The Comparison of Transformer and Improved BiLSTM Model for Document-level Sentiment Analysis

Tao He

School of Public Administration, Nanjing University of Finance & Economics, Nanjing, China

taohe@ldy.edu.rs

Abstract. Sentiment Analysis involves techniques to assess emotions and opinions in digital content like text, audio, and video. It is widely applied to analyze and interpret large volumes of text, such as customer feedback or online reviews, making it to make use of such a valuable asset in various fields such as marketing, which is crucial for understanding customer preferences, enhancing recommendation systems, and building business decision systems. The most state-of-art sentiment analysis method is machine learning methods. This study conducts the evaluation of multiple deep-learning models in performing document-level sentiment analysis tasks. The experiments carried out on Amazon book review data reveal that the proposed 2-layer BiLSTM model, which incorporates several improving techniques such as fine-tuning, layer normalization, attention, and pooling mechanisms, consistently outperforms other tested models including BiGRU-BiLSTM and BERT, achieving a recall rate of 0.71 with an F1 score of 0.7112, while BERT model achieving a 0.685 and 0.6895 separately.

INTRODUCTION

Sentiment Analysis is a wide field encompassing techniques and methods. Its main goal is to interpret emotions, feelings, and opinions that users express in their digital records, be it in text, audio, or video formats. Among all possible media of sentiment analysis, the main application area is still the analysis of natural language. Sentiment analysis is currently applied in many fields, such as marketing, political events, health, education, smart cities, etc [1]. For instance, sentiment analysis on comments can reveal users' emotional responses to products or services in the markets. This is crucial for understanding customer preferences, enhancing recommendation systems to offer more relevant suggestions, and aiding business decisions.

Traditional text representation systems include TF-IDF, BoW and Lexicons [2,3]. The main objective of these techniques is to pull out valuable features from the text that can be utilized in machine learning algorithms. These methods represent text as discrete elements, such as words or phrases, without accounting for their order. They disregard the text's syntactic structure and the sequence of words, focusing instead on the presence or frequency of words within the text. Typically, sentiment analysis employs machine learning methods using word embedding techniques. Recurrent neural networks (RNNs) can address certain defects but struggle with gradient issues [4]. The scaled dot-product attention is recognized as a highly effective method for incorporating attention mechanisms in the Transformer model [5]. Word embeddings are dense, real-valued vectors generated using neural language models, taking into account various lexical relationships [6-8]. These models utilize word embeddings for text representation, effectively addressing the issues of high dimensionality and data sparsity often found in traditional methods. Word-pre-trained embedding models like Word2Vec, FastText, and GloVe have demonstrated promising results in the field of sentiment analysis [9,10].

With emotional polarity, subjectivity issues may arise in the definition of emotional labels due to differences in subjectivity, context, cultural variations, the complexity of emotions, and implicit meanings [11]. Also, beyond polarity sentiment analysis, a significant subset of research explores multi-level sentiment classification tasks. These tasks are not only to judge text as positive, negative, or neutral but also to examine emotions at different levels. This might involve classifying fine-grained emotions, such as categorizing feelings into emotions like joy, anger, sorrow, and happiness. In the multilevel sentiment classification task, the challenge lies in the limited coherence, as well as the contextual and semantic information [6].

Sentiment analysis's applications on NLP tasks have been classified by various levels [11]. Recent research has seen a rise in the popularity of aspect-level studies. Despite this, document-level research shouldn't be dismissed as it holds potential and value. Specifically, in document-level sentiment analysis, two crucial areas are cross-domain and cross-language sentiment analysis [12].

Sentiment analysis typically employs hybrid models based on BiLSTM and BiGRU [13]. Comparisons involving transformers are limited to the aspect level [14], with minimal research conducted on document-level reviews. While the Transformer is compared with models grounded in Bayesian, CNN, and GRU [15], a comparison with LSTM-based models is absent. There's also a scarcity of research into implementing a double-layer bidirectional LSTM based on bert-base. In this study, the performances of various cutting-edge models and a suggested 2-layer BiLSTM model are compared in document-level sentiment analysis tasks, including hybrid BiGRU+BiLSTM methods and transformers. To further extract key features of the sequence, this improved model introduces attention mechanism, pooling mechanism and layer normalization. The models are trained and their performance is compared under the same evaluation metrics, including recall rate and F1 score. Model fine-tuning is done through cross-validation and validation sets.

