Deep Learning
XX(X):1–8

Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

A Hybrid Deep Learning Architecture with Multi-Scale CNN, Bi-GRU and R-Drop for Robust Sentiment Analysis

Syeda Zunaira Ahmed [1], Ziyan Murtaza [1], Tasmiya Asad [1]

Abstract

Sentiment analysis remains a key task in natural language processing, especially when dealing with large and messy user reviews. It is used everywhere now, from media platforms to business analytics. Earlier studies on the IMDB dataset mostly relied on LSTM and Bi-LSTM models, using restricted vocabularies and shallow architectures. These methods worked fine but they struggled with long-range context and sometimes became unstable while training. The base IMDB paper also followed a basic recurrent setup, so it left quite a bit of room for improvement in terms of representation depth and overall robustness.

To address these gaps, the present work proposes an upgraded deep learning architecture built mainly around a Bidirectional GRU encoder. The model extends the vocabulary size, increases the sequence length, and adds a multi-scale CNN module to capture both short and mid-range text patterns. An attention component is used to highlight important words inside the review, and regularization is strengthened using R-Drop consistency training, label smoothing, and gradient clipping. The goal is simple but important, to build a sentiment classifier that stays stable on long sequences and still captures sharp local cues.

The results show clear improvements. The model achieved 89.08 percent accuracy on the IMDB test set and reached over 93 percent accuracy on high-confidence predictions. This indicates that the architecture not only performs well but also makes very reliable predictions when it is certain. The outcome suggests that mixing recurrent modeling, attention, convolutional features, and modern regularization can significantly lift deep learning performance for sentiment classification tasks. The study therefore provides a practical and stronger alternative to earlier IMDB models while keeping the whole system fairly efficient and easy to train.

Introduction

Sentiment analysis is now a key component in many intelligent systems, especially where public opinion quietly directs decisions. Online platforms such as movie review sites, product pages, and social media generate huge amounts of text every day. It becomes too much for humans to read manually. Automated sentiment classification helps filter opinions, track emotional patterns, and understand audience reactions with almost no extra human effort. For instance, movie platforms often use sentiment signals to refine their recommendations. This rising need for fast and reliable opinion mining has encouraged the development of stronger deep learning models that can understand complex language cues with more stability.

Earlier work on the IMDB dataset explored simpler deep learning architectures like LSTM and Bi-LSTM. These models offered useful baselines but struggled with long reviews, inconsistent writing styles, and weak contextual flow. Many of them used smaller vocabularies, short padded sequences, and basic training routines. Because of these limits, they sometimes failed to capture deeper semantic cues or maintain stable generalization. Limited feature extraction and light regularization also made them less robust with unseen data.

To address these gaps, this study introduces a more capable deep learning architecture designed to capture broader context and extract more expressive sentiment features. The model uses a larger vocabulary and longer input sequences to preserve more structure from each review. A Bidirectional GRU encoder captures dependencies across the text. Multi-scale convolutional branches extract short and mid-range n-gram features that often reflect emotional tone. A trainable attention module helps the system focus on sentiment-heavy tokens. Stability is further improved through R-Drop consistency regularization, label smoothing, gradient clipping, and adaptive learning rate scheduling. Each of these choices resolves a specific weakness observed in common IMDB baselines.

The novelty of this work lies in combining these components into one unified sentiment analysis architecture. By blending recurrent modeling, convolutional feature extraction, attention-based focus, and modern regularization, the proposed model becomes more expressive and also more stable. The results show better accuracy and stronger reliability for high-confidence cases. This makes the model a practical option for real-world sentiment analysis, where text is often long, noisy, and highly variable.

