EAI Endorsed Transactions Preprint Research Article/Editorial

SENTIMENT ANALYSIS FOR MOVIES REVIEWS DATASET USING DEEP LEARNING MODELS(MID-TERM PROJECT)

Dinh Dat. Pham [1]

[1]Phenikaa University

Abstract

Due to the prolific generation, sharing, and dissemination of a vast volume of data and opinions daily through the internet and various media channels, sentiment analysis has become paramount in the evolution of opinion mining systems. This paper introduces an advanced sentiment analysis classification approach employing deep learning networks and presents comparative findings across different architectures. A Multilayer Perceptron (MLP) is established as a foundational benchmark, against which the results of other networks are gauged. Specifically, Long Short-Term Memory (LSTM) recurrent neural networks, LSTM-Attention, and a hybrid model amalgamating LSTM and CNN are developed and deployed on the IMDB dataset, comprising 50,000 movie reviews. The dataset is evenly partitioned into 50% positive reviews and 50% negative reviews for comprehensive analysis. In our experiments, CNN-LSTM models get the highest accuracy on IMDB datasets with 87.88% and LSTM-Attention has the best performance on the Tweeter dataset with a 95.20% accuracy score.

Keywords: Deep learning, LSTM, CNN, Sentiment Analysis, Movies Reviews, Binary Classification, Natural Language Processing

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

1. Introduction

Sentiment analysis, also known as opinion mining, is defined as the application of text analysis, computational linguistics, or Natural Language Processing (NLP) to achieve a semantic quantification of the scrutinized information [1]. The primary objective of sentiment analysis is to discern the opinion expressed in a specific text, such as a tweet or a product review. Decision-makers utilize these insights to inform their strategic planning and make well-informed decisions, including those related to marketing strategies, customer acquisition, and business expansion in specific geographic regions.

With the exponential growth of data and the constant exchange and production of information, the imperative to comprehend, mine, and analyze this vast volume of data has significantly intensified. Traditional machine learning techniques and conventional Neural Networks have proven inadequate for handling such extensive datasets. Consequently, deep learning has emerged as the cornerstone of the big data era, providing the requisite capabilities to effectively process and extract meaningful insights from large-scale datasets[2].

Deep learning, a subset of machine learning, represents an evolution of traditional neural networks. While a typical neural network comprises a single structure with input and output layers, along with hidden layers for computation, deep neural networks differ by incorporating multiple neural networks. In this configuration, the output of one network serves as the input to the next, creating a sequential hierarchy of interconnected networks. This innovation has effectively addressed the previous limitations imposed on the number of hidden layers in neural networks, rendering the processing of large datasets more feasible[3].

A distinguishing feature of deep learning networks is their ability to autonomously learn features from the data[4]. This characteristic has positioned deep learning as a robust machine learning technique capable of discerning multiple layers of features and generating predictive outcomes. The application of deep learning has gained prominence in various domains, particularly in signal and information processing, aligning with the surge in big data[4]. Moreover, deep learning networks have found utility in sentiment analysis and opinion mining, contributing to their versatility and widespread adoption.

2. Related works

In a published work on sentiment analysis, the authors undertook the classification of online product reviews into positive and negative categories. The study employed various machine learning algorithms to assess trade-offs, utilizing approximately 100,000 web-based product reviews. Three classifiers, namely the Passive-Aggressive (PA) Algorithm-Based Classifier, Language Modeling (LM) Based Classifier, and Winnow Classifier, were applied. Additionally, n-grams were employed for extracting linguistic features[5]. The results indicated that the use of high-order n-grams as features, combined with a robust classifier, can yield comparable or superior performance compared to what is reported in academic papers.

In another study, SVM demonstrated better classification performance with an accuracy of 82.9%, outperforming Naive Bayes, which achieved 81% accuracy on positive and negative movie reviews from the IMDB dataset (imdb.com). This dataset comprised 752 negative and 1301 positive reviews[6].

A third research endeavor proposed a Recursive Neural Tensor Network (RNTN) model for identifying sentences as positive or negative, utilizing fully labeled parse trees. The dataset used in this study consisted of 11,855 movie reviews. The RNTN achieved an accuracy of 80.7% in fine-grained sentiment prediction across all phrases, demonstrating improved accuracy in capturing negations of different sentiments and scope compared to previous models.

In a fourth paper focusing on customer reviews, the study collected product reviews from Amazon.com (11,754 sentences). It introduced a novel deep learning framework named Weakly Supervised Deep Embedding for sentence sentiment classification, achieving an accuracy of 87%[7].

