative to neutral, and extracting these sentiments accurately requires sophisticated algorithms. This report contributes to the existing body of knowledge by exploring and comparing the performances of three machine learning models—logistic regression [1], random forest [2], and naive bayes [3]—and two deep learning models—Long Short-Term Memory (LSTM) [4] and Convolutional Neural Network and LSTM combined (CNN-LSTM) [5]—on the specified datasets.

2. Related work

Sentiment analysis, a crucial task in natural language processing (NLP), has witnessed significant advancements with the adoption of deep learning techniques. Several scholarly works have extensively explored the application of deep learning models in sentiment analysis tasks, as highlighted in the following studies.

Chinmayee Sahoo, Mayur Wankhade, and Binod Kumar Singh conducted a comprehensive review on sentiment analysis employing deep learning techniques [6]. Their study delves into the utilization of recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models. Particularly, they investigated the effectiveness of RNNs, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), in capturing sequential dependencies in textual data. Furthermore, they discussed recent advancements in sentiment analysis achieved through transformer architectures.

Another survey by [7] provides a detailed evaluation of deep learning techniques employed in sentiment analysis tasks. The most prevalent deep learning methods include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks. These techniques are utilized either in combination or independently depending on the application domain.

In a comparative study by [8], recent research employing deep learning for sentiment analysis is assessed. Models utilizing term frequency-inverse document frequency (TF-IDF) and word embeddings have been applied across various datasets, demonstrating the versatility and efficacy of deep learning methodologies in sentiment analysis tasks.

Furthermore, a survey conducted by [9] offers an overview of deep learning techniques and provides a comprehensive survey of its current applications in sentiment analysis. The study highlights the diverse range of applications leveraging deep learning methods for sentiment analysis tasks.

In summary, the reviewed literature underscores the pivotal role of deep learning techniques in sentiment analysis, showcasing their effectiveness across various datasets and domains.

3. Methodology

3.1. Data Preprocessing

The first crucial step in preparing the textual data for sentiment analysis involves tokenization using the Spacy and Keras Tokenizer libraries. Spacy, a natural language processing library, is employed for its robust tokenization capabilities, breaking down raw text into individual words and capturing essential linguistic features. Simultaneously, the Keras Tokenizer is utilized to convert these tokens into numerical representations by assigning a unique index to each word. This process not only facilitates model input but also helps capture the semantic relationships between words.

Simultaneously, the Keras Tokenizer is utilized for its simplicity and efficiency in converting tokenized words into numerical indices. This step is crucial for facilitating the input of words into machine learning and deep learning models. Keras pad sequences is then applied to ensure uniformity in sequence lengths. This involves padding sequences shorter than a specified length and truncating longer sequences, enabling consistent input dimensions for subsequent model training.

3.2. Logistic Regression [1]

Logistic Regression serves as a foundational machine learning model for sentiment analysis. The model takes the numerical representations of words as input and is trained to predict sentiment labels. During training, the weights assigned to features are adjusted to minimize a logistic loss function. Regularization techniques, such as L2 regularization, are applied to prevent overfitting by penalizing large weight values. Hyperparameter tuning involves optimizing the learning rate, a crucial factor in controlling the step size during optimization. The model is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score.

3.3. Random Forest [2]

Random Forest, an ensemble learning technique, is employed to enhance the accuracy of sentiment predictions. The model is trained on the preprocessed data, and during training, multiple decision trees are constructed. Each tree is trained on a random subset of the data, contributing to the model's diversity and robustness. Hyperparameters, including the number of trees and maximum depth of each tree, are fine-tuned to optimize the model's performance. Evaluation metrics include accuracy and the Gini impurity index, which quantifies the quality of splits in the decision trees. The aggregated predictions from multiple trees contribute to the final sentiment prediction.