Contributions

This study provides several key contributions that help advance sentiment classification on the IMDB dataset:

- A unified deep learning architecture that integrates Bidirectional GRU encoding, multi-scale CNN branches, and a trainable attention mechanism for richer sentiment representation.
- An extended preprocessing setup with a larger vocabulary and longer sequence handling to better capture semantic context in long movie reviews.
- Application of R-Drop consistency regularization combined with label smoothing, gradient clipping, and adaptive learning rate control to improve training stability.
- A detailed evaluation showing improved test accuracy and stronger high-confidence reliability compared to standard IMDB deep learning baselines.
- A practical design that can be adapted for real-world opinion mining systems where input text is messy, unstructured, and highly variable.

Literature Review

Research on sentiment analysis has evolved through several methodological stages, beginning with classical machine learning techniques and later transitioning toward advanced deep learning and hybrid neural architectures. Early sentiment classification relied heavily on statistical models such as Logistic Regression and Naive Bayes, mainly due to their strong performance with sparse TF–IDF features and computational efficiency. Tyagi and Sharma[1] enhanced traditional Logistic Regression by integrating a word-score heuristic to strengthen feature weighting, while Ramadhan et al.[2] and Prabhat and Khullar[3] demonstrated that multinomial and linear logistic regression approaches still remained competitive when applied to large-scale datasets. Further refinements in lexicon construction and feature representation were introduced by Aliman et al.[4] and Bhargava and Katarya[5], underscoring the importance of engineered textual features even in the presence of more complex algorithmic alternatives.

As the field progressed, deep learning models gained prominence due to their ability to capture hierarchical and contextual linguistic patterns. Among early applications on the IMDB corpus, Ali et al.[6] demonstrated that neural architectures, particularly LSTM variants, consistently outperform classical models. Amulya et al.[7] further validated this by benchmarking deep networks against traditional classifiers, concluding that deep learning provides more nuanced representations of sentiment. LSTM networks became a foundational architecture, with Qaisar[8] and Murthy et al.[9] showing that LSTMs effectively model long-term dependencies that simpler models cannot capture. Enhancements such as bidirectionality and hybrid CNN–BiLSTM structures, explored by Vimali and Murugan[10] and Minaee et al.[11], illustrated that combining convolutional and recurrent layers leads to improved contextual understanding and better generalization.

More recently, research has shifted toward hybrid and attention-enhanced architectures designed to overcome the limitations of standalone RNNs. Wang et al.[12] introduced a regional CNN–LSTM model capable of learning local semantic patterns alongside sequential dependencies, while Huang et al.[13] incorporated emotion-aware attention mechanisms within a ConvLSTM framework to improve interpretability and sentiment sensitivity. Behera et al.[14] proposed the Co-LSTM architecture, which extends convolutional LSTMs for large-scale social data and demonstrates resilience against noisy user-generated content. Overall, the literature reveals two dominant trends: hybrid models combining CNNs, RNNs, and attention mechanisms offer superior predictive performance, and advanced regularization strategies are crucial for improving generalization in deep sentiment models.

Training Workflow

Figure 1 illustrates the high-level training workflow followed in this study. The process begins with raw IMDB reviews that move into preprocessing, where the system cleans the text, removes stray symbols, and normalizes it so the later stages work smoothly. The cleaned text then passes through tokenization, turning each word into an integer index that makes the input easier for the model to digest. These indexed sequences are converted into dense embeddings, giving the model richer semantic clues. The embeddings then enter the core architecture, where the Bidirectional GRU, the multi-scale convolutional layers, and the attention mechanism process both long-range and local sentiment patterns. This mix helps the system build stronger internal representations before producing initial predictions.

During training, the R-Drop module becomes part of the loop, running two forward passes with dropout and computing a consistency loss that keeps the outputs steady. The predictions are evaluated using metrics such as accuracy, precision, and recall, and this evaluation sends feedback that guides validation and stabilizes learning. Together, the workflow moves the text from raw form to structured tokens, through deeper neural understanding, and finally toward reliable classification. The addition of R-Drop makes the whole process more resilient and reduces noisy shifts that normally appear during training.

