exeley INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.21307/ijssis-2021-011 Issue 1 | Vol. 14 (2021)

# An efficient sentiment analysis using topic model based optimized recurrent neural network

Nikhlesh Pathik* and Pragya Shukla

Institute of Engineering and Technology Devi Ahilya Vishwavidhyalaya, Indore, India.

*E-mail: pathiknikhlesh@gmail.com

This paper was edited by Piyush Kumar Shukla.

Received for publication February 21, 2021.

## Abstract

In recent years, topic modeling and deep neural network-based methods have attracted much attention in sentiment analysis of online reviews. This paper presents a hybrid topic model-based approach for aspect extraction and sentiment classification of textual reviews. Latent Dirichlet allocation applied for aspect extraction and two-layer bi-directional long short-term memory (LSTM) for sentiment classification. This work also proposes a hill climbing-based approach for tunning model hyperparameters. The proposed model evaluated on three different datasets. Compared to the single-layer Bi-LSTM model, the proposed model gives 95, 95, and 86% accuracy for the movie, mobile, and hotel domain, respectively.

**Keywords**
Bi-LSTM, LDA, Hill-climbing, Classification, Hyperparameters.

## Introduction

Nowadays, people are very talkative on the web. Due to the exponential growth in user feedback data, it becomes necessary for every product or service to do the mining of this feedback. Web opinion data are one of the main influential factors for purchasing products or services online. Web opinion data were the primary factor in decision-making during the last year when movements are restricted due to pandemic situations. To get the in-depth inside, the web opinion data aspect-based sentiment analysis (ABSA) has gained much attention in the last decade, and it continues a thrust area of research. Various studies have shown the effectiveness of recurrent neural network-based models for the sentiment classification task in the recent past. Different hybrid models are proposed, which use convolution neural network (CNN) and LSTM for the classification task.

Now the research question/direction that comes out is to propose an efficient deep neural network model for the sentiment classification task. So this work focuses on topic modeling and recurrent neural network-based approaches for ABSA. We have chosen LDA (Latent Dirichlet allocation), the most popular unsupervised topic model, and LSTM recurrent neural network. LDA is extensively used for unsupervised topic mining, and LSTM able to handle long-term dependencies. Following are the contribution of this paper:

1. A hybrid model based on topic modeling and recurrent neural network is proposed for sentiment analysis.
2. An efficient multi-layer Bi-LSTM is proposed for sentiment classification with only two stacked layers for keeping the model less complicated.
3. A hill climbing-based approach is proposed for tunning the hyper-parameter model to improve the proposed model's accuracy. Pre-trained embeddings like Glove is used to improve efficiency.
4. A comparative analysis using multiple datasets is presented, demonstrating the performance improvement of the proposed approach.

The main objective of this work is to present an efficient hybrid model for sentiment classification using topic modeling and recurrent neural network. Model hyperparameters are tunned using an incremental approach, and pre-trained embeddings are also used for further performance improvement. The novelty of the proposed work is its efficiency with less complexity.

The paper's remaining sections are as follows: the second section sheds light on the field's most recent work. The background and intuition of LDA, Bi-LSTM, and pre-trained embedding are provided in the third section. In the fourth section, the methodology and proposed algorithms are explained. Experimental details with results are described in the fifth section. With a summary and future direction, the sixth section concludes the paper.

## Related work

Sentiment analysis has got very much attention for more than one decade. Recently deep neural network-based approaches have gained popularity. Various hybrid models are presented using RNN, CRF, and topic modeling. This section discusses recent work done in the field of sentiment analysis of text reviews. Mainly LDA and RNN-based latest approaches are discussed along with their hybrid models.

Hameed et al. discuss a deep neural network-based model for sentiment classification with a single Bi-LSTM layer (Hameed and Garcia-Zapirain, 2020). Minaee et al. (2019) presented an ensemble approach based on LSTM and CNN for sentiment analysis with Glove pre-trained embedding. Rhanoui et al. (2019a) proposed an integrated CNN and Bi-LSTM model for document-level sentiment analysis with pre-trained Doc2vec embedding.

A deep neural network model approximates LDA to speed up the inference time (Zhang, D. et al., 2016). A novel approach is proposed using the CNN model with general-purpose embeddings and domain-specific embeddings of pre-trained embeddings for aspect extraction (Xu et al., 2018). A cosine similarity and Jensen–Shannon divergence are used to compute the similarity among topics and associate them into an aggregated model. The model is generated by the latent Dirichlet allocation and non-negative matrix factorization (Blair et al., 2020).

Bi-LSTM is used to analyze reviews through statistical analysis and sentiment classification (Huang et al., 2018a). The LSTM attention model for aspect-level sentiment analysis is proposed using embedding and common sense knowledge. An attention-based LSTM model is used for word sequence in a given document with a latent topic modeling layer. A tree-structured LSTM generates a semantic representation of the text (Zhang, W. et al., 2019).

A hybrid model based on LDA and LSTM has been given for COVID tweet analysis (Jelodar et al., 2020). The authors introduced LSTM based two models for aspect-based sentiment analysis (Huang et al., 2018b). One CRF-based character level model is for opinion target extraction, and the second attention-based sentence level model for classifying sentiment polarity.

A recommendation system using the topic model and DNN for the crowdfunding platform was developed (Shafqat and Byun, 2019). Two Bi-LSTM-based model in combination with 2D-Poling and 2D-CNN is given for sentiment analysis task (Zhou et al., 2016). A semantic similarity-based hybrid LDA model with LSTM is used for sentiment analysis of hotel reviews (Priyantina and Sarno, 2019). A hybrid approach was proposed (Luo, 2019) for sentiment analysis based on LDA and GRU-CNN. LDA is used for feature vector construction, and CNN with GRU is used for sentiment classification. A framework used topic modeling for finding rare named entities from text. This hybrid approach used LDA, LSTM, and CRF in an integrated way (Jansson and Liu, 2017). A CNN-Bi-LSTM-based hybrid model for document-level sentiment analysis is combined with the Doc2vec embeddings to improve the performance (Rhanoui et al., 2019b).