The following are this study's primary contributions:

- Utilizing sentiment labels based on user-subjective ratings can help avoid subjectivity issues in sentiment label processing.
- Introducing an enhanced 2-layer BiLSTM model specifically fine-tuned for sentiment analysis of reviews.
- Making a comparison between the standard BiLSTM and the improved BiLSTM in terms of accuracy. And the effectiveness of this improvement is verified.
- Making a comparison between performances of the 2-layer BiLSTM and BERT.

METHODOLOGY

Fig. 1 displays the overall model using the BERT embedding layer(last hidden layer) to obtain context encoding. The model parameters are set trainable and adjusted through fine-tuning. After each Bidirectional LSTM layer, add a layer normalization layer and an attention layer to introduce the self-attention mechanism. After the concatenation of two BiLSTM sets, add a maximum pooling layer to reduce the dimensions of output.

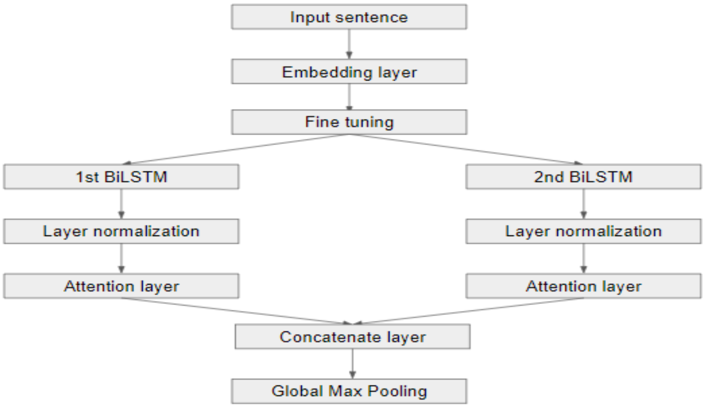

FIGURE 1. BiLSTM Model Structure (Photo/Picture credit :Original)

Text Embedding

Text embedding is the technique of converting text to a vector representation in deep learning methods. The main purpose is to map words, phrases, or entire text to vectors in a high-dimensional space. This vector representation captures lexical, syntactic, and semantic information, allowing computers to better understand and process textual data. For example, two words will be similar if their vector representations are close to each other. An Embedding

Matrix is simply a collection of embedding values for all words in the vocabulary. To get the text embedding based on a large corpus, a pre-trained word vector model (such as Word2Vec, GloVe, or FastText) is often used. To consider the context of words in context, context embedding models (e.g. BERT, GPT, etc.) are also frequently considered. BERT, as a transformer model, features key components such as the Self-Attention Mechanism, Multi-Head Attention, Positional Encoding, Layer Normalization, and Residual Connections.BERT conducted two pre-training tasks on large-scale corpus. Masked Language Model (MLM) predicts parts of the word that is obscured to learn the relationship in the context. Next Sentence Prediction (NSP) asks the model to determine whether two sentences are semantically adjacent.

Bidirectional LSTM model

Among the most advanced techniques, LSTM is the most widely used. RNN structure serves as the foundation for Long Short Time Memory (LSTM). Sequence data (text or time series) is processed using RNNs, because it introduces loop connections. LSTM can solve the problem of gradient disappearance of RNNs. How much of the prior data in the cell state is forgotten is controlled by the forget gate, using the Sigmoid activation function to limit its output between 0 and 1. The input gate regulates how much fresh data enters the cell state. Candidate Cell Value proposes potential new information to update the cell state, taking advantage of the hyperbolic tangent activation function, which yields values between -1 and 1. Cell State Update decides how much of the previous information to retain and new information to accept, with updating the cell state based on the candidate cell value. The output gate controls the amount of data sent from the cell state to the current hidden state by using the Sigmoid activation function to determine which features of the cell state will be output. The current hidden state is created by Hidden State Update using the output gate and cell state. The following are the mathematical formulas:

Forget Gate:

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t] +b_f) \tag{1}$$

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

Candidate Cell Value:

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Cell State Update:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

Hidden State Update:

$$h_t = o_t \cdot tanh(C_t) \tag{6}$$

Where:

W_f, W_i, W_C, W_o are weight matrices.

$[h_{t-1}, x_t]$is the concatenation of the vectors of the previous hidden state.

b_f, b_i, b_C, b_o are bias of each step.

Fig. 2 shows the anatomy of the two-layer BiLSTM model consisting of the input layer, BiLSTM sets, concatenation layer, and out layer. BiLSTM consists of two separate LSTM structures, one responsible for the backward processing of input data from the beginning position of the sequence and the other responsible for the forward processing of input data from the end position of the sequence. This way, each moment's concealed state considers contextual information both before and after it in the sequence, in addition to the input at that particular moment. By considering both forward and reverse context information, BiLSTM can better understand the overall context of the input sequence, helping to capture long-distance dependencies in the sequence. Due to the bidirectional structure, BiLSTM usually has twice the number of parameters as normal LSTM. This adds complexity to the model, allowing it to adapt more flexibly to different types of data. BiLSTM is generally more computationally expensive. On the one hand, this increases the time for training and reasoning and also requires more computing resources.

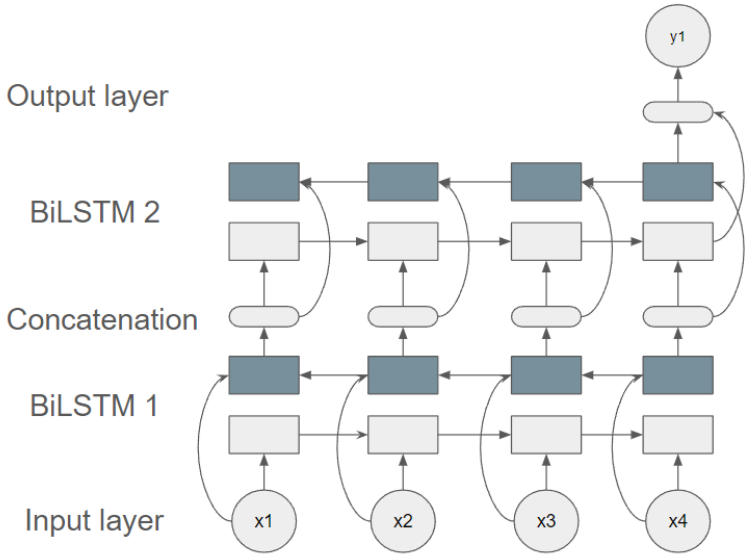

FIGURE 2. Two-layer BiLSTM Anatomy (Photo/Picture credit :Original)

The following techniques are used to improve the efficiency of the basic 2-layer BiLSTM model:

- Fine-tuning: This study fine-tunes the BiLSTM model, tweaking it on specific tasks to make it more suitable for the application scenarios. The model's parameters are configured to be trainable. and adjusted to improve performance and ensure that the model can better adapt to the specific requirements of the task.
- Attention mechanisms: The Self-Attention Mechanism is a mechanism for processing sequence data, originally used in the Transformer model. It allows the model to assign a different weight of attention to each element in the sequence, rather than just depending on their position. In the Self-Attention Mechanism, a weight is calculated for each element in the input sequence. These weights are calculated using a learnable attention function. The attention weight, also known as the attention score, can be computed using the formula below:

$$\alpha_{i,j} = softmax\left(\frac{Q_i \cdot K_j}{\sqrt{d_k}}\right) \tag{7}$$

  Where:
  Q_i is the 1uery vector for the i-th element.
  K_j is the key vector for the j-th element.
  d_k is the dimensionality of the key vectors.
- Max Pooling: To reduce the BiLSTM output's dimensionality, max pooling reduces the quantity of parameters and thus improves computational efficiency.
- Layer normalization: Layer normalization computes normalization statistics within each sample individually. It normalizes the input of each layer in the model individually, and lessens internal covariate shifts.