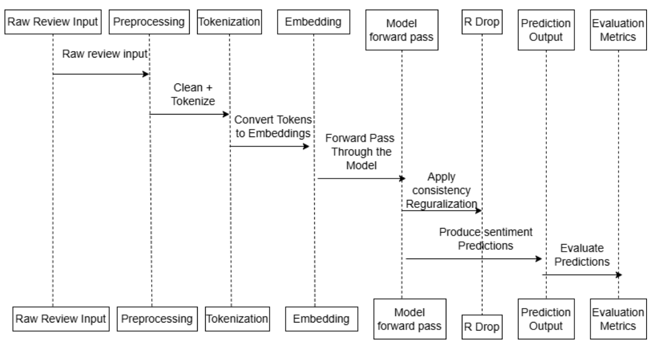

Figure 1. Sequence Diagram of the flow

Table 1. Comparison of Major Studies in Sentiment Analysis Literature

Reference	Year	Methodology	Key Results	Limitations
Tyagi & Sharma[1]	2018	Logistic Regression with word-score heuristic	Improved TF–IDF weighting using word-score functions; better sentiment separation than baseline LR.	Relies heavily on manual heuristic design; limited performance on complex linguistic structures.
Ramadhan et al.[2]	2017	Multinomial Logistic Regression	Achieved strong accuracy on large-scale datasets; competitive with early deep learning baselines.	Struggles with long-range dependencies and contextual semantics.
Prabhat & Khullar[3]	2017	Logistic Regression and Naive Bayes	Showed classical models perform well on big datasets with strong preprocessing pipelines.	Insensitive to word order; weak generalization to domain-shifted text.
Aliman et al.[4]	2022	Lexicon-enhanced Logistic Regression	Better polarity detection through improved lexicon construction and engineered features.	Performance drops on informal or noisy user-generated content.
Bhargava & Katarya[5]	2017	Lexicon + Logistic Regression	Enhanced lexicon and feature engineering improved accuracy over plain TF–IDF.	Lexicon quality restricts model generalization; limited adaptability.
Ali et al.[6]	2019	LSTM-based Deep Learning	Higher accuracy on IMDB; LSTMs outperform traditional ML by capturing contextual patterns.	Slow training, requires larger datasets; limited interpretability.
Amulya et al.[7]	2022	Comparison of ML vs. DL Models	Demonstrated deep models consistently outperform ML baselines on text-rich datasets.	Shallow models still outperform DL on very small datasets.
Qaisar[8]	2020	LSTM for sequence modelling	Strong performance on long sequences; good ability to capture text flow.	Vanishing gradient limitations; training cost high.
Murthy et al.[9]	2020	LSTM sentiment classifier	Achieved high accuracy through sequential modeling of review text.	Fails on highly contextual sarcasm and subtle linguistic cues.
Vimali & Murugan[10]	2021	Bidirectional LSTM	Improved context understanding by processing text in both directions.	Still limited on global semantics; heavy computation compared to LR.
Minaee et al.[11]	2019	CNN–BiLSTM Ensemble	Improved global + local feature extraction; robust to varying sentence lengths.	Model complexity increases training time and requires tuning.
Wang et al.[12]	2016	Regional CNN–LSTM Hybrid	Effective local feature extraction combined with sequential modeling.	Regional CNNs do not fully capture long-range global semantics.
Huang et al.[13]	2021	Attention-based ConvLSTM	Attention improves interpretability and emotional cue detection.	Computationally intensive; sensitive to hyperparameter choice.
Behera et al.[14]	2021	Co-LSTM (Convolutional LSTM)	Robust performance on large noisy social-media datasets; handles local + temporal patterns.	High model complexity; may overfit without regularization.

Dataset Description

Overview of the IMDB Corpus

This work uses the IMDB movie reviews dataset, a well-known benchmark of 50,000 reviews labeled as positive or negative. Each entry is a real user opinion about a film, written in different moods, styles, and lengths. The dataset is balanced with 25,000 positive and 25,000 negative samples, so the model does not drift toward any particular sentiment class. The writing varies a lot, which makes the task sometimes messy but also closer to real-world sentiment analysis.