For short-text classification, an LSTM-based model is presented with Word2Vec (Wang et al., 2018). A model with a bidirectional recurrent neural network (RNN) with LSTM is implemented for recommendation and sentiment classification (Agarap and Grafilon, 2018). Pre-trained embeddings are used for training contextual semantics. A mortality prediction model was implemented from ICU admitted patient's clinical remarks using LDA and LSTM with simultaneous training and learning (Jo et al., 2017). A recurrent structure is applied for contextual information with a max-pooling layer that analyses suitable words for text classification to capture the key components in texts (Lai et al., 2015).

Coronavirus (COVID-19)-based tweets are analyzed with NLP and sentiment classification using RNN (Nemes and Kiss, 2021). Tweet classification is done based on coronavirus using ML-based Naïve Bayes method (Jim et al., 2020) and ML with LDA (Xue et al., 2020). The author proposed a Gaussian membership function implementation based on a fuzzy rule base to analyze sentiments from tweets on COVID-19 (Chakraborty et al., 2020). An AI-based method is used for COID-19 sentiments (Man et al., 2020).

The performance of the LSTM model significantly depends on the value of its hyperparameter. There is no well-defined rule for selecting the values of hyperparameters. Yadav et al. (2020) presented an incremental approach for tuning LSTM hyperparameters.

From the above study, it is observed that most of the models are developed using multi-layer Bi-LSTM and CNN. Some of the models use hybrid or ensemble models. This study proposes a research direction to develop an efficient hybrid model for the sentiment classification task with optimization of various model hyperparameters. A new model can be designed, which may give better accuracy with comparatively less complexity.

## Background

### Latent Dirichlet allocation (LDA)

Topic modeling is an unsupervised NLP technique representing a group of text documents with several topics that can best explain each document's underlying information. LDA primarily assumes that each document is a mixture of topics, and each word has a certain probability of falling into a particular topic (https://www.kaggle.com/rahulin05/sentiment-labelled-sentences-data-set). In LDA, every word in every document comes from a topic. The topic selects from a per document distribution over topics. The topic distribution $\Theta$ for every document is proportional to Dirichlet ($\alpha$), and the word distribution $\Phi$ is proportional to Dirichlet ($\beta$). Hyperparameters $\alpha$ and $\beta$ have a vital role in the generative process of LDA. Parameter $\alpha$ controls the documents topic concentration means. The small value of $\alpha$ represents the documents as a mixture of a few topics, whereas its high value results in more topics per document. Similarly, parameter $\beta$ controls topic word concentration. A low value of $\beta$ represents the topics with fewer unique words making it more distinct, and its high value results in more unique words in each topic.

The topic generation process of LDA depends on the following two probability distributions:

- $P(t|d)$ = The probability distribution of topics in documents $\Theta_{td}$.
- $P(w|t)$ = The probability distribution of words in topics $\Phi_{wt}$.

The utmost goal of LDA is to estimate the probability of a word given document, i.e. $P(w|d)$, with the help of the above two mentioned probabilities, which is provided by:

$$P\left(\frac{w}{d}\right) = \sum_{t\varepsilon T} P(w/d)P(t/d) \qquad (1)$$

It is the dot product of $\Theta_{td}$ and $\Phi_{wt}$ for each topic $t$ where:

Gibbs sampling is applied for successively sampling conditional distributions of variables. In the long run, distribution over states converges to the accurate one. The equation for the same is below:

$$p\left(z_{d,n} = k \mid \vec{z}_{-d,n,}\vec{w},\alpha,\beta\right) = \frac{n_{d,k} + \alpha_k}{\sum_i^k n_{d,i} + \alpha_i} X \frac{v_{k,w_{d,n}+\beta w_{d,n}}}{\sum_i v_{k,i} + \beta_i} \qquad (2)$$

where $n_{d.k}$ is the # document $d$ use topic $k$; $v_{k,w}$ is the # topic k uses the given word; and $\alpha$ and $\beta$ are the Dirichlet parameter for the document to topic and topic to word distribution, respectively (Blei et al., 2003).

### Long short-term memory (LSTM)

Cell-state and multiple gates are LSTM's central concepts. Cell-state transmits relative data through the succession chain. It is just like to network's memory. Important information retains in the cell state during sequence processing. To minimize the lack of memory effects, data coming from the previous time step leads to the next steps. The cell state moves when data are going to be added or removed via gates to that state. The gates are various neural networks that evaluate activated cell state data. Gate learns which data are necessary to keep or forget in the duration of training (http://colah.github.io/posts/2015-08-Understanding-LSTMs/).

### Forget gate

The first gate is called the 'forget gate', which determines what data to store and discard. Present state input and hidden state information pass via the sigmoid function that returns the values from 0 to 1. If the value is around 0, discard it, and if the value near 1, then store it.

### Input gate

The cell state changes through the gate. Next, the current input and the previous hidden state is transfer into a sigmoid function. We also convert current input and hidden state into the tan*h* function to squash the values from –1 to 1. Now multiply sigmoid output with tan*h* output. The sigmoid output specifies the retention of the needed information from the tan*h* output.

### Cell state

There is now ample information for cell state calculation. In the first case, multiply the cell state with the forgot vector. If values compound it close to 0, the cell state decreases the values. We then take the input gate's output and apply a point-wise addition to change the cell state to new values that the neural network can accept. This process gives a new cell state.

