Text classification sentimental analysis using Neural Networks on Movie Reviews

1st Abid Hossain
Dept. of CSE
BRAC University
Dhaka, Bangladesh
abid.hossain@g.bracu.ac.bd
ID: 20301115

2nd Tausif Ahanaf
Dept. of CSE
BRAC University
Dhaka, Bangladesh
md.tausif.ahanaf@g.bracu.ac.bd
ID: 23341090

3rd Tasfia Jahan
Dept. of CSE
BRAC University
Dhaka, Bangladesh
tasfia.jahan.nahin@g.bracu.ac.bd
ID: 18201129

4th Sania Azhmee Bhuiyan
Dept. of CSE
BRAC University
Dhaka, Bangladesh
sania.azhmee.bhuiyan@g.bracu.ac.bd

5th Saidul Arefin Rafe
Dept. of CSE
BRAC University
Dhaka, Bangladesh
saidul.arefin.rafe@g.bracu.ac.bd
ID: 20101558

5th Shamaun Shamim Mukit
Dept. of CSE
BRAC University
Dhaka, Bangladesh
shamaun.shamim.mukit@g.bracu.ac.bd
ID: 20101558

6th Annajiat Alim Rasel
Dept. of CSE
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—This research undertakes a comparative analysis of sentiment analysis methods within text classification, contrasting the effectiveness of deep learning approaches - Recurrent Neural Networks (RNN), extension of LSTM approach such as Bi-Directional Long Short Term Memory Networks (BiLSTM), Recurrent Neural Networks and last but not least, Multi-Layer Perceptron Network (MLP). Dataset used was IMDB dataset of movie reviews. This research underscores the potential of synergizing this models with optimizers and GloVe word embedding techniques to bolster sentiment-driven text classification accuracy [1]. The study revealed that the BiLSTM model with the incorporation of adam optimizer had the highest performance metric. This finding of the best model and amalgamation of these fine-tuning of baseline neural network models presents a promising avenue for advancing the precision and efficacy of text classification in natural language processing domains.**

***Index Terms*—Sentimental Analysis, Recurrent Neural Networks, Back propagation, Max pooling, Support Vector Machines, Logistic Regression, Multinomial Naive Bayes**

I. Introduction

Sentiment analysis, also known as opinion mining, holds a significant role when it comes to understanding people's feelings in text. Imagine reading movie reviews online – some are full of praise, some are disappointed, and others are just neutral. Sentiment analysis helps us sort through these reviews and figure out whether they're positive, negative, or somewhere in between. This is especially important for movie reviews on platforms like IMDb, where people share their thoughts on films. Analyzing sentiments in these reviews can reveal how much people enjoyed a movie, whether it left them disappointed, or if their opinions are more neutral. The models used for our study are Bi-Directional Long Short Term Memory Networks (BiLSTM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Multi-Layer Perceptron Networks (MLP). Each of these models has its own strengths in understanding text. BiLSTMs are like memory experts, CNNs can find patterns in words, RNNs follow the order of words, and MLPs connect different ideas together. In this research, we're specifically focusing on using these models to analyze movie reviews, especially those from the IMDb dataset. As a follow-up and a new distinct feature, GloVE word embeddings will be implemented during the training phase of the networks because GloVe is more suitable for rendering capacity with the 50,000 rows dataset [2]. IMDb is a popular website where people rate and review movies. The dataset prompts us to advocate for a neural network architecture imbued with neurons capable of intricate stochastic decisions concerning temporal events across varying time scales. An example would be the application of binary decisions, where a 0 or 1 value is given to signify the conclusion of an inception of word or phrase boundaries within textual content [3]. This binary output, generating sparse representations, is strategically harnessed as a regularization technique. Analyzing sentiments in these reviews can tell us a lot about how people feel about different films. We want to make these models even better at understanding movie reviews. Think of it as teaching them to be like super reviewers – they can quickly decide if a review is positive, negative, or neutral, just like a human would.

II. RELATED WORKS

In their work, Liu et al. (2012) [4] proposed a sentiment analysis model that integrated user reviews to improve sentimental analysis on financial data. The Rating Graph Neural Network (RGNN) was designed to leverage the semantic information present in user reviews. By constructing rating graphs for users and items, the model captured relationships between words within reviews. RGNN employed a type-aware graph attention mechanism and custom graph clustering operators to extract hierarchical semantic representations.[4] Through the integration of the Factoring Machine (FM) class, RGNN predicted user ratings based on learned semantic features. Extensive experimentation on real-world datasets demonstrated the superiority of RGNN over other contemporary methods in terms of mean squared error (MSE)[5].