3. Dataset and Preprocessing

3.1. Dataset

IMDB 50K Review Dataset: To construct appropriate models for categorizing movie reviews, we utilized the standardized dataset known as the "Large Movie Review Dataset," curated by Stanford researchers [8] [3] hereafter referred to as the "Stanford dataset." This dataset comprises 50,000 reviews sourced from IMDb, evenly divided into training and test sets, each containing 25,000 reviews. Each review is stored in an individual plain text file. Furthermore, both the training and test sets are carefully balanced to include 12,500 positive and negative reviews each. IMDb ratings of 7 or higher characterize positive reviews, while negative reviews have ratings of 4 or lower. Notably, neutral reviews were omitted from this dataset. Additionally, to prevent the influence of correlated ratings, the dataset limits the number of reviews for a single movie to a maximum of 30. Lastly, the training and test sets consist of distinct sets of movies, mitigating the risk of performance improvement through the memorization of movie-specific terms.

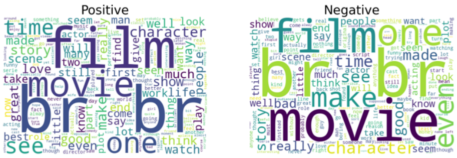
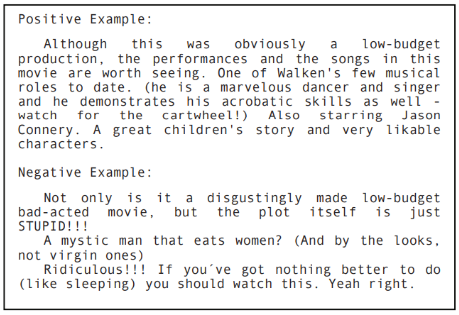


Figure 1. The presentation of word cloud on IMDB Movie Reviews dataset



Positive Example:

Although this was obviously a low-budget production, the performances and the songs in this movie are worth seeing. One of Walken's few musical roles to date. (he is a marvelous dancer and singer and he demonstrates his acrobatic skills as well - watch for the cartwheel!) Also starring Jason Connery. A great children's story and very likable characters.

Negative Example:

Not only is it a disgustingly made low-budget bad-acted movie, but the plot itself is just STUPID!!!
A mystic man that eats women? (And by the looks, not virgin ones)
Ridiculous!!! If you´ve got nothing better to do (like sleeping) you should watch this. Yeah right.

Figure 2. An example of IMDB dataset

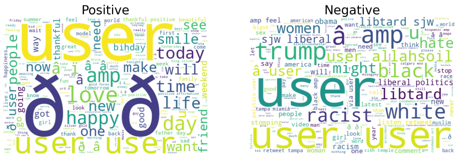


Figure 3. The presentation of word cloud on Tweeter Sentiment Analysis dataset

Twiter Dataset 10K: Twitter Sentiment Analysis Dataset 10K is based on the complete Twitter Sentiment140. It's comprised of a sample of 5000 positive and 5000 negative tweets.

4. Proposed Approach

The method proposed in this paper employs a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures for word-level sentiment classification using the IMDb review dataset. The CNN branches' outputs are subsequently inputted into an LSTM layer before being concatenated and forwarded to a fully connected layer, culminating in a single output as shown in Figure 4.

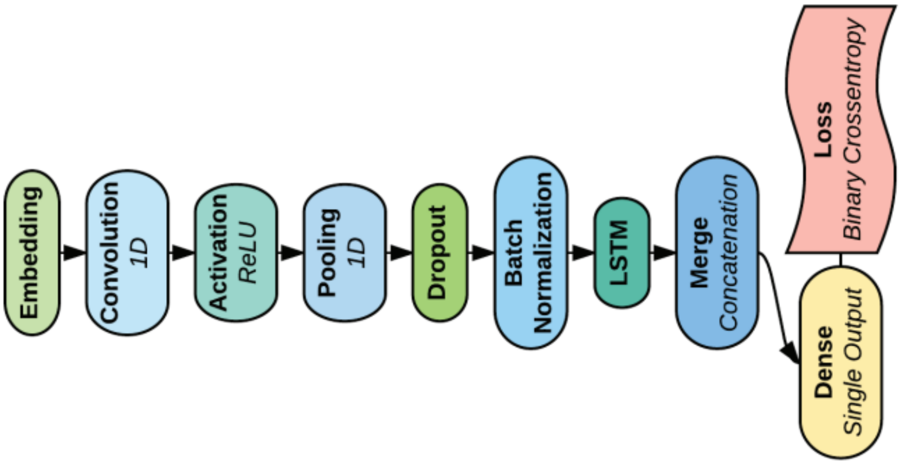


Figure 4. Our proposed basic CNN-LSTM model architecture

The network undergoes training and testing using mini-batches ranging from 16 to 128.