3.4. Naive Bayes [3]

Naive Bayes, a probabilistic model, is based on Bayes' theorem and assumes independence between features. In sentiment analysis, it is trained on the preprocessed data using the feature-label pairs from the training set. Laplace smoothing is applied to handle cases where certain features may not be present in the training set, preventing zero probabilities. The model calculates the probability of a document belonging to a particular sentiment class, and the class with the highest probability is assigned. Evaluation involves standard classification metrics, providing insights into the model's ability to make probabilistic predictions based on observed features.

3.5. LSTM [4]

In this study, an LSTM layer with 256 units was incorporated after the embedding layer for the Long Short-Term Memory (LSTM) model. This architectural choice aimed to introduce a sufficient number of units to capture and learn complex sequential patterns present in the textual data.

The LSTM layer, a type of recurrent neural network (RNN), is well-suited for tasks involving sequential data processing due to its ability to retain long-term dependencies and mitigate the vanishing gradient problem. By introducing 256 units in the LSTM layer, the model was empowered to analyze and extract intricate relationships between words and phrases, enhancing its capacity to comprehend sentiment expressions within the text.

The embedding layer preceding the LSTM layer facilitated the transformation of tokenized words into dense vectors, thereby representing each word in a continuous vector space. This embedding process facilitated the model's ability to learn and generalize patterns present in the textual data, contributing to its overall performance in sentiment analysis tasks.

The inclusion of the LSTM layer with 256 units represents a deliberate architectural decision aimed at enhancing the model's ability to capture and understand the nuanced semantic structures inherent in sentiment-laden text. This methodology choice aligns with best practices in deep learning architecture design

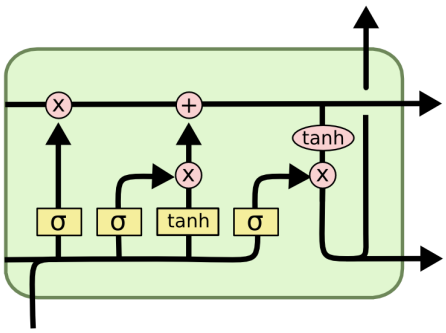

Figure 1. LSTM Structure diagram

and contributes to the robustness and efficacy of the LSTM model in sentiment analysis applications.

3.6. CNN-LSTM Networks [5]

In addition to the LSTM model architecture previously described, a Convolutional Neural Network (CNN) coupled with LSTM was explored to further enhance sentiment analysis performance. This hybrid model integrates the strengths of CNNs in feature extraction with the sequential understanding capabilities of LSTMs.

The CNN + LSTM architecture consists of two Conv1D layers followed by two MaxPooling1D layers. These Conv1D layers are responsible for convolving over the input tokenized sequences to extract local features effectively. By leveraging filters of different sizes, the CNN can capture patterns at various scales within the text data. The MaxPooling1D layers subsequently reduce the dimensionality of the feature maps, retaining the most salient features while discarding redundant information.

The output of the MaxPooling1D layers is then fed into two LSTM layers with 128 units per layer. The LSTM layers provide the model with the ability to capture long-range dependencies and contextual information present in the text data. This sequential processing aspect complements the local feature extraction capabilities of the CNN, enabling the model to capture both short-term and long-term relationships within the text.

Following the LSTM layers, the feature representations are passed through two fully connected layers, with the numbers of units being 64 and 16, respectively. Batch normalization (BatchNorm) is applied to normalize the activations of the fully connected layers, stabilizing the training process and accelerating convergence. Dropout regularization is also employed to mitigate overfitting by randomly dropping a fraction of the neurons during training, encouraging the network to learn more robust and generalizable representations.

Finally, a classification layer is added to the model, which performs sentiment prediction based on the learned features. The classification layer outputs the predicted sentiment label for each input sequence, thereby completing the sentiment analysis task.

By integrating CNNs for feature extraction and LSTMs for sequential understanding, the CNN + LSTM model offers a comprehensive approach to sentiment analysis, effectively capturing both local and global patterns within the text data. The inclusion of BatchNorm and Dropout further enhances the model's generalization capability and prevents overfitting, contributing to improved performance on sentiment analysis tasks.

