Sentiment Analysis of IMDb Reviews Using DistilBERT and a Custom Feedforward Network

Anhad Kashyap[1] **& Vinay Vishwakarma**[2]

Received October 7, 2025
Accepted December 27, 2025
Electronic access February 15, 2026

This research investigates the use of transformer-based embeddings together with feedforward neural architectures for binary classification of sentiment in text data. We used the IMDb movie review dataset as a test dataset and employed DistilBERT to generate fixed-length contextual embeddings to subsequently pass into a feedforward neural network that we built in PyTorch. We conducted multiple rounds of experimentation to adjust the hyperparameters of learning rate, batch size, and model depth and width. The best performing version of the model passed to the test set achieved a test accuracy of 86.6 percent, suggesting the capability of a hybrid model. The results showed that hyperparameter optimization does matter and combining transformer-based embeddings with thin classifiers provides an effective option for sentiment analysis. The model also vastly generalized, with stable performance on multiple randomized train–test splits. Finally, deeper feedforward layers provided diminishing returns compared to learning rates and regularization. Overall, research illustrates transformer-based embeddings greatly simplify downstream architectures without sacrificing classification performance.

Keywords: Transformer embeddings, DistilBERT, Sentiment analysis, Feedforward neural network, IMDb dataset, PyTorch, hyperparameter tuning

Introduction

Sentiment analysis of user-generated reviews is widely used in recommendation systems and industry applications. Existing methods range from traditional machine-learning classifiers to transformer-based architectures with varying computational cost. This project aims to create and optimize an Artificial Intelligence model to recognize the leading sentiment in movie reviews taken from the IMDb database. We used a mixture of pre-existing tools and a custom-built model to perform this task. A pre-existing Large Language Model (LLM) and a tokenizer to produce a sentence embedding and then use the embeddings of our data to train a small feedforward network.

Tokenizers convert text into integer token IDs that can be processed by neural networks and the Large Language Model here is used to transform the tokenized vector into a sentence encoding vector that can be processed by our feedforward sentiment analysis model in a more efficient way.

The problem this model sets out to solve is interesting because it may be used to fuel further research based on the public opinion of various movies or shows. For example, since the model can gauge whether a movie review's leading sentiment is positive or negative, this model may be used to recognize sentiments of multiple movies and a hypothesis may be drawn linking characteristics of the movies or their casts and the reviews (and by proxy, their reception).

The model may also be repurposed to gauge the leading sentiment of other texts, such as book reviews, product reviews, or comments on internet content. These may also broaden the scope of research that this model (or its repurposed variants) support.

Literature review

Sentiment analysis has emerged as a crucial research domain in natural language processing, particularly for understanding public opinions expressed in textual data such as movie reviews. The proliferation of online platforms has created vast repositories of user-generated content, making automated sentiment analysis increasingly important for businesses and researchers. This literature review examines 15 key papers that provide foundational and contemporary insights into sentiment analysis techniques, with particular focus on movie reviews, deep learning approaches, and transformer-based models relevant to this research project.

Bashiri and Naderi conducted a systematic review and comparative analysis of major transformer architectures including BERT, RoBERTa, DistilBERT, ALBERT, and others across multiple sentiment analysis datasets. They implemented standardized evaluation protocols and performed extensive bench-

[1] *Vasant Valley School, India*
[2] *On My Own Technology Pvt. Ltd., Lokhandwala, Oshiwara, Mumbai, India.*

marking to assess performance, computational efficiency, and practical deployment considerations, This comprehensive review focuses on understanding the relative strengths and weaknesses of different transformer variants for sentiment classification tasks, with particular emphasis on the trade-offs between model size, accuracy, and computational requirements. The study aims to provide practical guidance for researchers and practitioners in selecting appropriate transformer models for sentiment analysis applications The authors noted limitations in cross-dataset generalizability and highlighted that evaluation metrics varied across studies, making direct comparisons challenging. The review also identified gaps in long-text analysis and limited exploration of domain-specific adaptations. Additionally, the study acknowledged that deployment-specific constraints (memory, latency) were not uniformly considered across all compared approaches [1].