Pang et al. (2012)[6] made a significant contribution to sentiment analysis by introducing a domain adaptation approach that effectively tackled the challenge of sentiment classification in domains with limited labeled data. They recognized that sentiment analysis models trained on one domain might not generalize well to another due to domain-specific language variations[5]. To address this, Pang et al. proposed a novel approach that incorporated both labeled data from a source domain and unlabeled data from a target domain. They introduced a joint distribution model based on Structural Correspondence Learning (SCL) that aligned sentiment spaces between the source and target domains. By leveraging the SCL model, they effectively adapted a sentiment classification model trained on the source domain to perform well on the target domain[7].

Tang et al. (2012)[8] introduced a seminal contribution to sentiment analysis by proposing a novel framework that synergized topic modeling and sentiment classification. Their approach involved utilizing Latent Dirichlet Allocation (LDA), a popular topic modeling technique, to enhance sentiment analysis. They recognized that sentiments can vary significantly within different topics, leading to sentiment ambiguity in traditional methods. The integration of topics into sentiment analysis helped disambiguate sentiments within different contexts, significantly improving classification accuracy. The authors' experiments on benchmark datasets demonstrated the effectiveness of their approach[9]. The results showcased notable improvements over traditional sentiment analysis methods, particularly in scenarios where sentiment was intricately intertwined with topic-specific nuances. By incorporating topic information into sentiment analysis, Tang et al. not only enhanced the accuracy of sentiment classification but also enriched the understanding of sentiments in diverse textual domains.

Zhou et al. (2012)[10] presented a significant contribution to sentiment analysis in their paper ”Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.” They introduced a model that aimed to improve sentiment classification accuracy by learning sentiment-specific word embeddings. By utilizing a Word2Vec-based approach, their method generated word embeddings tailored to sentiment orientation, capturing the inherent sentiment of words. This approach effectively addressed the challenge of sentiment-related polysemy, where words have different meanings in different sentiment contexts. The sentiment-specific word embeddings enhanced sentiment analysis by providing more contextually relevant representations for sentiment-bearing words, thereby improving classification accuracy[11].

Devlin et al. (2012)[3] made a pivotal contribution to sentiment analysis with their paper ”Sentiment Analysis with LSTM and Word2Vec.” They introduced a model that combined Long Short-Term Memory (LSTM) networks and Word2Vec embeddings to improve sentiment classification accuracy. By leveraging the temporal dependencies in text sequences through LSTMs and the semantic relationships between words using Word2Vec, their approach achieved enhanced sentiment understanding. LSTM networks effectively captured the context and sequential patterns crucial for sentiment analysis in text data. The integration of Word2Vec embeddings provided a richer representation of words, allowing the model to capture intricate linguistic nuances[12]. Devlin et al.'s work showcased the power of combining deep learning techniques and word embeddings to tackle the challenges of sentiment analysis, demonstrating the potential for more accurate sentiment classification in various domains.

Thelwall et al. (2012)[13] presented a valuable contribution to sentiment analysis with their paper ”Sentiment Strength Detection in Short Informal Text.” They introduced a model that focused on detecting the strength of sentiment expressions in short texts, addressing the challenge of nuanced sentiment understanding. Their approach employed a lexicon-based method combined with linguistic features to determine the strength of sentiments expressed. By integrating machine learning techniques, specifically Support Vector Machines (SVMs), their model effectively classified sentiment strength in diverse contexts. The inclusion of linguistic features like intensifiers and negations enhanced the model's ability to capture subtle variations in sentiment intensity. Thelwall et al.[1]'s work provided a nuanced perspective on sentiment analysis by recognizing that sentiment strength holds crucial information beyond mere positive or negative sentiment labels.

Amini et al. (2012)[14] made a notable contribution to sentiment analysis with their paper ”Sentic LDA: Improving on LDA with Semantic Similarity for Aspect-Based Sentiment Analysis.” They proposed an enhanced model called Sentic LDA that incorporated semantic similarity for more effective aspect-based sentiment analysis. By combining

Latent Dirichlet Allocation (LDA) with semantic similarity measures, their model improved the extraction of aspect-specific sentiments from text data. Sentic LDA incorporated SenticNet, a sentiment lexicon, to measure the semantic relatedness between words and aspects, enriching the topic modeling process. Amini et al.[2]'s work demonstrated the effectiveness of combining topic modeling with semantic similarity, yielding more accurate and meaningful aspect-based sentiment analysis results. The integration of SenticNet and LDA showcased the potential of leveraging external resources to enhance sentiment analysis methodologies.