4. Result

Table 1. Models' performance comparision

Model	IMDB 50k		IMDB 5k	
	Train	Valid	Train	Valid
Logistic Regression	51.60%	50.86%	55.40%	50.70%
Random Forest	99.99%	53.99%	100.0%	47.90%
Naive Bayes	51.52%	50.31%	50.80%	55.80%
LSTM	100.0%	77.68%	100.0%	72.50%
CNN-LSTM	99.82%	81.01%	100.0%	71.13%

Table 2. Models' parameters and execution time

Models	Parameters	Time Execution (s)	
		IMDB 5k	IMDB 50k
LSTM	5.017.125	42.87	183.89
CNN-LSTM	4.913.189	87.02	274.36

The evaluation was conducted on two datasets: IMDB 50k, consisting of 50,000 samples, and IMDB 5k, containing 5,000 samples. This selection enables the assessment of model performance across datasets of varying sizes, addressing concerns related to dataset scale. During training, the datasets were split in a 6:2:2 ratio for training, validation, and testing, respectively.

Table 1 illustrates the performance of three machine learning models (Logistic Regression, Random Forest, and Naive Bayes) and two deep learning models (LSTM and CNN-LSTM) on both IMDB 50k and IMDB 5k datasets. Deep learning models notably outperform traditional machine learning methods. While machine learning models achieve an average validation accuracy of 50%, deep learning models achieve 71.5% and 79% accuracy on the IMDB 5k and IMDB 50k datasets, respectively. This highlights the superior learning capabilities of deep learning models in NLP tasks, particularly the LSTM model's ability to capture the

contextual meaning of entire sentences rather than focusing solely on individual words.

Among the machine learning models, the Random Forest model while capable of achieving nearly perfect accuracy (almost 100%) on the training set, exhibits a clear challenge—overfitting. This overfitting becomes evident as the model's accuracy sharply drops to approximately 50% on the validation set. This discrepancy highlights the difficulty the model faces in generalizing its learned patterns to new, unseen data.

Table 2 displays the number of parameters and training times for the two deep learning models across the two datasets. Despite CNN-LSTM having fewer parameters due to weight sharing between CNN and LSTM layers, its training time is longer than that of the LSTM model. This phenomenon is attributed to weight sharing within the CNN and LSTM layers, where the weights of filters in CNN are shared across spatial dimensions, thereby reducing the parameter count compared to utilizing fully connected layers. However, it is not necessary that a network with fewer parameters will necessarily train faster. Despite having fewer parameters, the CNN-LSTM model exhibits a significantly longer training time compared to the LSTM model. This prolonged training duration is primarily due to the more intricate network architecture resulting from the integration of CNN and LSTM layers. The computation involved in performing convolutions within CNN and temporal sequence computations within LSTM can contribute to increased computational time.

These two models exhibit comparable results, wherein, with a smaller dataset, the LSTM model demonstrates a slightly higher validation accuracy of nearly 1.4%. However, with larger datasets, up to 50k records, the CNN-LSTM model achieves a higher accuracy of approximately 3.5% on the validation dataset.

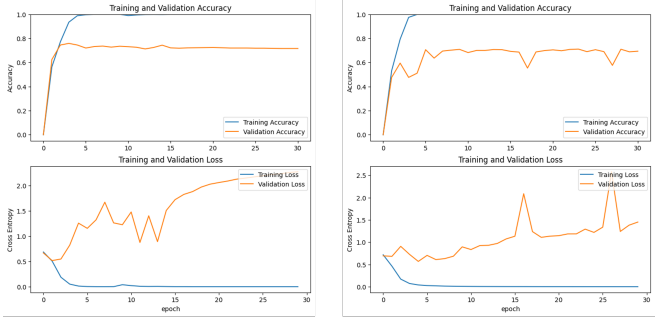

Figure 2. Deep learning models' performance in 5k dataset

Figure 2 and 3 reveal that LSTM models generally converge to stable validation accuracies more steadily than CNN-LSTM. However, LSTM models tend to have higher validation losses, which continue to increase during training. In contrast, CNN-LSTM models exhibit less stable convergence in validation accuracy, often peaking significantly higher than LSTM models but stabilizing to lower validation losses across both datasets.