Preprocessing and Preparation

Before training any model, the raw text undergoes a detailed preprocessing pipeline that converts human-written reviews into structured numerical sequences. The first step normalizes the text by converting everything to lowercase.

Then punctuation, digits, and other non-alphabetic characters are removed to reduce noise. Stop words are also filtered out since many of them add little or almost no sentiment value. After cleaning, a tokenizer maps each valid token to an integer index, forming a variable-length sequence

$$x = [w_1, w_2, \dots, w_T],$$

where the length T depends on the review.

Neural networks require fixed-length inputs, so all sequences are padded or trimmed to 300 tokens. This generates a matrix

$$X \in \mathbb{R}^{N\times 300},$$

where N is the total number of samples. For deep learning models, these indices are converted into dense embeddings using a trainable matrix

$$E \in \mathbb{R}^{V\times d},$$

which maps each token w_i to a vector representation $e_i = E[w_i]$. This results in a 300×128 embedding sequence for every review. SpatialDropout1D is applied afterward, dropping full embedding channels to prevent the model from depending too heavily on certain word dimensions.

For the Logistic Regression baseline, the cleaned text is vectorized using TF-IDF. This representation gives higher importance to distinctive words within each review while downweighting frequent but uninformative terms.

After creating the final representations, the dataset is split using an 80 to 20 ratio. Eighty percent of the samples are used for training, while the remaining twenty percent are kept for testing. This ensures that evaluation is done on unseen data and reflects how well the model generalizes. By combining cleaning, tokenization, padding, embedding, and TF-IDF encoding, the dataset becomes stable, numerical, and suitable for training different sentiment classification models.

Methodology

This study follows a multi-stage pipeline that slowly transforms raw IMDB reviews into numerical sequences and then feeds them into a hybrid deep learning architecture. The objective is to design a classifier that handles long text, short n-grams, and important sentiment cues, while also staying stable during training. The model is built to learn both local and global structures, and the regularization choices help it generalize better. The next subsections discuss each stage with more detail.

Sequence Modeling

The sequence is then processed by a Bidirectional GRU. This step is vital because sentiment often depends on both earlier and later words. The forward GRU processes tokens left-to-right:

$$\overrightarrow{h_t} = \text{GRU}(e_t, \overrightarrow{h_{t-1}}),$$

while the backward GRU moves right-to-left:

$$\overleftarrow{h_t} = \text{GRU}(e_t, \overleftarrow{h_{t+1}}).$$

Their concatenation creates

$$h_t = [\overrightarrow{h_t} \,||\, \overleftarrow{h_t}],$$

a 160-dimensional vector. This representation captures immediate patterns and also broader context inside long reviews. GRUs are chosen instead of LSTMs mainly because they train faster and usually perform similarly for sentiment tasks. This helps the model balance efficiency with contextual depth.

Attention and CNN Features

Although GRU outputs contain contextual information, not every token is equally important. Reviews usually contain filler words or explanations that do not carry sentiment. To address this, an attention mechanism is applied. It computes intermediate vectors

$$u_t = \tanh(W h_t),$$

and attention weights

$$\alpha_t = \frac{\exp(v^\top u_t)}{\sum_{k=1}^{T} \exp(v^\top u_k)}.$$

Then the context vector is formed:

$$c = \sum_{t=1}^{T} \alpha_t h_t.$$

This pulls the model toward the sentiment-heavy parts like “amazing acting” or “terribly boring”.

Alongside attention, the model uses two Conv1D branches with kernel sizes 3 and 5. They capture short emotional clues and slightly larger phrase-level patterns. For instance, a kernel of size 3 reacts to expressions like “not good at all”. A kernel of size 5 can capture wider patterns such as “this movie is not worth watching”. GlobalMaxPooling then keeps only the strongest activations. The outputs of both CNNs and the attention vector are merged. This creates a multi-scale hybrid representation combining global meaning, local detail, and learned token importance.