EXPERIMENTAL RESULT

The experiment was run on Google Colab which offers a free GPU-accelerated environment for efficient model training. This study used TensorFlow and PyTorch frameworks, and pre-trained models like BERT from the Transformers library. The data set was the Amazon book review data from Kaggle, relabelled to classify emotions. To balance data, each sentiment category contained 1000 samples, with an 80/20 split for training and validation. The first BiLSTM layer was configured with 64 hidden units and the second with 50. The model was trained over 10 iterations on the entire training data set, using a batch size of 16. Throughout the experiment, the loss value and performance index were monitored. The learning rate was fixed at 1e-5, and the loss function was categorical cross entropy.

Evaluation

To evaluate the baseline and comparative models, the assessment criteria are recall rate and F1 score. The recall rate measures how many of the true positive categories are successfully predicted by the model to be positive categories. The F1 score is a harmonic average of accuracy and recall.

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{10}$$

Where:

TP is the number of instances correctly predicted as positive.

FN is the number of instances incorrectly predicted as negative.

N is the number of total predictions.

FP is the number of instances incorrectly predicted as positive.

Result Analysis

Table 1 exhibits the results comparing the performances between the improved model and other baseline models. The 2-layer LSTM enhanced accuracy by approximately 2 percentage points when compared to the hybrid BiGRU and BiLSTM. BiLSTM has a better ability to efficiently capture long-distance dependencies in the document-level text. BiLSTM has more parameters and can be more flexible to adapt to different types of data. In sentiment analysis tasks, understanding the text may require more parameters to capture semantic information.

Further, after setting the parameters trainable and fine-tuning the embedding model, thus improving the generalization ability, the accuracy of the model was improved by 20 percentage points. By setting the parameters trainable and adjusting the model parameters to a state more appropriate to the task, the model can be better adapted to a specific sentiment analysis task. This can lead to improved performance on the validation set, which improves accuracy. The results also reveal that models using layer normalization are more accurate than those using batch normalization. Layer normalization helps mitigate gradient disappearance and explosion problems consequently improving the model's training stability and performance better, achieving 0.7017 of recall and 0.7063 of F1 Score.

In addition, the attention layer was applied to further improve the performance by about 1 percentage point. The self-attention mechanism for each BiLSTM layer better focuses on the more important input parts differently.

TABLE 1. Comparison results

Model	Recall	F1 Score
BERT transformer	0.6850	0.6895
Hybrid model:BiGRU + BiLSTM	0.4783	0.4760
2-layer BiLSTM	0.4900	0.4870
2-layer BiLSTM with fine-tuning and batch normalization	0.6633	0.6602
2-layer BiLSTM with fine-tuning, attention and pooling	0.6683	0.6750
2-layer BiLSTM with fine-tuning, layer normalization and pooling	0.7017	0.7063
2-layer BiLSTM with fine-tuning, layer normalization, attention and pooling	0.7100	0.7112

Fig. 3 shows the confusion matrix suggesting the model's strong accuracy in positive and negative categories, but more errors in the neutral category. These errors could stem from similarities between the neutral and the other categories, insufficiently captured features of the neutral category, or mislabeling in the neutral category.