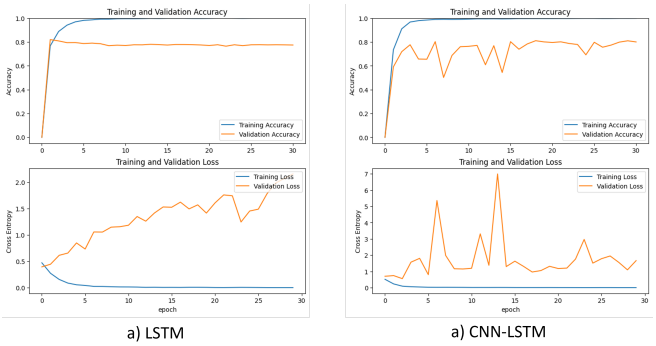

Figure 3. Deep learning models' performance in 50k dataset

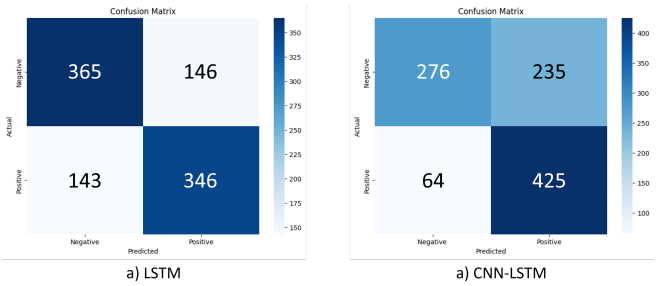

Figure 4. Deep learning models' confusion matrix in 5k dataset

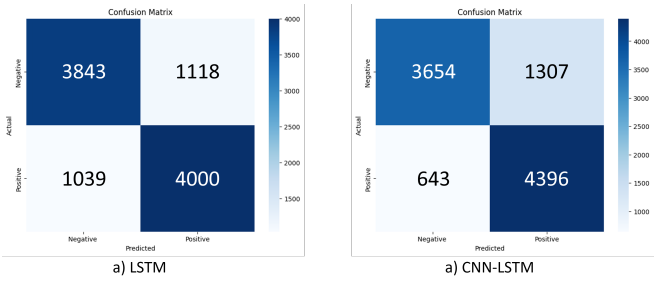

Figure 5. Deep learning models' confusion matrix in 50k dataset

Additionally, the analysis of confusion matrices in Figure 4 and 5 provides insights into the models' predictive behavior. The LSTM model tends to exhibit more balanced errors across both Positive and Negative labels. In contrast, the CNN-LSTM model often leans towards predicting more Positive instances, resulting in a substantially higher number of misclassifications for the Negative label compared to the LSTM model. However, this tendency also contributes to higher Precision and Recall scores for the Positive label, compensating for the increased misclassifications.

In summary, the results underscore the superior performance of deep learning models, particularly CNN-LSTM, in NLP tasks across datasets of varying sizes. While LSTM models may offer more stable convergence, CNN-LSTM models demonstrate greater potential for achieving higher accuracy, albeit with longer training times.

5. Discussion

The results obtained from the evaluation shed light on the efficacy of various machine learning and deep learning models in sentiment analysis tasks, particularly within the context of movie review datasets.

The performance comparison between traditional machine learning algorithms (Logistic Regression, Random Forest, and Naive Bayes) and deep learning architectures (LSTM and CNN-LSTM) underscores the superior capabilities of deep learning models in handling natural language processing (NLP) tasks. While machine learning models struggled to achieve an average validation accuracy of 50%, deep learning models consistently surpassed this threshold, achieving validation accuracies ranging from 71.5% to 79%.