The initial layer of the network receives input reviews in the form of index sequences and embeds each word into a vector of a predefined size (e.g., a vector of 500 word indices embedded at 32 results in 500 vectors, each of length 32). This embedding process is facilitated by a trainable weight matrix, where matrix multiplication generates the vectors corresponding to each word index. Therefore, during training, the embedding layer improves upon the embeddings of each word.

4.1. Convolution

The output from the embedding layer is distributed to each of the b branches. Each branch commences with a 1-dimensional convolutional layer with a kernel size specific to that branch, denoted as c. The kernel utilized during 1-dimensional convolution is of dimensions c by the embedding size (c x e). Unlike typical 2-dimensional convolution, this operation convolves over entire words, rather than filtering partial widths or dividing words into segments. Multiple outputs are generated by employing multiple filters, denoted as f.

The CNN layer's objective is to examine word combinations within the specified kernel size c, enabling an understanding of how words interact with one another. For instance, when c equals 3, the layer considers combinations of 3 consecutive words, thereby establishing a sense of trigram combinations. The output of this layer is characterized by a shape determined by the input height and the number of filters (words by f).

4.2. Activation

Following the convolutional operation in each branch, a rectified linear unit (ReLU) activation function is applied to the output of the CNN layer. This activation function replaces any negative outputs with zero, thereby introducing non-linearity into the network. Consequently, the output shape of this layer remains identical to the input shape.

4.3. Maxpooling

Subsequent to the ReLU activation, each branch undergoes 1-dimensional max pooling, wherein each kernel size of the input is condensed into a single output representing the maximum observed value. This process results in a reduced, down-sampled version of the input. The primary objective of this layer is to mitigate overfitting while facilitating further processing. Similar to the CNN layer, the shape of the 1-dimensional max pooling kernel is tailored to the width of the data, such that the parameter kernel size, denoted as p, implies a kernel shape of p by the data width. This approach ensures that pooling is executed with an understanding that the data comprises complete words. Consequently, the output of this layer is a reduction in height proportional to the kernel size p (i.e., input height divided by p).

4.4. Dropout

Following the max pooling operation, each branch is subjected to a dropout layer, which randomly zeroes out a fraction of the inputs. Specifically, the dropout is applied to a designated fraction, denoted as d, of the inputs. This layer's purpose is twofold: to combat overfitting and to encourage the network to generalize by not relying too heavily on specific input elements. Importantly, the output shape remains unchanged, maintaining equivalence to the input shape.

4.5. Batch Normalization

Subsequently, each branch is equipped with a batch normalization layer, which normalizes the distribution for each batch following the dropout operation. This normalization process aims to diminish internal covariate shift, thereby facilitating faster convergence during training. Notably, the output of this layer maintains the same shape as the input.

4.6. LSTM

The final layer for each branch is a LSTM layer with a specified number of units l. The LSTM is used because of the nature of sequential data. The layer's persistence allows knowledge of previous input (convoluted word combinations) to influence subsequent input. The output has a length of the number of units l.

4.7. Dense

The branches are finally merged through concatenation. The LSTM layers' outputs are combined in an array. The output shape of this layer is equal to the summation of the output of all the branches (lxb).

The last layer is a fully-connected layer from the concatenated input to a single output. The layer is followed by a simple sigmoid activation function to conform the output between 0 and 1. The final yield is a single output.

4.8. Loss Function and Optimizer

The network is compiled using a binary cross-entropy loss function, which computes the loss based on two classes: 0 and 1. In the context of this paper, 0 denotes negative sentiment while 1 signifies positive sentiment. The loss is computed using the single final output of the dense layer.

Additionally, the network is compiled with an optimizer. Specifically, Adam, and RMSprop were employed as optimizers during the experimentation phase. Each optimizer was utilized with different learning rates and learning rate decay parameters.