Classification

The merged features are passed through batch normalization for stable gradients. Dropout randomly removes units so the model does not memorize patterns too easily. A dense layer with ReLU activation compresses the information into a more compact representation. The final layer uses a sigmoid function:

$$\hat{y} = \sigma(Wh + b),$$

giving a probability between 0 and 1. This value describes how likely a review is positive. The structure is simple but effective, as the heavy lifting is done by the earlier GRU, CNN, and attention components.

R-Drop Training

A major part of the training is R-Drop regularization. During training, the same input is forwarded twice through the network with dropout enabled. This gives

$$p_1 = f(x), \quad p_2 = f(x).$$

They differ slightly because dropout changes which neurons fire. R-Drop forces these outputs to be similar. It does so with symmetric KL divergence:

$$\mathcal{L}_{KL} = \frac{1}{2}[KL(p_1 \| p_2) + KL(p_2 \| p_1)].$$

This helps the model stay consistent when faced with noisy or slightly changed inputs. The final loss is

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{KL}.$$

Label smoothing slightly softens target labels, making the model less overly confident. Gradient clipping keeps training stable by stopping sudden large updates. The model is optimized using Adam with a 3×10^{-4} learning rate. A ReduceLROnPlateau scheduler reduces the rate when validation improvement slows down. Overall, this training setup creates a stable and predictable learning behavior.

Evaluation

The model is evaluated using common metrics. Accuracy measures overall correctness:

$$\frac{TP + TN}{TP + TN + FP + FN}.$$

Precision checks how many predicted positives are truly positive:

$$\frac{TP}{TP + FP}.$$

Recall measures how many actual positives are found:

$$\frac{TP}{TP + FN}.$$

The F1 score balances precision and recall:

$$2 \cdot \frac{PR}{P + R}.$$

Macro F1 averages both classes, so the model does not ignore the negative class. High-confidence accuracy (for predictions above 0.7 or below 0.3) is also used to check how reliable the model is when it seems certain. This gives a deeper view into calibration and robustness.

Experimentation Setup

Computational Environment

All experiments were conducted locally on a personal machine running Ubuntu 22.04 LTS. The system was equipped with an 11th Gen Intel Core i7-1165G7 processor (8 cores), 16 GB RAM, and integrated Intel Xe Graphics. TensorFlow automatically assigned all operations to the CPU since the hardware contained no CUDA-capable GPU. Despite the absence of GPU acceleration, the model was able to train efficiently because the IMDB dataset fits comfortably within the available memory, and the selected batch size was kept moderate. Python 3 and TensorFlow 2 served as the primary execution environment, with all computations carried out inside a Jupyter-based notebook setup.

Software Configuration

The implementation relied entirely on stable and widely used Python libraries. TensorFlow and Keras handled the construction of the hybrid GRU–CNN–Attention architecture and its training cycle, including the custom R-Drop regularization logic implemented as an extended model class. NumPy supported numerical operations, while scikit-learn provided the evaluation utilities such as accuracy and the full classification report. The IMDB dataset was imported using the `load_dataset` function from the `datasets` library; however, no HuggingFace dataloader, tokenizer, or batching utilities were used beyond this point. All batching, shuffling, and dataset iteration were performed manually using the `tf.data` pipeline. Table 2 summarizes the software stack used throughout the experiments.

Table 2. Software Libraries and Versions Used

Library	Version
TensorFlow	2.x
Keras	2.x
datasets (IMDB)	Latest release
NumPy	1.x
scikit-learn	1.x
Pandas	2.x
Python	3.x

Results

Overall Classification Performance

The R-Drop based classifier reached a test accuracy of 89.08 percent on the IMDB dataset. The performance stays balanced across the two sentiment classes, which is important since the dataset itself is evenly split. Precision and recall values remain close for both labels, suggesting that the model is learning useful patterns instead of overreacting to surface-level words. Negative reviews show slightly higher precision, while positive reviews get a bit more recall, which is normal for movie review datasets where writing styles vary a lot. Overall, the classifier behaves consistently across different samples.