The LSTM model, renowned for its ability to capture temporal dependencies in sequential data, exhibited stable convergence and balanced errors across Positive and Negative labels. In contrast, the CNN-LSTM model, integrating convolutional and recurrent layers, showcased higher predictive potential, albeit with longer training times and a propensity for over-predicting Positive instances. This tendency led to elevated misclassifications in the Negative label but compensated with higher Precision and Recall scores for the Positive label.

The analysis of parameter counts and training times revealed interesting trade-offs between model complexity and training efficiency. Despite having fewer parameters, the CNN-LSTM model necessitated longer training times due to its intricate network architecture. This observation highlights the importance of striking a balance between model complexity and training efficiency, especially in resource-constrained environments.

The findings have significant implications for real-world NLP applications, where accurate sentiment analysis plays a pivotal role in understanding user feedback and sentiment trends. While LSTM models may offer stability and interpretability, CNN-LSTM models demonstrate superior predictive power, especially in scenarios demanding nuanced sentiment analysis.

Future research could explore hybrid architectures that leverage the strengths of both LSTM and CNN components while mitigating their respective limitations. Additionally, investigating techniques to improve the efficiency of CNN-LSTM training without compromising predictive performance would be valuable. It's essential to acknowledge the limitations of this study, including the focus on sentiment analysis within a specific domain (movie reviews) and the potential biases inherent in the dataset selection and preprocessing procedures.

In conclusion, the study provides valuable insights into the comparative performance of traditional machine learning and deep learning models in sentiment analysis tasks, paving the way for further advancements in NLP research and applications.

6. Conclusion

The present study delved into the comparative analysis of machine learning and deep learning models for sentiment analysis using IMDB movie review datasets of varying sizes. Through rigorous experimentation and evaluation, several key findings emerged, shedding light on the efficacy, strengths, and limitations of different model architectures.

It is essential to acknowledge the limitations inherent in this study, including the focus on sentiment analysis within a specific domain (movie reviews) and the potential biases associated with dataset selection and preprocessing methodologies. Future studies could broaden the scope by encompassing diverse domains and datasets to ascertain the generalizability of the observed trends.

In conclusion, the study provides valuable insights into the comparative performance of machine learning and deep learning models in sentiment analysis tasks. By elucidating the strengths and weaknesses of different model architectures, this study contributes to the ongoing discourse in the field of natural language processing and sets the stage for future advancements in sentiment analysis research and applications.

References

[1] Joanne Peng, Kuk Lee, and Gary Ingersoll. An introduction to logistic regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, 96:3–14, 09 2002. doi:10.1080/00220670209598786.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1): 5–32, 2001. doi:10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[3] Vikramkumar, Vijaykumar B, and Trilochan. Bayes and naive bayes classifier, 2014.

[4] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019.

[5] Shuanglong Liu, Chao Zhang, and Jinwen Ma. Cnn-lstm neural network model for quantitative strategy analysis in stock markets. pages 198–206, 10 2017. ISBN 978-3-319-70095-3. doi:10.1007/978-3-319-70096-0_21.

[6] Chinmayee Sahoo, Mayur Wankhade, and Binod Kumar Singh. Sentiment analysis using deep learning techniques: a comprehensive review. *International Journal of Multimedia Information Retrieval*, 12(2):41, 2023. doi:10.1007/s13735-023-00308-2. URL https://doi.org/10.1007/s13735-023-00308-2.

[7] M. Indhraom Prabha and G. Umarani Srikanth. Survey of sentiment analysis using deep learning techniques. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pages 1–9, 2019. doi:10.1109/ICIICT1.2019.8741438.

[8] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3), 2020. ISSN 2079-9292. doi:10.3390/electronics9030483. URL https://www.mdpi.com/2079-9292/9/3/483.

[9] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis : A survey, 2018.