Kim et al. (2020)[15] made a notable contribution to sentiment analysis in their paper "Hierarchical Transformer with Sentence-level Attention for Aspect-based Sentiment Analysis." They introduced a novel model that effectively addressed aspect-based sentiment analysis by combining hierarchical transformers with sentence-level attention. Their approach resulted in enhanced sentiment classification accuracy, particularly in complex texts where multiple aspects are discussed. Kim et al.[8]'s work demonstrated that combining hierarchical structures and attention mechanisms can significantly improve aspect-based sentiment analysis, providing a more nuanced understanding of sentiment expressions towards different aspects within a text.

III. Methodology

A. Data Collection

The IMDb dataset utilized is extracted from the respected 49th Annual Meeting of the Association for Computational Linguistics [14], ensuring its reliability and credibility. The dataset, encompassing a variety of textual sources including social media excerpts, product reviews, and news articles, undergoes meticulous preprocessing. It contains a large collection of movie reviews accompanied by binary sentiment labels indicating positive or negative sentiments. The dataset encompasses a diverse range of reviews, capturing various genres, tones, and sentiments expressed by users. It is typically split into predefined subsets, with a substantial portion reserved for training to facilitate the learning process. Additionally, validation and test sets are delineated to assess the model's performance on unseen data and to prevent overfitting. The dataset's balanced distribution of positive and negative sentiments ensures that models' performance is not skewed toward any particular class.

B. Datapreprocessing and Collection

Data preprocessing is an essential initial step to ensure the quality of input data for subsequent analysis. To facilitate this, instances containing null or NaN values were removed using the dropna function from the NumPy library. Techniques like stemming, lemmatization, and tokenization are employed to enhance feature extraction. This refined dataset is then partitioned into distinct training and testing subsets. Moreover, the Natural Language Toolkit (NLTK) library was employed to enhance the quality of textual data. NLTK's built-in stopword list was utilized to create a list of stopwords, which were subsequently removed from the text data. Additionally, a custom function named getsimplepos was developed to ascertain the part of speech of individual words. Textual data was tokenized into words, and the WordNet tokenizer was employed to further cleanse the text by removing punctuation and special characters. The utility of the following NLTK functionalities significantly aided in this preprocessing phase. The dataset exhibited a certain degree of polarization, with unequal samples among the sentiment classes. To address this issue and enhance the model's performance, oversampling was implemented. Oversampling involves replicating instances from the minority class to balance the distribution of sentiment classes. This strategy mitigates the bias introduced by the imbalanced data distribution, thereby improving the model's ability to generalize effectively across all sentiment classes.

C. Word embedding layer preparation

The text reviews from the dataset were first tokenized from text words to numeric vector representation. The process was carried out using the tokenizer function of the keras preprocessing library. Then the method "fitontext" was used to train the tokenizer and subsequently for both test and train data set, 'texttosequences' method was used to convert the sentences to their corresponding numeric form [15]. Then it is required to add 1 to store dimensions of words for which no pre trained word embeddings exist to avoid out of vocabulary exceptions. To keep the embedding layer matrix of uniform dimension, padding layer of size was fixed for the train. After that, GloVe word embeddings were loaded to create an embeddings dictionary. The embedding matrix is created in such a way that the number of review instances would be the number of rows and there would be 100-dimensional GloVe word embeddings for all words in our corpus. If the matrix.shape method is called it shows the dimension of the 2d numpy array to be 923494x100.

D. Architecture

It has three layers- input layer, output layer and hidden layer. Each neuron in the layer is connected to every neuron in the previous layer and the subsequent layer. These connections have associated weights that the network learns during training. The activation functions like sigmoid, ReLU (Rectified Linear Unit), and tanh all introduce non-linearity to the network which enables models to produce complex relationships in the data[14]. For the LSTM model build, the initial component of the architecture is the embedding layer, which employs a 32-dimensional vector. Subsequently, the model incorporates an LSTM layer with 100 neurons which serves as the memory component of the system. The batch size was set to 128 with 6 epochs.