## Output gate

The last gate is named as output gate that defines the condition next hidden state. The hidden state contains previous input information and is used to predict future value. The hidden state and new input are then given to a sigmoid function. Then we transfer to the tan*h* feature the newly changed cell status. We raise the tan*h* output with the sigmoid output to determine which data will be in the hidden state. The new cell state and the hidden state will shift to the next level.

The equations for the gates in LSTM are:

$$\text{Input gate } I_t = \sigma(w_i[h_{t-1},x_t] + b_i) \tag{3}$$

$$\text{Forget gate } F_t = \sigma(w_f[h_{t-1},x_t] + b_f) \tag{4}$$

$$\text{Output gate } O_t = \sigma(w_o[h_{t-1},x_t] + b_o) \tag{5}$$

where sigmoid function is $\sigma$; weight for the respective gate neuron ($x$) = $w_x$; so $w_i$, $w_o$, and $w_f$ are the weight of input, output, and forget gate neuron; the output of the previous hidden state is at time $t$–1 is $h_t$–1; current state input, i.e., at time-stamp $t$, is $x_t$; biases for the respective gates is $b_x$; so $b_i$, $b_o$ and $b_f$ are the biases of input, output, and forget gate.

Candidate for cell state at time t:

$$C_t' = \tanh(w_c[h_{t-1},x_t] + b_c) \tag{6}$$

Cell state at time t is:

$$C_t = F_t \times C_{t-1} + i_t \times C_t' \tag{7}$$

$$\text{Final output at t } h_t = O_t \times \tanh(C_t) \tag{8}$$

## Bi-directional LSTM

A bidirectional LSTM or Bi-LSTM consists of two LSTMs: one heading in a forward direction and the other backward. It is a sequence processing model. BiLSTMs increase the amount of information available to the network efficiently and boost the algorithm's meaning (e.g., knowing what words to follow immediately and head a word in a sentence).

## Pre-trained embeddings (GLOVE)

In NLP, word embedding is vital for neural networks because of its brilliant ability to capture the semantics of words from massive unlabeled data. Word embedding can be used for polarity classification and also boost the performance of sentiment analysis models.

The pre-trained word embedding is an example of transfer learning. It is already trained on large datasets. So, instead of initializing our neural network weights randomly, these pre-trained embeddings are used to initialize weights. It helps to speed up training and improve the performance of NLP models. Pre-trained embeddings are used to reduce testing time, which enhances the effectiveness.

## Methodology used

In this work, the primary focus is on developing a hybrid approach for online review classification. This hybrid model is a combination of topic modeling, pre-trained embeddings, and multi-layer recurrent neural networks. For extracting thematic information from review data, the optimized LDA configuration is used. Feature extraction is done using LDA. Aspect expansion and categorization are done using frequent terms and domain knowledge. The Glove is used as pre-trained embedding, which reduces training time and improves the effectiveness most of the time.

Multi-layer stacked Bi-LSTM is used as a classifier. Bi-LSTM is an extensive version of traditional LSTM with improved performance on the sequence classification problem. Only two layers of Bi-LSTM are used so that model will not become more complex. Model hyperparameters are tuned using a hill-climbing-based approach.

Three algorithms are presented for various tasks perform in the proposed work. Algorithm 1 performs the aspect extraction and their categorization. Algorithm 2 performs LSTM hyper-parameter tuning for improving the accuracy of the model, and finally, Algorithm 3 represents the two-layer bidirectional LSTM model for sentiment classification. Figure 1 illustrates the flow of the proposed work.

**Algorithm 1: Aspect Extraction**
**Input:** Review dataset DS
**Output:** Aspect list with categories.

```
1. Input review corpus and pre-process it.
2. Tokenize pre-processed corpus into words.
3. Remove outliers from tokenized words.
4. Create a Bag of Words (BOW) representation.
5. Apply LDA on BOW to get topic-word probability
   distribution.
```

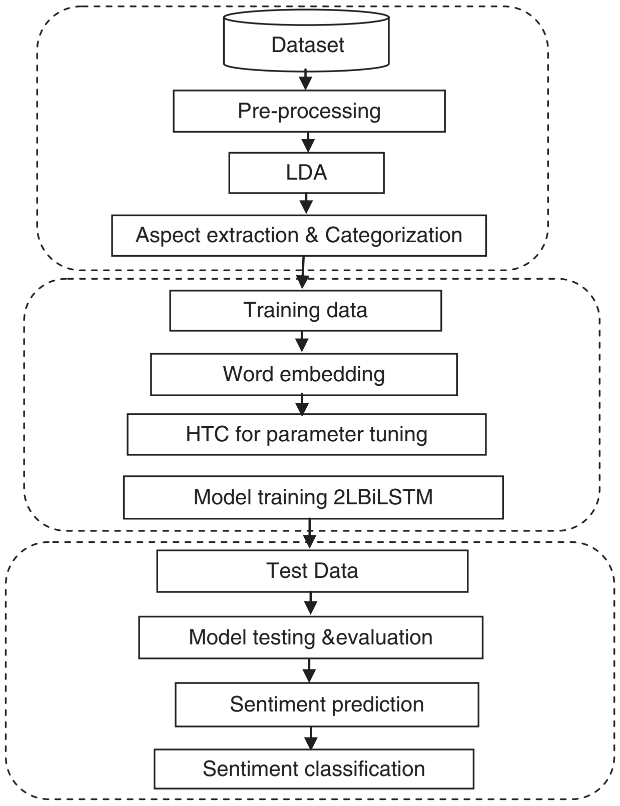

Figure 1: The flow of the proposed algorithm.