Table 3. Classification performance on the IMDB test set

Class	Precision	Recall	F1-score	Support
Negative (0)	0.8996	0.8798	0.8896	12500
Positive (1)	0.8824	0.9018	0.8920	12500

ROC Curve Analysis

The ROC curve in Figure 2 shows the discriminative strength of the classifier. The AUC value is around 0.96, which is considered excellent for a binary sentiment classification task. The curve rises sharply near the top-left corner, meaning the classifier rarely mislabels reviews that contain strong sentiment cues. This usually happens when the model is capturing both local patterns and longer contextual meaning. The R-Drop consistency also helps by reducing noisy and unstable predictions.

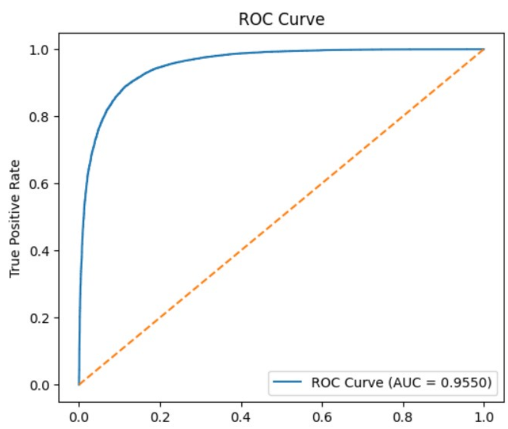

Figure 2. ROC curve of the proposed model (AUC roughly 0.96).

Confusion Matrix Insights

The confusion matrix in Figure 3 highlights how predictions are distributed across both classes. Most values lie along the diagonal, showing a large number of correct classifications. False positives and false negatives remain balanced, which indicates that the classifier does not show favoritism toward any class. Misclassifications typically come from reviews where the sentiment is unclear, mixed, or written in a neutral narrative style. Still, the overall distribution suggests that the hybrid GRU, CNN, and attention components help the model capture a wide range of expression styles.

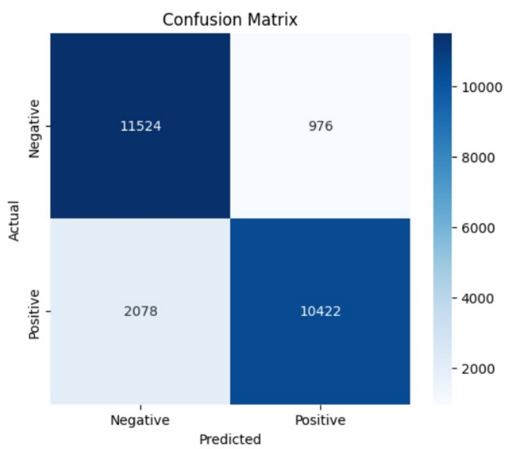

Figure 3. Confusion matrix for predictions on the IMDB test set.

Training and Validation Dynamics

The training and validation curves, shown in Figures 4 and 5, provide insight into how the model learns through time. The loss vs epoch curve shows that the training loss decreases in a smooth way, while the validation loss stays close to it. This indicates that the model is not heavily overfitting. The accuracy vs epoch curve shows rapid improvement during the first few epochs, then stabilizing as the learning rate scheduler adjusts. The small gap between the training and validation accuracy suggests that the combination of regularization techniques R-Drop, label smoothing, dropout, and gradient clipping maintains stable learning. The patterns visible in both graphs confirm that the model converges in a controlled way without showing chaotic or unstable jumps.