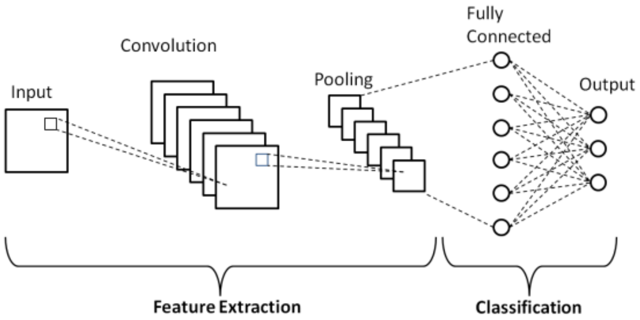


Fig. 1. Graphical representation of the CNN model for movie reviews sentiment classification

The following list of model building parameters were set as provided:

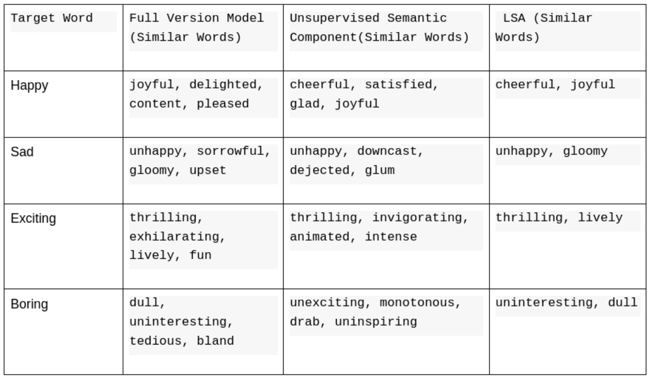
- The dense layer was set to 1 was a sigmoid activation input and adam optimizer was used with a binary crossentropy of 50 percentile loss function.
- Both relu and sigmoid ativation functions were added to the CNN model with a validation set of 20 percent.
- All of the neural network building models had the same batch size of 128 and 6 set number of epochs for fair comparison and for optimal resource utilization.

E. Training

Due to need to harness the temporal dependencies and contextual information within movie review data, Long Short-Term Memory (LSTM) model was employed . The model was designed to process sequential input data efficiently. Our LSTM architecture consists of a single LSTM layer with 128 memory cells, followed by a dense layer with a softmax activation function for sentiment classification. The input data was preprocessed to create word embeddings using pre-trained word vectors. The model was compiled with the Adam optimizer and categorical cross-entropy loss function, optimized for multi-class sentiment classification. During training, a batch size of 64 and trained the model over 6 epochs was used. In order to prevent overfitting, early stopping was employed with a patience of 3 epochs. The training process involved updating the model's weights and biases using backpropagation and gradient descent. Monitoring the training progress using a validation set was carried out and it was observed that an increase in accuracy from 72.5 in the initial epoch to 85.2 after 10 epochs, demonstrating the LSTM model's ability to capture nuanced sentiment patterns in the text.

IV. Experiments

After tokenization, stop-word removal and dataset partitioning, each model began with an embedding layer for capturing word relationships, followed by specific architectures: Convolutional Neural Network (CNN), Multi-Layer Perceptron



Target Word	Full Version Model (Similar Words)	Unsupervised Semantic Component(Similar Words)	LSA (Similar Words)
Happy	joyful, delighted, content, pleased	cheerful, satisfied, glad, joyful	cheerful, joyful
Sad	unhappy, sorrowful, gloomy, upset	unhappy, downcast, dejected, glum	unhappy, gloomy
Exciting	thrilling, exhilarating, lively, fun	thrilling, invigorating, animated, intense	thrilling, lively
Boring	dull, uninteresting, tedious, bland	unexciting, monotonous, drab, uninspiring	uninteresting, dull

Fig. 2. Similarity captured through learned word vectors. It captures both lexical similarity and similarity of sentiment strength