6. Apply POS rules on LDA output distribution to get the most probable aspects.
7. Categorization of aspects into various categories is done through domain knowledge and frequent cohesive terms.
8. Based on step 7, categories review sentences into the aspect category.

Algorithm 1 produces various categories of aspect sentences that are labeled as positive and negative. The aspect list generated using Algorithm-I expanded using frequent corpus terms and domain knowledge. The aspect expansion helps cover more reviews, and aspect categorization gives a better picture of various aspects. Now we pass these categories data into our RNN model for sentiment classification.

Bi-LSTM model's hyperparameters are tuned using hill-climbing tuning (HCT) algorithm. HCT picks the best direction in the hyperparameters space to choose the next hyperparameter value at each iteration. The optimization loop ends if no neighbor

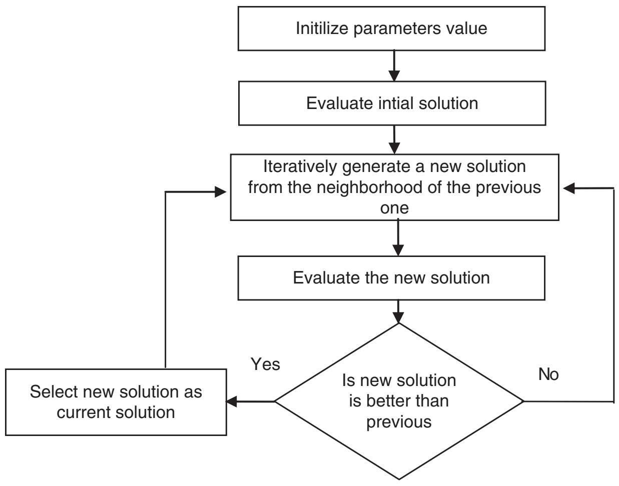
Initilize parameters value
Evaluate intial solution
Iteratively generate a new solution from the neighborhood of the previous one
Evaluate the new solution
Is new solution is better than previous
Yes
No
Select new solution as current solution

Figure 2: Flow of the proposed HCT algorithm.

enhances the accuracy of the model. Figure 2 shows the flow of algorithm 2.

The formal description of the HCT algorithm is as follows:

**Algorithm 2: HCT algorithm for hyper-parameter tuning**
**Input:** Test data (x_test, y_test), No of iterations *n*
**Output:** sol, out_val

```
1. initialize a list for out_val: list()
2. generate solution with random prediction on x_test: sol = ran_pred(x_test)
3. evaluate initial value & put on score list: eval_pred(y_test, sol)
4. append out_val
5. for i= 1 to n do:
6. append out_val // appending output value in a list
7. if out_val = 1 // already present in the list
8. break
9. candi = mod_pred(sol) // generating new candidate
10. val = eval_pred(y_test, candi) // evaluate
11. if val>= out_val // checking for the better value
12. sol, out_val = candi, val
13. Return sol, out_val
```

Algorithm 2 gives the optimal configuration for hyperparameter, which offers improved accuracy of the model. Now we can proceed with the classification task using our Bi-LSTM model, which is explained in Algorithm 3.

**Algorithm 3: Two-layer HCT Bi-LSTM Algorithm for sentiment classification.**

**Input:** Dataset DS, DS divides into training and testing, i.e., DSTR and DSTS.

**Output:** Sentiment Classification of test data

```
Start:

1.  for each category review, data DSTR do // Training of Bi-LSTM.
2.  Import word vector set W from pre-trained word embedding //(Glove 100d)
3.  Tuned model hyperparameters using HCT algorithm(Algorithm 2) // (learning rate, epoch etc.)
4.  Initialize Bi-LSTM model hyperparameters with optimal values from step 3.(classes, layers, epochs, document length, vocab)
5.  for each review sentence, s $\in DSTR_i$ do
6.  get the word vectors s = $[w_1, w_2, w_3, w_4..., w_{n-1}, w_n]$ of all words in s from W
7.  for Bi-LSTM forward pass, do
8.  forward_pass for
            i. f_state LSTM;
            ii. b_state LSTM;
9.  end of for
10. for Bi-LSTM Backward pass, do
11.     backward_pass for
            i. f_state LSTM;
            ii. b_state LSTM;
12. end of for
13. representation sequence generated by the memory cell = $[h_1, h_2, h_3, h_4..., h_{n-1}, h_n]$;
14. vector h is generated by a max-pooling operation;
15. The output layer obtains the sentiment class of the input sentence
16. end of for// Evaluating the model for test data
17. for each review sentence, s $\in DSSTS_i$ do
18. get the word vectors s = $[w_1, w_2, w_3, w_4..., w_{n-1}, w_n]$ of all words in s from W
19. for Bi-LSTM forward pass, do
20. forward_pass for
            i.  f_state LSTM;
            ii. b_state LSTM;
21. end of for
22. for Bi-LSTM Backward pass, do
23. backward_pass for
            i.  f_state LSTM;
            ii. b_state LSTM;
24. end of for
25. Representation sequence generated by the memory cell = $[h_1, h_2, h_3, h_4..., h_{n-1}, h_n]$;
26. Vector h is generated by a max-pooling operation;
27. The output layer obtains the sentiment class of the input sentence
28. end of for
29. classified test data.

End:
```

Steps 7 to 9 represent the forward pass for Bi-LSTM. Except the input sequence is presented in opposite directions to the two hidden layers, it is equivalent to unidirectional LSTM, and the output layer is not changed until both hidden layers have interpreted the whole input sequence. Steps 10 to 12 represent the backward pass for Bi-LSTM. Except that all the output layer and terms are first determined, then fed back to the two hidden layers in opposite directions for all *t*, Backward passes for the output layer in any order, storing and terms at each stage. It is comparable to unidirectional LSTM.