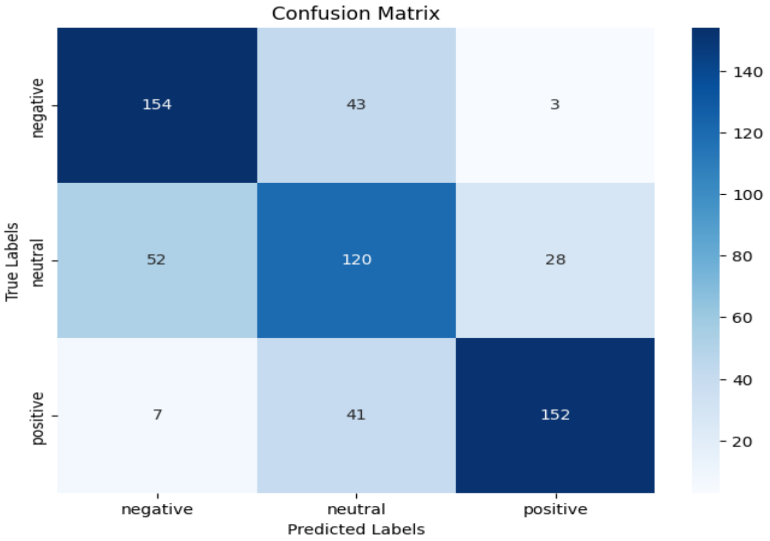

FIGURE 3. Confusion Matrix for Two-Layer BiLSTM (Photo/Picture credit :Original)

CONCLUSION

This study compares the performance of the improved 2-layer BiLSTM model multiple benchmark models and the BERT model for document-level sentiment analysis. The experiments on Amazon book review data demonstrated that the proposed model, improved by fine-tuning, layer normalization, attention, and pooling mechanisms, achieved better performance compared to other models. The 2-layer BiLSTM model had a recall rate of 0.71 with an F1 score of 0.7112, while BERT model achieved 0.685 and 0.6895 separately. However, challenges remain in the neutral category, requiring further research and improvement to enhance the model's performance. In the future, possible improvement schemes will be researched, including model structure adjustment, data preprocessing, feature engineering, etc. Further research includes experiments with the average pooling method and other word embedding techniques. A thorough comparison is planned on the suitability, pros, and cons of different models across various scenarios and languages and the scalability and stability of the improved method including delving further into the comprehensive research on the intricate task of multilevel sentiment classification.

REFERENCES

1. M.A. Alonso, D. Vilares, C. Gómez-Rodríguez, and J. Vilares, *Electronics* 1348 (2021).
2. K. Jindal and R. Aron, Materials Today: Proceedings (2021).
3. H. Zhao, Z. Liu, X. Yao, and Q. Yang, Information Processing &amp; *Management* 102656 (2021).
4. K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, and A. Hussain, Entropy 596 (2021).
5. M.M. Agüero-Torales, J.I. Abreu Salas, and A.G. López-Herrera, *Applied Soft Computing* 107373 (2021).
6. Md.S. Islam and N.A. Ghani, in Lecture Notes in Electrical Engineering,*Recent Trends in Mechatronics Towards Industry* 4.0 (2022), pp. 403–414.
7. A. Onan, Computer and Information Sciences, *Journal of King Saud University* 2098 (2022).
8. S. S, P. Sunagar, S. Rajarajeswari, and A. Kanavalli, *International Journal of Advanced Computer Science and Applications* (2022).
9. K. Zhang, L. Feng, and X. Yu, in Web and Big Data,*Lecture Notes in Computer Science* (2023), pp. 444–458.
10. M. Pota, M. Ventura, H. Fujita, and M. Esposito, *Expert Systems with Applications* 115119 (2021).
11. M. Wankhade, A.C.S. Rao, and C. Kulkarni, *Artificial Intelligence Review* 5731 (2022).
12. G. D. Aniello, M. Gaeta, and I. La Rocca, *Artificial Intelligence Review* 5543 (2022).
13. L. Yang, Y. Li, J. Wang, and R.S. Sherratt, IEEE Access 23522 (2020).
14. Z. Wu, C. Ying, X. Dai, S. Huang, and J. Chen, *in Natural Language Processing and Chinese Computing*,Lecture Notes in Computer Science (2020), pp. 546–557.
15. B. Zhang and W. Zhou, *Neural Processing Letters* (2022).