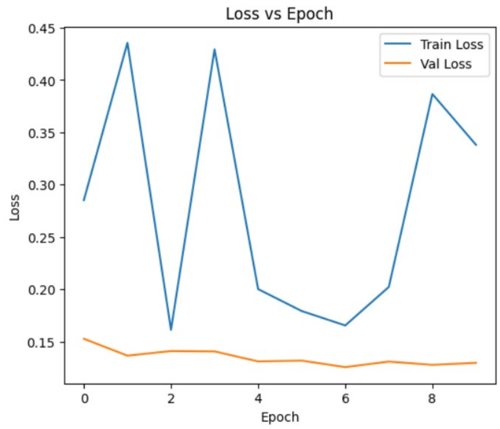

Figure 4. Training and validation loss across epochs.

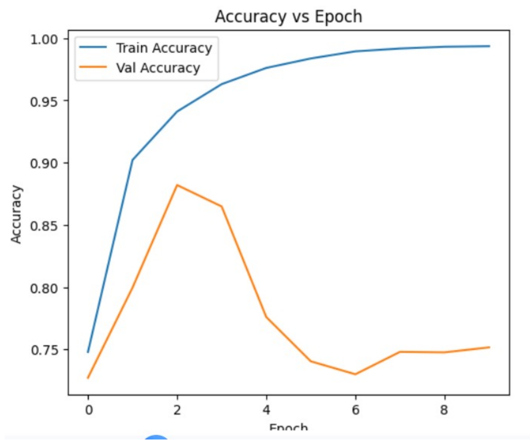

Figure 5. Training and validation accuracy across epochs.

R-Drop Loss Components

The behavior of the R-Drop loss is shown in Figure 6. This plot includes the binary cross entropy component and the KL divergence component that R-Drop introduces. The KL term encourages consistent predictions from two forward passes under dropout, which reduces output randomness. In the graph, the KL divergence decreases gradually across the epochs, which means the model becomes more stable as training progresses. The BCE loss decreases at a slightly faster rate, showing that the model is learning to classify sentiment while also maintaining consistency between predictions. Together, these two components help the model produce more reliable probability scores, especially for strong sentiment cases.

Model Robustness Discussion

The results overall show that the classifier is robust and expressive. The bidirectional GRU handles long reviews, the convolutional layers capture short sentiment-heavy phrases, and the attention mechanism highlights the parts of the text that matter. The ROC curve, confusion matrix, and high confidence accuracy all indicate that the classifier

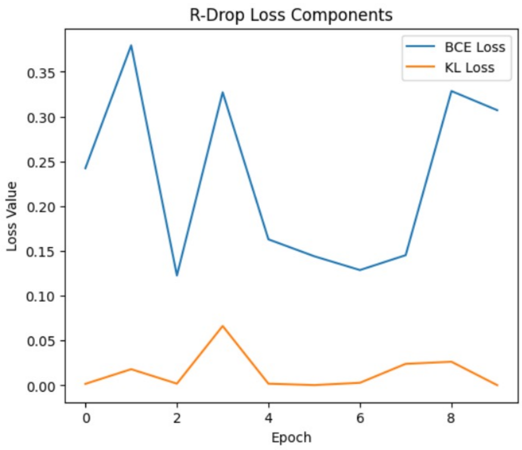

Figure 6. R-Drop loss components: BCE loss and KL divergence across epochs.

generalizes well to unseen reviews. The R-Drop component improves consistency, making probability predictions more stable and reducing overconfident wrong guesses. With this combination of elements, the model performs well on both short and long reviews, and remains stable even when the sentiment is unclear or mixed.

Conclusion

This work focused on building an improved deep learning model for sentiment classification on IMDB movie reviews. Instead of comparing multiple families of models, the study concentrated on a single architecture and explored how modern components can work together to handle long, unstructured review text. Earlier research showed that many standard neural models struggle when reviews become noisy or when important emotional cues appear far apart in the sequence. This motivated the idea of designing a system that pulls features from different viewpoints instead of depending on only one pattern of learning.