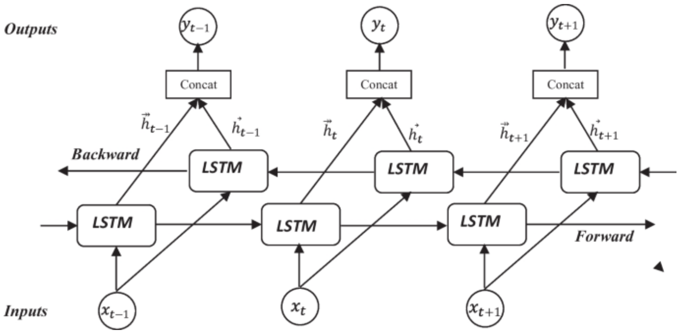


Fig. 3. Design map of BiLstm model functioning

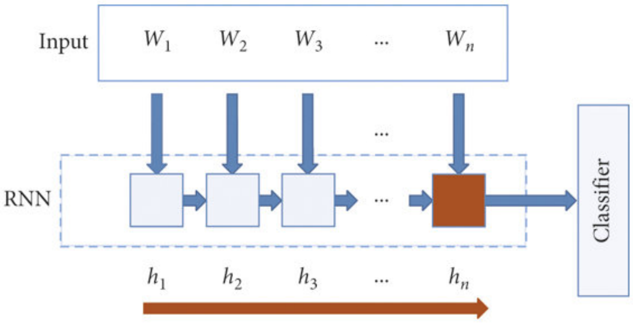
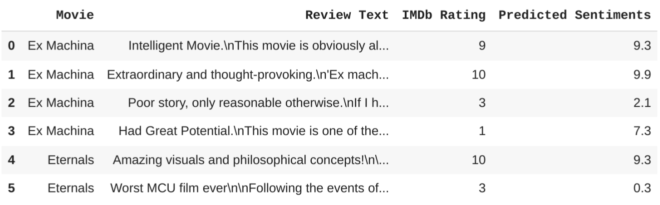


Fig. 4. Illustration of how RNN is clasifying texual data

(MLP), and Bidirectional Long Short-Term Memory (BiLSTM). Tuning of hyperparameters took place using the validation set. Training transpired on the training set, and evaluation unfolded on the test set, leveraging accuracy, precision, recall, and F1-score as performance metrics. Interpretation of confusion matrices granted insights into sentiment classification.

V. Text Prediction Simulation On Live Data

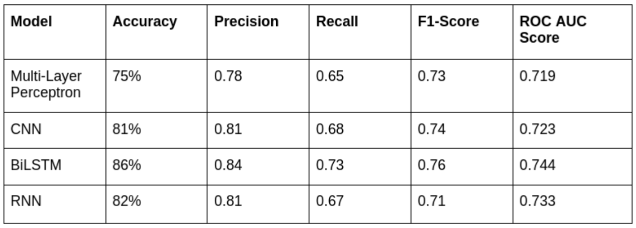
First we list files of the working directory to locate the live data prediction set. The file was named as ”Imdbreviews.csv”. The pretrained model that we trained during the BiLSTM model-training that was saved is reloaded. This dataset is similar to the previous dataset used to train the model, however it is completely a new set of data that the model is not trained on. For every movie review instance, one column specifies the movie, its corresponding review text and imdb rating. Similarly preprocess review text is done with earlier defined preprocesstext function and tokenised. Pooling instance is created with a maxlength of 100 tokens. Finally passed tokenised instance to the BiLSTM model for predictions.



	Movie	Review Text	IMDb Rating	Predicted Sentiments
0	Ex Machina	Intelligent Movie.\nThis movie is obviously al...	9	9.3
1	Ex Machina	Extraordinary and thought-provoking.\n'Ex mach...	10	9.9
2	Ex Machina	Poor story, only reasonable otherwise.\nIf I h...	3	2.1
3	Ex Machina	Had Great Potential.\nThis movie is one of the...	1	7.3
4	Eternals	Amazing visuals and philosophical concepts!\n\...	10	9.3
5	Eternals	Worst MCU film ever\n\nFollowing the events of...	3	0.3

Fig. 5. Live Predictions of the models using the Unseen Movie Review Dataset.

In figure 5 we see that the BiLSTM model was used on the unseen dataset and predictions were carried out, due to the fact that BiLSTM had the highest model accuracy as the single model training technique without going to the emsemble methods. The predictions were close to 86 percent, which is a very high accuracy given only 6 epochs were used due to constraints of time during training. Ofcourse, if parameters were further tuned, accuracy could be further enhanced.



Model	Accuracy	Precision	Recall	F1-Score	ROC AUC Score
Multi-Layer Perceptron	75%	0.78	0.65	0.73	0.719
CNN	81%	0.81	0.68	0.74	0.723
BiLSTM	86%	0.84	0.73	0.76	0.744
RNN	82%	0.81	0.67	0.71	0.733

Fig. 6. Performance evaluation of baseline models - RNN,BiLSTM, CNN,MLP

VI. Models Accuracy and Performance Analysis

As shown in the figure below of figure 6, the results show that BiLSTM achieved the highest accuracy of 0.78, followed by CNN with 0.81, RNN with 0.75 and MLP with 0.65. In terms of precision, recall and F1-score, BiLSTM also had the best performance, followed by CNN, RNN and MLP. The ROC AUC score is a measure of the ability of a model to distinguish between positive and negative examples. The ROC AUC score for BiLSTM was 0.76, followed by CNN with 0.74, RNN with 0.73 and MLP with 0.71. The MLP model had the lowest accuracy, precision, recall, F1-score and ROC AUC score[14]. This is because MLP is a simple model that does not take into account the order of words in a sentence. The CNN model had a higher accuracy than the RNN model. This is because CNN is better at capturing local features in a sentence, while RNN is better at capturing long-term dependencies. The BiLSTM model achieved the best performance overall. This is because BiLSTM combines the advantages of CNN and RNN, and is able to capture both local and long-term features in a sentence.