## Experimental setup and results

### Environment and system configuration

The Gensim implementation of LDA on the Anaconda platform is used for Python. The tests on Core I5 CPU @ 2.5 GHz 2.49 GHz with 8 GB of RAM were performed on Windows 8 OS. For implementing Bi-LSTM Model in Python, Tensor-Flow with Keras is used. For performing pre-processing in Python scikit-learn library is used.

### Datasets

We have used three different datasets for the evaluation of our algorithm. Kaggle has provided a labeled Sentences Data Set, publicly available by the University of California for Sentiment Analysis (https://www.kaggle.com/rahulin05/sentiment-labelled-sentences-data-set). Table 1 gives the detail about the datasets.

### Model detail

Algorithm 1 applies to these datasets, and after pre-processing, we have got frequent words that

**Table 1. Dataset statistics.**

| Dataset domain | Total | +ve | –ve |
|---|---|---|---|
| Restaurant from Yelp | 1,000 | 500 | 500 |
| Mobile from Amazon | 1,000 | 500 | 500 |
| Movies from IMDB | 1,000 | 500 | 500 |

Figure 3: Hotel dataset frequent terms.

are shown as word clouds in Figures 3-5. Dataset is tokenized into word and finally converted into the BOW. LDA operates on BOW and generates topic word probability distribution for supplied input. Linguistic rules (POS) applied to LDA output and extracted aspects from their probability distribution value. These aspects are stored for further process.

The sample distribution for the Mobile domain is as follows:

Figure 4: Movie dataset frequent terms.

Figure 5: Mobile dataset frequent terms.

T0: (0.025*"battery"+0.014*"one"+0.013*"month"+ 0.013*"screen"+0.012*"it"+0.012* "problem"+0.010* "buy"+0.010*"get"+0.010*"work"+0.009* "htc"')

T1: (0.015*"camera"+0.014*"phone"+0.010*"good-"+0.008*"price"+0.008*"it"+0.008* "working"+0.007-*"battery"+0.007*"well"+0.007*"work"+' '0.007* "htc"')

T2: (0.023*"screen"+0.012*"phone"+0.009*"battery"+ 0.008*"htc"+0.008*"get"+0.008*"work"+0.007*"it"+0.007 *"new"+0.007*"problem"+0.007*"would"').

In the above example, we have only shown three topics with 10 words in each. We can see that the 'battery' aspect is coming in multiple topics in the above distribution. We have considered its highest probability value. The same is applied to each aspect. Same we have applied on all three datasets for extracting aspects from the datasets. Based on the LDA output, i.e., topic probability distribution values and POS rules, the highest probability aspect is selected from each topic.

Algorithm 1 produces the aspect list based on the probability distribution values that are clustered into predefined categories. This categorization is done based on the domain knowledge and coherence between the terms based on specific topics. Review sentences are also clustered accordingly.

In total, 100 dimensional glove embeddings are used. GloVe word embeddings are generated from a vast text corpus like Wikipedia and can find a meaningful vector representation for each word in our dataset. It allows us to use transfer learning and train further over our data (https://www.kaggle.com/danielwillgeorge/glove6b100dtxt).

Before applying our Bi-LSTM-based classifier, its hyperparameter tunning is required. Algorithm 2 gives the optimal model hyperparameter configuration for better model performance. Table 2 represents the various model parameters used in our proposed Bi-LSTM model.

Model hyperparameters like learning rate and the number of epochs are tuned using the HCT algorithm. This optimized configuration is applied to three considered datasets for performance evaluation.

**Table 2. Various model parameters.**

| Parameter | Value |
|---|---|
| Vocabulary size | 10,000 |
| Bi-LSTM | 2 layer |
| Dense | 1 |
| Activation | Sigmoid |
| Optimizer | Adam function |
| Loss Function | Binary cross-entropy |
| Input Length | 100 |
| Learning rate | 0.002 |
| Epoch | 10 |

Table 3. Comparison of proposed HCL-Bi-LSTM model.

| Model | Single-layer Bi-LSTM (Hameed and Garcia-Zapirain, 2020) | | Two-layer Bi-LSTM | | Two-layer HCT Bi-LSTM | |
|---|---|---|---|---|---|---|
| Dataset | T | V | T | V | T | V |
| Amazon | 0.83 | 0.51 | 0.91 | 0.70 | 0.95 | 0.76 |
| Yelp | 0.84 | 0.70 | 0.85 | 0.72 | 0.86 | 0.75 |
| IMDB | 0.71 | 0.81 | 0.90 | 0.81 | 0.95 | 0.82 |

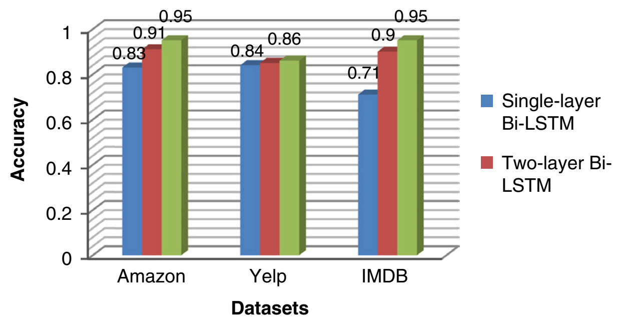

Figure 6: Accuracy comparison of the proposed model for three different datasets.

## Results

The proposed approach's evaluation result is compared with the single-layer Bi-LSTM model (Hameed and Garcia-Zapirain, 2020) and the two-layer Bi-LSTM model. In this work, we have presented two models. The first is a two-layer Bi-LSTM model, and the second is the HCT Bi-LSTM models. Table 3 represents the comparison of the accuracy of these models for three different datasets.