5. Experiments

Numerous experiments were conducted to ascertain the optimal parameters and network architecture. Both the IMDb review sentiment dataset and the Tweeter 10K sentiment dataset were employed across all experiments. The datasets were preprocessed to encompass a dictionary size ranging from 5,000 to 10,000 words, with a maximum sequence length of 500 words per review, augmented with zero-padding where necessary. Beyond this threshold, additional data provided diminishing returns in terms of relevance to the network's objectives. Notably, the most effective network configuration utilized a batch size of 128. The subsequent sections delineate the refined parameters derived from the conducted experiments.

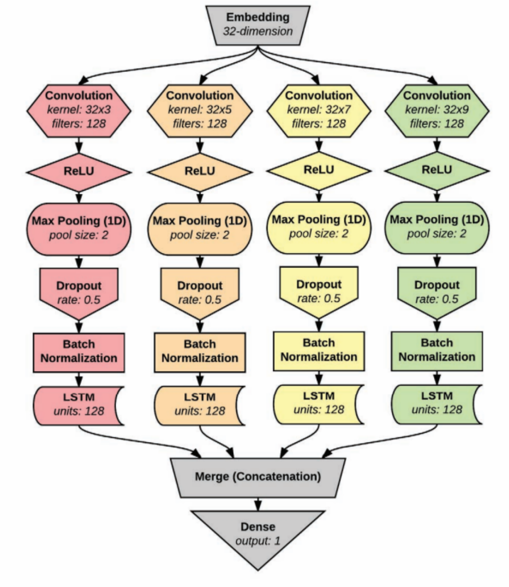


Figure 5. Diagram of the best performing proposed model

5.1. Embedding

An embedding size of 32 best fits the dataset. Vocabulary size was put directly in Embedding layer without using pre-trained embedding form GoogleNews(word2vec), FastText or Glove..

5.2. Convolution and Activation in Multiple Branches

The most significant adjustments were determined by the selection of the number of branches, which directly correlated with the chosen kernel sizes. Optimal 1-dimensional kernel sizes were identified as three, five, seven, and nine, leading to the establishment of four branches. Utilizing these kernel sizes proved instrumental in extracting pivotal information and enhancing accuracy. Additionally, the optimal number of filters was determined to be 128.

Furthermore, the convolutional layer was augmented with a ridge regression (l2) kernel regularizer to mitigate overfitting, with the l2 parameter set to 0.01.

The inclusion of the ReLU activation layer was pivotal in achieving heightened accuracy. Remarkably, this layer does not possess any parameters.

5.3. Max Pooling, Dropout, and Batch Normalization

While max pooling effectively addressed the primary concern of overfitting, it was observed that larger pooling sizes resulted in decreased accuracy. Through experimentation, the optimal kernel size was determined to be 2, effectively halving the height of the input.

The dropout layer emerged as the most effective means to combat overfitting, with a dropout rate of 0.5 applied to encourage other weights to contribute to the network's generalization. This approach not only facilitated convergence but also led to enhanced accuracy and a deeper comprehension of the data.

Batch normalization was integrated into the network to alleviate overfitting, particularly in conjunction with the LSTM network. Notably, this layer does not possess any parameters.

5.4. LSTM, Dense Layer, and Optimizer

Each LSTM layer within the branches comprises 128 units, as variations in this unit count were observed to either diminish accuracy or exacerbate overfitting. The merging and dense layers do not require tuning parameters.

Despite the optimizer not exerting a significant impact, RMSprop yielded the most favorable outcomes. Consequently, the learning rate was elevated to 0.01, while the learning rate decay was established at 0.1. These parameter configurations were determined to be optimal, resulting in the desired model performance.

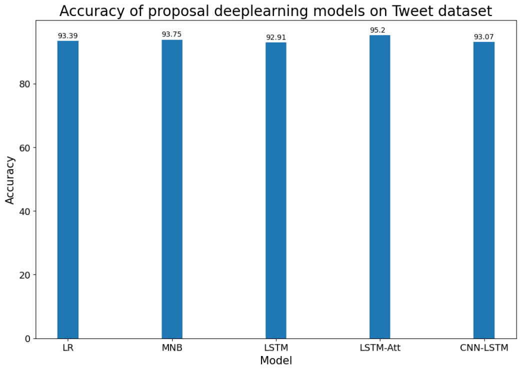


Figure 6. Performance of deep learning models in Tweeter dataset

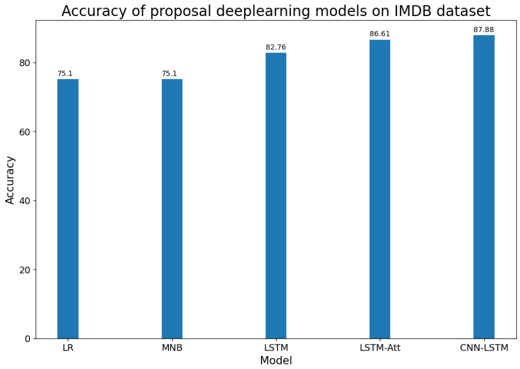


Figure 7. Performance of deep learning models in Tweeter dataset

6. Results and Analysis

Table 1. Evaluation on Test Set.