Areshey and Mathkour implemented and compared five major transformer models (BERT, RoBERTa, ALBERT, DistilBERT, and XLNet) on multiple sentiment classification datasets using standardized fine-tuning procedures. They measured accuracy, F1-score, training time, and inference speed across different dataset sizes and text lengths. This empirical study focuses on providing a systematic comparison of popular transformer architectures specifically for sentiment classification tasks, examining both performance metrics and practical considerations for deployment. The research aims to establish clear guidelines for model selection based on specific use case requirements,RoBERTa and XLNet consistently outperformed other models on most benchmarks, achieving the highest accuracy scores. DistilBERT demonstrated the best balance of performance and efficiency, with only modest accuracy reduction compared to larger models while offering significant speedup in training and inference. The research identified significant dataset-specific performance variations that limit generalizability of findings. The study was limited to relatively short text sequences and did not extensively explore hybrid architectures or ensemble methods [2].

Papia et al. developed a novel hybrid architecture that combines DistilRoBERTa embeddings with bidirectional LSTM networks and attention mechanisms. They implemented a fusion strategy that integrates transformer-based contextual representations with sequential modeling capabilities and evaluated the approach on multiple sentiment analysis datasets including IMDb This study focuses on creating an efficient hybrid model that leverages the strengths of both transformer-based embeddings and recurrent neural networks for sentiment analysis. The research aims to address the computational limitations of full transformer models while maintaining high accuracy through strategic architectural fusion On the IMDb dataset, the model achieved 91.2% accuracy while maintaining faster inference times than full transformer models. The authors noted increased architectural complexity compared to simple feedforward approaches, requiring more careful hyperparameter tuning. The model's performance gains were dataset-dependent, and the approach required additional memory overhead for the LSTM components [3].

Hany and Gomma conducted a systematic evaluation of classical machine learning models, recurrent neural networks, and transformer-based approaches across different text length categories on the IMDb dataset. They stratified the data by review length and measured performance degradation patterns as text length increased, The study's findings were specific to the IMDb dataset and may not generalize to other domains with different text characteristics. The length stratification approach, while systematic, may not reflect real-world text distributions. The research also noted that preprocessing choices significantly affected results, making standardized comparison challenging [4].

Saad et al. developed a transformer-based architecture specifically optimized for movie review sentiment analysis, incorporating attention mechanisms and custom preprocessing pipelines. They implemented various transformer variants and compared their performance on IMDb and other movie review datasets with focus on binary classification accuracy The optimized transformer approach achieved 89.7% accuracy on IMDb reviews, The approach was specifically tailored to movie reviews and may not generalize well to other sentiment analysis domains. The custom optimizations increased model complexity and training time. The study also noted sensitivity to hyperparameter choices and the need for extensive validation to avoid overfitting to the specific dataset characteristics [5].

Papadimitriou et al. implemented comprehensive fine-tuning strategies for BERT on IMDb reviews, exploring different learning rates, batch sizes, sequence lengths, and regularization techniques. They conducted systematic hyperparameter optimization and compared various training strategies to identify optimal configurations This research focuses specifically on optimizing BERT fine-tuning for IMDb sentiment classification, providing detailed analysis of hyperparameter sensitivity and training strategies. Optimal hyperparameter configuration achieved 92.1% accuracy on the IMDb test set, with learning rate and batch size being the most critical parameters. The study identified that careful regularization and learning rate scheduling significantly improved generalization. The findings were specific to BERT and IMDb dataset, with limited exploration of other transformer variants like DistilBERT [6].

Petridis conducted a comprehensive survey of text classification approaches, systematically comparing classical machine learning methods, neural networks, and pre-trained transformer models across multiple datasets and tasks. This research provides a systematic overview of the evolution from shallow to deep learning approaches in text classification, with emphasis on understanding when and why different ap-

proaches are most effective. The study aims to guide practitioners in choosing appropriate methods based on specific requirements and constraints Pre-trained transformers, including DistilBERT, consistently outperformed classical and shallow neural methods across most benchmarks. However, the study found that pipeline decisions (tokenization, preprocessing, feature engineering) often had more impact than model choice. The survey highlighted significant heterogeneity in experimental setups across different studies, making direct comparisons challenging. The research noted the lack of standardized evaluation protocols and the difficulty of reproducing results across different implementations [7].

Jahan and Rahman developed diagnostic methods using Hessian matrix analysis to understand failure modes and optimization landscapes in attention-based models including DistilBERT. They applied second-order optimization insights to improve training stability and performance in sentiment classification tasks Hessian analysis revealed important insights about optimization difficulties in transformer training, identifying specific layers and attention heads that contribute to training instability.The mathematical complexity of the approach limits its accessibility to practitioners without strong optimization backgrounds. The computational overhead of Hessian analysis makes it impractical for very large models or extensive hyperparameter searches [8].