For better representation, Figure 6 shows the comparative graph.

From Table 3, it is clear that the proposed model gives better accuracy for all three datasets. It achieves maximum accuracy of up to 95% and an average accuracy of 92%.

Figures 7-9 represent the accuracy and loss graph of the single-layer Bi-LSTM model for all three considered datasets concerning the number of epochs.

Figures 10-12 represents the accuracy and loss graph of our proposed HCT two-layer Bi-LSTM model for all three considered datasets concerning the number of epochs.

The proposed optimized model represents better accuracy for all three datasets.

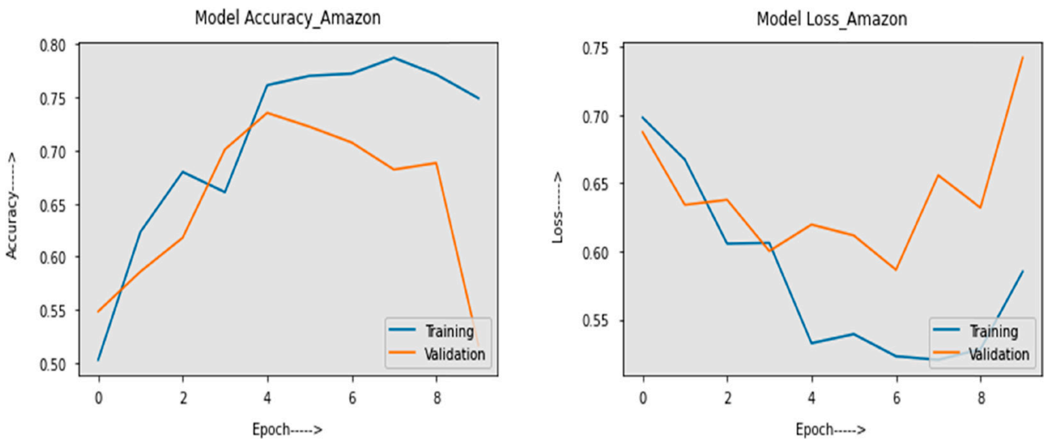

Figure 7: Performance of single-layer Bi-LSTM on Amazon dataset.

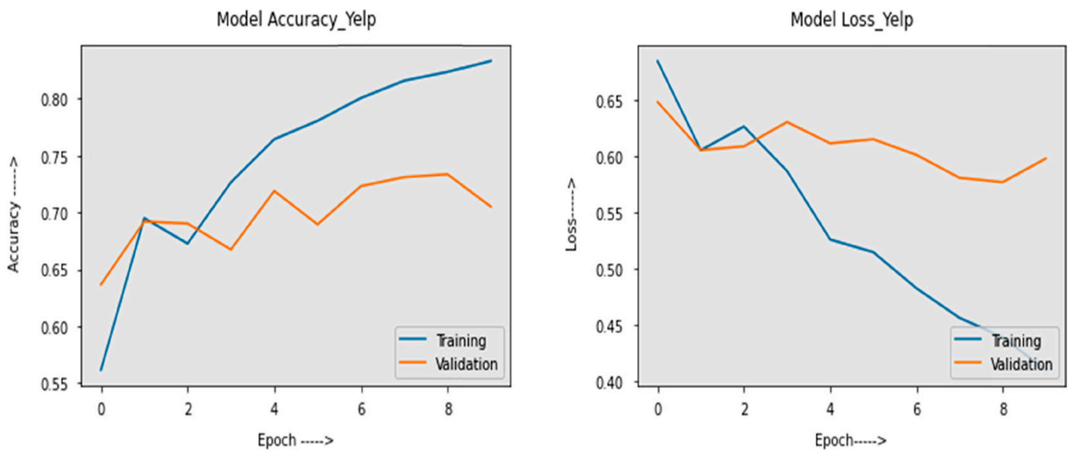

Figure 8: Performance of single-layer Bi-LSTM on Yelp dataset.

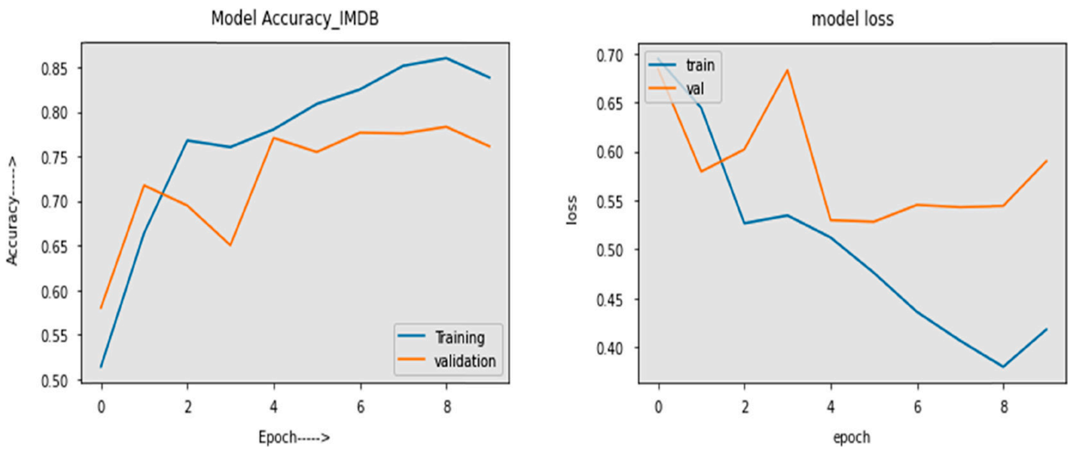

Figure 9: Performance of single-layer Bi-LSTM on IMDB dataset.