Model	IMDB dataset	Tweeter dataset
LR(BoW)	75.1	93.39
LR(TF-IDF)	75.09	92.83
MultinomialNB(Bow)	75.1	93.75
MultinomialNB(TF-IDF)	75.09	92.83
LSTM	82.76	92.91
LSTM-Attention	86.61	**95.20**
CNN-LSTM	**87.88**	93.07

In this study, the models are performed on the test sets of both data sets. In particular, the models are compared with each other according to the accuracy parameter for the binary classification problem as described in Table 1. It can be seen that machine learning algorithms have quite low results with the IMDB data set. only achieving 75.1% accuracy, however with the Tweeter data set, these two algorithms achieve better results with 93.39% and 93.75% respectively. With deep learning models, the performance of these models is significantly better than machine learning algorithms. The CNN-LSTM model has the best results on the IMDB data set, while the LSTM-Attention model has the highest accuracy on the Tweeter data set when reaching 95.20%. In Table 2, the models are compared in terms of a number of parameters. The CNN-LSTM model has the highest number of params with more than 6.5M params on the IMDB set and nearly 4.8M params on the Tweeter dataset, which is understandable for this model to achieve the best results on the IMDB dataset. Meanwhile, the two models LSTM and LSTM-Attention have approximately the same number of parameters with more than 3.6M params on the IMDB dataset and more than 1M params on the Tweeter dataset.

Table 2. Parameter of Deep learning model.

Model	Vocab Size	Dataset	Parameters
LSTM	121301	IMDB	3,644,295
LSTM-Attention	121301	IMDB	3,652,151
CNN-LSTM	88584	IMDB*	6,509,825
LSTM	35125	Tweeter	1,059,015
LSTM-Attention	35125	Tweeter	1,066,871
CNN-LSTM	35125	Tweeter	4,799,137

It can be easily seen that the complexity of the model depends greatly on the vocab size. With IMDB data, the large amount of data and variety of topics in movies leads to a lot of vocabulary in the data, so the vocab size of this set is up to 121,301 words and 35,125 in the tweeter dataset. The decrease in vocab size leads to a significant reduction in the number of model parameters. With the CNN-LSTM model, the number of model parameters is reduced from 6.5M to 4.7M. Both LSTM and LSTM-Attention models also have a difference between the two vocab sizes of up to 2.6M parameters.

7. Conclusion

In this study, we have researched and analyzed the sentiment analysis problem on two data sets IMDB Movie Reviews and Tweeter Sentiment Analysis. Machine learning algorithms and deep learning models are proposed for this problem. The results show that deep learning models demonstrate effectiveness in evaluating sentiment analysis. The proposed CNN-LSTM model has the highest results among all the models on the IMDB dataset. The LSTM-Attention model has the highest results on the Tweeter dataset. In future work, we will apply language models to further increase the model's accuracy. In addition, optimizing and selecting text features to use for machine learning algorithms to help speed up the analysis process is also the plan we aim for.

References

[1] J. Allen, "Natural language processing," pp. 1218–1222, 01 2006.

[2] A. L'Heureux, K. Grolinger, H. El Yamany, and M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. PP, pp. 1–1, 04 2017.

[3] P. Chitkara, A. Modi, P. Avvaru, S. Janghorbani, and M. Kapadia, "Topic spotting using hierarchical networks with self attention," 2019.

[4] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl. Based Syst.*, vol. 108, pp. 42–49, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:10276916

[5] H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'06. AAAI Press, 2006, p. 1265–1270.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, Jul. 2002, pp. 79–86. [Online]. Available: https://aclanthology.org/W02-1011

[7] Z. Guan, L. Chen, W. Zhao, Y. Zheng, S. Tan, and D. Cai, "Weakly-supervised deep learning for customer review sentiment classification," in *International Joint Conference on Artificial Intelligence*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:9036821

[8] Y. Kim, "Convolutional neural networks for sentence classification," 2014.