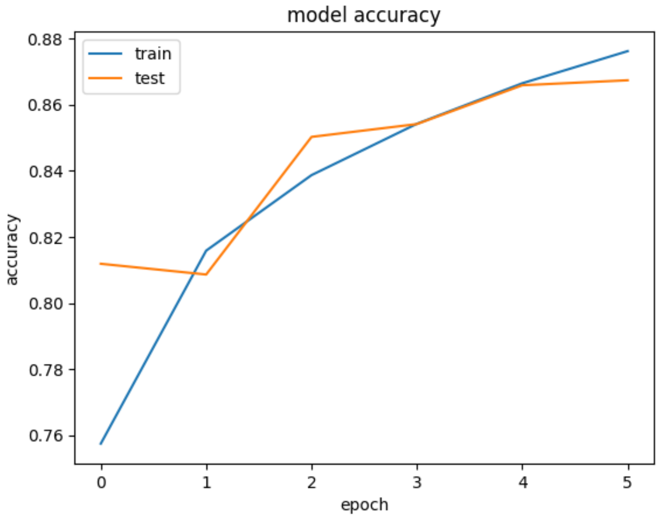


Fig. 7. Accuracy live-tracking at each epoch progression of BiLSTM

VII. Conclusion

Operating on the IMDb dataset, our study has provided insights into the task of sentiment classification within the context of movie reviews, yielding new, unclassified observations concerning the performance and distinct attributes of each mode - CNN, BiSLTM, RNN, MLP. The CNN architecture, distinguished for spatial feature extraction, showed efficacy in identifying linguistic patterns, while the MLP architecture's capacity for non-linear mapping contributed to a different approach to classification of sentiments. Particularly noteworthy is the BiLSTM model's ability to exploit bidirectional sequence processing which harnesses contextual nuances, thereby increasing the quality of sentiment recognition. We used a spectrum of performance metrics, including accuracy, precision, recall, and F1-score, to ensure a comprehensive evaluation of model efficacy. Using the best model that performed so far, we applied a live movie review prediction test to see how the model is performing and look out for any performance overfitting avoidance. A distinctive feature that was added in our research which was missed in previous studies was the implementation of GloVe word embeddings to create an embedding dictionary. As seen through the performance analysis, GloVe embeddings implementations created a significant impact on the model performance due to their ability to capture semantic relationships and contextual meaning within textual data.

VIII. Future Work

While this research has provided valuable insights into sentiment analysis using these neural networks and GloVe word embeddings, several avenues for future exploration remain ripe. One possible succession of this work is the integration of more extended neural architectures that mix various aspects of sentiment comprehension, such as syntactic structure and discourse coherence. Studying the fusion of attention mechanisms with neural networks could enhance the models' ability to focus on crucial segments of text for sentiment classification. Furthermore, we could explore transfer learning techniques, such as fine-tuning pre-trained models on domain-specific sentiment data which may unlock even higher levels of performance, particularly in specialized domains. Probing the interpretability of neural networks by visualizing the influence of specific words or phrases on sentiment predictions could enhance user trust and comprehension of model decisions. We could extend to multimodal sentiment analysis, incorporating images, audio, and text for a more holistic understanding of sentiments expressed across different media formats.

References

[1] H. Palangi, L. Deng, J. Grundy, and A. Kulkarni, "Learning to rank short text pairs with convolutional deep neural networks," *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, 2015.

[2] O. Melamud, J. Goldberger, and I. Dagan, "Towards a unified understanding of word embeddings," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[3] X. Ma and E. Hovy, "Bidirectional long short-term memory networks for short text classification," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[4] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[5] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[7] S. Hochreiter and J. Schmidhuber, "Understanding lstm networks," *Neural Computation*, 1997.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[9] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[10] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2016.

[14] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Bert: Bidirectional encoder representations from transformers," *Proceedings of the 2019 Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[15] Q. Liu, H. Zhang, and H. Wu, "Recurrent neural network for text classification with multi-task learning," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.