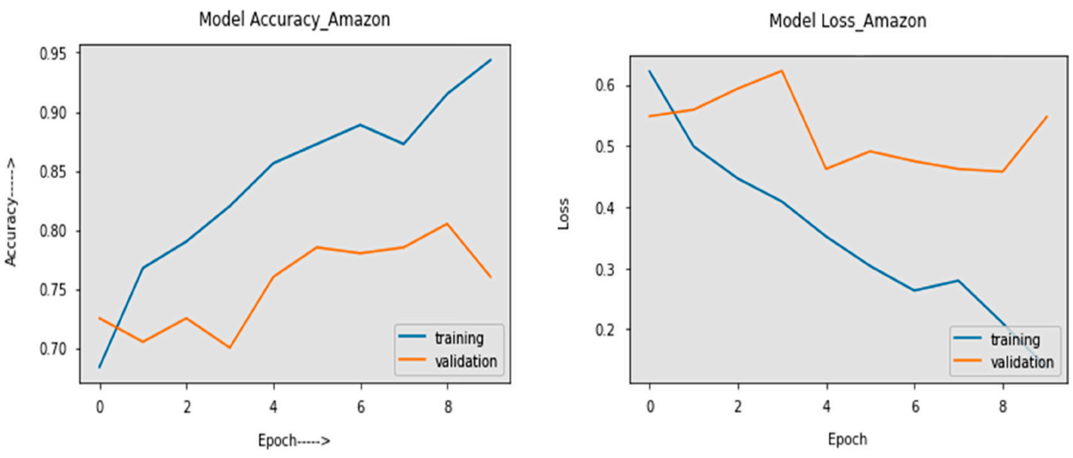

Figure 10: Performance of HCT two-layer Bi-LSTM on Amazon dataset.

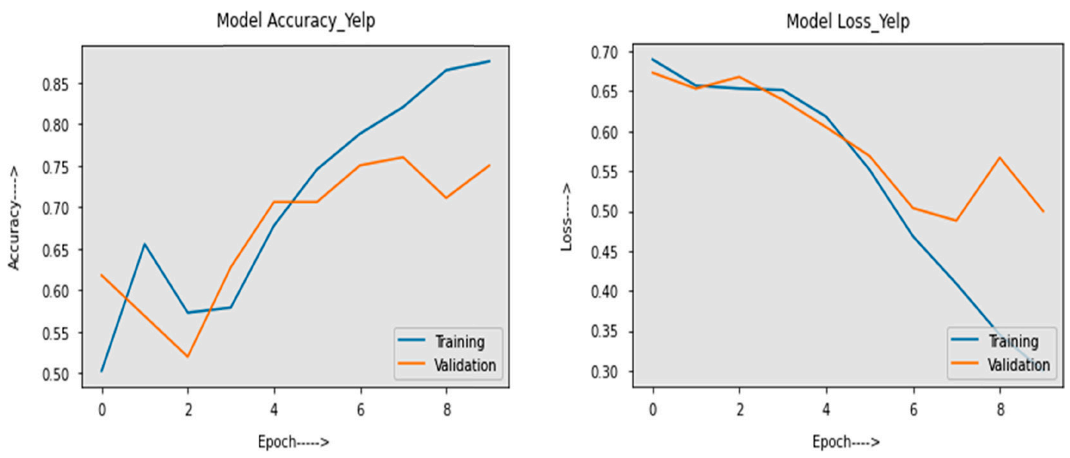

Figure 11: Performance of HCT two-layer Bi-LSTM on Yelp dataset.

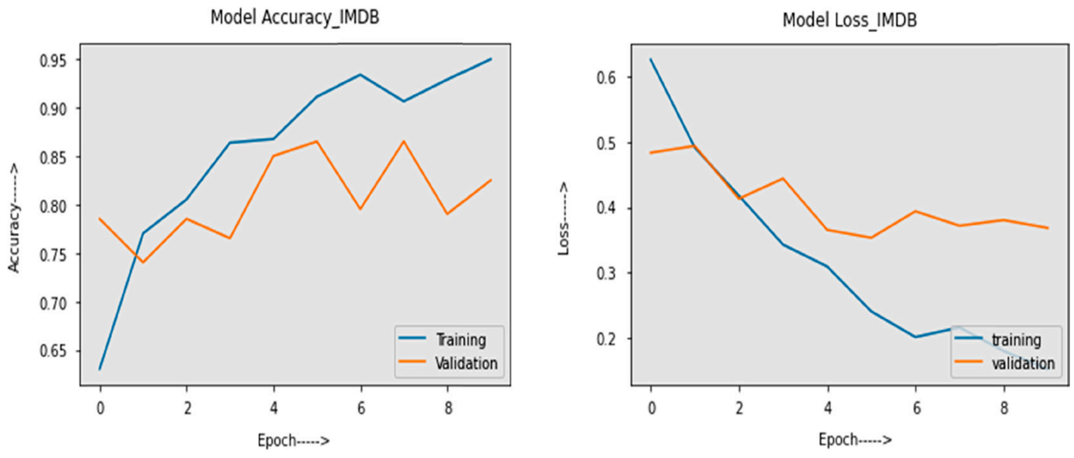

Figure 12: Performance of HCT two-layer Bi-LSTM on IMDB dataset.