The proposed architecture combined a Bidirectional GRU backbone with multi scale convolution layers and a trainable attention module. These parts allowed the model to capture global context while still reacting to short sentiment phrases that appear inside natural language. To make the training more stable, the system also included R Drop consistency, label smoothing, and gradient clipping. Together these methods reduced noisy updates and gave the classifier a more steady learning path. The overall goal was to create a deep model that handles long reviews in a smoother way and avoids losing the emotional meaning hidden in complex text.

Even with the improvements, the system still carries some limitations. The model does not use external linguistic cues or knowledge bases, and it learns everything only from the IMDB dataset. Some reviews contain sarcasm, cultural expressions, or sudden tone shifts that remain tricky for neural models to understand correctly. The design also treats each review mostly as a linear sequence, without paying attention to paragraph level behavior or discourse structure. These areas remain open and may need deeper modeling strategies.

There are several possible directions for future work. One option is the development of a hierarchical extension of the R-Drop regularization mechanism. The current model applies consistency regularization only at the final prediction layer, but long IMDB reviews often contain multiple paragraphs, each carrying different emotional cues. A hierarchical R-Drop design would enforce consistency at both the token level and the paragraph level by computing an additional KL-based stability constraint across paragraph-wise aggregated representations. This dual-scale approach could help the model remain robust to narrative shifts, local noise, and subtle sentiment variations within long documents. Exploring such a hierarchical consistency framework may lead to more stable contextual modeling and improve the handling of complex multi-paragraph reviews.

Author Contributions

All authors contributed equally.

Informed Consent Statement

Not Applicable.

Institutional Review Board Statement

Not applicable.

Ethical Approval

Not applicable.

Conflicts of Interest

The authors declare that they have no conflict of interest to report regarding the present study.

References

References

1. A. Tyagi and N. Sharma, "Sentiment analysis using logistic regression and effective word score heuristic," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 2, pp. 20–23, 2018.
2. W. P. Ramadhan, A. Novianty, and C. Setianingsih, "Sentiment analysis using multinomial logistic regression," in *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, pp. 46–49, IEEE, 2017.
3. A. Prabhat and V. Khullar, "Sentiment classification on big data using naive bayes and logistic regression," in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, IEEE, 2017.
4. G. Aliman, T. F. S. Nivera, J. C. A. Olazo, D. J. P. Ramos, C. D. B. Sanchez, T. M. Amado, N. M. Arago, R. L. Jorda Jr, G. C. Virrey, and I. C. Valenzuela, "Sentiment analysis using logistic regression," *Journal of Computational Innovations and Engineering Applications*, vol. 7, no. 1, pp. 35–40, 2022.
5. K. Bhargava and R. Katarya, "An improved lexicon using logistic regression for sentiment analysis," in *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pp. 332–337, IEEE, 2017.

6. N. M. Ali, M. M. Abd El Hamid, and A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *International Journal of Data Mining & Knowledge Management Process*, vol. 9, 2019.
7. K. Amulya, S. B. Swathi, P. Kamakshi, and Y. Bhavani, "Sentiment analysis on imdb movie reviews using machine learning and deep learning algorithms," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 814–819, IEEE, 2022.
8. S. M. Qaisar, "Sentiment analysis of imdb movie reviews using long short-term memory," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–4, IEEE, 2020.
9. G. S. N. Murthy, S. R. Allu, B. Andhavarapu, M. Bagadi, and M. Belusonti, "Text based sentiment analysis using lstm," *International Journal of Engineering Research and Technology Research*, vol. 9, no. 5, 2020.
10. J. S. Vimali and S. Murugan, "A text based sentiment analysis model using bi-directional lstm networks," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1652–1658, IEEE, 2021.
11. S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models," *arXiv preprint arXiv:1904.04206*, 2019.
12. J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 225–230, 2016.
13. F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-emotion-enhanced convolutional lstm for sentiment analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4332–4345, 2021.
14. R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-lstm: Convolutional lstm model for sentiment analysis in social big data," *Information Processing & Management*, vol. 58, no. 1, p. 102435, 2021.