## Conclusion

In this paper, a topic modeling based multi-layer Bi-LSTM model with pre-trained embedding is used for aspect-based sentiment analysis. The proposed model is efficient and more accurate. Only two layers of Bi-LSTM are stacked to keep model complexity low. Further performance is improved through hyperparameters tuning using the HCT algorithm. The proposed model is compared with the single-layer Bi-LSTM and two-layer Bi-LSTM models. It has shown better accuracy and efficiency when evaluated on three different datasets. The proposed model is giving 95, 95, and 86% accuracy for movie, mobile, and hotel domains. In the future, the ensemble approach can be used for improving performance. Integrated CNN-LSTM can also be try. In terms of pre-trained embeddings, domain-based embeddings can also be used for more efficiency.

## Literature Cited

Agarap, A. F. and Grafilon, P. 2018. Statistical analysis on E-commerce reviews, with sentiment classification using bidirectional recurrent neural network (RNN). arXiv preprint arXiv:1805.03687.

Blair, S. J., Bi, Y. and Mulvenna, M. D. 2020. Aggregated topic models for increasing social media topic coherence. *Applied Intelligence* 50(1): 138–156.

Blei, D. M., Andrew, Y. N. and Michael, I. J. 2003. Latent Dirichlet allocation. *The Journal of Machine Learning Research* 3: 993–1022.

Chakraborty, K., Bhatia, S., Bhattacharyya, S., Platos, J., Bag, R. and Hassanien, A. E. 2020. Sentiment analysis of COVID-19 tweets by deep learning classifiers–a study to show how popularity is affecting accuracy in social media. *Applied Soft Computing* 28: 106754.

Hameed, Z. and Garcia-Zapirain, B. 2020. Sentiment classification using a single-layered BiLSTM model. *IEEE Access* 8: 73992–74001.

Huang, R., Taubenböck, H., Mou, L. and Zhu, X. X. 2018b. Classification of settlement types from Tweets using LDA and LSTM. *IGARSS 2018- IEEE International Geoscience and Remote Sensing Symposium*, IEEE, pp. 6408–6411.

Huang, Y., Jiang, Y., Hasan, T., Jiang, Q. and Li, C. 2018a. A topic BiLSTM model for sentiment classification. *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*, pp. 143–147.

Jansson, P. and Liu, S. 2017. “Topic modelling enriched LSTM models for the detection of novel and emerging named entities from social media. *2017 IEEE International Conference on Big Data* (*Big Data*), IEEE, pp. 4329–4336.

Jelodar, H., Wang, Y., Orji, R. and Huang, S. 2020. Deep sentiment classification and topic discovery on novel coronavirus or covid-19 online discussions: NLP using lstm recurrent neural network approach. *IEEE Journal of Biomedical and Health Informatics* 24(10): 2733–2742.

Jim, S., Ali, G. G., Rahman, M., Esawi, E. and Samuel, Y. 2020. Covid-19 public sentiment insights and machine learning for tweets classification. *Information* 11(6): 314.

Jo, Y., Lee, L. and Palaskar, S. 2017. Combining LSTM and latent topic modeling for mortality prediction. arXiv preprint arXiv:1709.02842.

Lai, S., Xu, L., Liu, K. and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Luo, L.-X. 2019. Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal and Ubiquitous Computing* 23(3): 405–412.

Man, H., Lauren, E., Hon, E. S., Birmingham, W. C., Xu, J., Su, S., Hon, S. D., Park, J., Dang, P. and Lipsky, M. S. 2020. Social network analysis of COVID-19 Sentiments: application of artificial intelligence. *Journal of Medical Internet Research* 22(8): e22590.

Minaee, S., Azimi, E. and Abdolrashidi, A. A. 2019. Deep-sentiment: sentiment analysis using ensemble of CNN and Bi-LSTM models. arXiv preprint arXiv:1904:04206.

Nemes, L. and Kiss, A. 2021. Social media sentiment analysis based on COVID-19”. *Journal of Information and Telecommunication*, 5(1): 1–15, available at: https://doi.org/10.1080/24751839.2020.1790793.

Priyantina, R. and Sarno, R. 2019. Sentiment analysis of hotel reviews using latent Dirichlet allocation, semantic similarity, and LSTM. *International Journal of Intelligent Engineering and Systems* 12(4): 142–155.

Rhanoui, M., Mikram, M., Yousfi, S. and Barzali, S. 2019a. A CNN-BiLSTM model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction* 1(3): 832–847.

Rhanoui, M., Mounia, M., Yousfi, S. and Barzali, S. 2019b. A CNN-BiLSTM model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction* 1(3): 832–847.

Shafqat, W. and Byun, Y.-C. 2019. Topic predictions and optimized recommendation mechanism based on integrated topic modeling and deep neural networks in crowdfunding platforms. *Applied Sciences* 9(24): 5496.

Wang, J.-H., Liu, T.-W., Luo, X. and Wang, L. 2018. “An LSTM approach to short text sentiment classification with word embeddings. Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), pp. 214–223.

Xu, H., Liu, B., Shu, L. and Yu, P. S. 2018. “Double embeddings and CNN-based sequence labeling for aspect extraction. arXiv preprint arXiv :1805.04601.

Xue, J., Chen, J., Chen, C., Zheng, C., Li, S. and Zhu, T. 2020. Public discourse and sentiment during the COVID 19 pandemic: using latent Dirichlet allocation for topic modeling on Twitter. *PLoS ONE* 15(9): e0239441.

Yadav, A., Jha, C. K. and Sharan, A. 2020. Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science* 167: 2091–2100.

Zhang, D., Luo, T. and Wang, D. 2016. “Learning from LDA using deep neural networks”, In *Natural Language Understanding and Intelligent Applications* Springer, Cham, pp. 657–664.

Zhang, W., Li, Y. and Wang, S. 2019. Learning document representation via topic-enhanced LSTM model. *Knowledge-Based Systems* 174: 194–204.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H. and Xu, B. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv:1611.06639.