Florencio et al. developed methods for implementing neural networks, including simplified transformer architectures, that can operate on encrypted data throughout the entire inference pipelineThis research addresses privacy concerns in sentiment analysis by developing techniques that allow neural network inference on encrypted text data.The approach required substantial computational overhead, making real-time applications challenging. The method was limited to simpler architectures and could not support full transformer complexity. Additionally, the study noted significant implementation complexity and the need for specialized cryptographic expertise [9].

Kerasiotis et al. developed a hybrid system combining DistilBERT embeddings with auxiliary features (user metadata, posting patterns, linguistic features) for detecting depression indicators in social media posts. They implemented a feedforward classifier that fused transformer embeddings with engineered features The hybrid approach combining DistilBERT with auxiliary features achieved 87.3% accuracy in depression detection, significantly outperforming either component alone; the approach required careful handling of sensitive data and raised ethical considerations about privacy and consent. The model's performance was highly dependent on the quality and availability of auxiliary features. [10].

Nguyen et al. applied large language models and transformer architectures to automate the design of microfluidic systems, adapting text classification techniques for engineering design tasks. They used DistilBERT-based approaches for classifying and generating design specifications. e adapted transformer approach successfully automated significant portions of the design process, achieving 85% accuracy in design classification tasks. The study noted challenges in handling technical terminology and the need for specialized training data. Additionally, the complexity of engineering constraints made direct application of standard NLP techniques insufficient without significant modification [11].

Leteno et al. conducted systematic analysis of gender bias in BERT and DistilBERT models, examining how these biases affect sentiment classification performance across different demographic groups. They developed methods for detecting and measuring bias in transformer embeddings and classification outputs. Bias detection and mitigation methods added computational overhead and complexity to the deployment pipeline. The study noted that bias patterns were context-dependent and difficult to predict across different applications. Additionally, mitigation strategies sometimes reduced overall accuracy, creating trade-offs between fairness and performance. [12].

N. D. Khan et al Bias detection and mitigation methods added computational overhead and complexity to the deployment pipeline. The study noted that bias patterns were context-dependent and difficult to predict across different applications. Additionally, mitigation strategies sometimes reduced overall accuracy, creating trade-offs between fairness and performance. Transfer learning from general sentiment models to quantum software classification achieved 82% accuracy with appropriate domain adaptation. The approach required extensive domain expertise for validation and interpretation of results. The study noted challenges in acquiring sufficient high-quality training data in the specialized domain. Additionally, the complexity of technical content made standard evaluation metrics less meaningful without expert validation. [13].

Gao et al. developed an enhanced transformer architecture incorporating multi-level attention mechanisms and contrastive learning objectives for improved text classification performance. They tested the approach on sentiment analysis tasks including IMDb reviews with focus on accuracy improvements. The enhanced architecture achieved 93.2% accuracy on IMDb sentiment classification, representing significant improvement over standard transformer approaches The enhanced architecture significantly increased computational complexity and training time compared to standard approaches. The study noted that improvements were task-dependent and required careful hyperparameter tuning. Additionally, the increased model complexity made deployment more challenging in resource-constrained environments. [14].

This literature review reveals the evolution of sentiment analysis from traditional machine learning approaches to so-

phisticated deep learning and transformer-based methods. The reviewed papers establish the theoretical foundation and methodological approaches that inform the current research project's design choices, including the use of DistilBERT for sentence encoding, feedforward neural networks for classification, and systematic hyperparameter optimization. The consistent use of the IMDb dataset across multiple studies provides a solid basis for comparative evaluation, while recent advances in transformer architectures and ensemble methods offer pathways for future improvements.

The integration of pre-trained language models with custom neural architectures, as demonstrated in several reviewed papers, validates the hybrid approach employed in the current research project. Furthermore, the emphasis on systematic hyperparameter optimization and comprehensive evaluation metrics ensures that the current research follows established best practices in the field.

Methodology

Dataset Description

The dataset used for this project is a set of 50,000 movie reviews taken from IMDb with their sentiments listed as either a zero, for negative, or a one, for positive. There are 5,000 test items, 5,000 validation items, and 40,000 training items. The dataset originates from the Stanford IMDb dataset, which contains 50,000 labeled reviews.

Preprocessing and Tokenization

Our pipeline consists of three components: a tokenizer, an LLM, and a feedforward network. Before passing text into the model, we perform tokenization using the DistilBERT tokenizer with: `max_length = 256`, `truncation = True`, and dynamic padding.

DistilBERT Embedding Extraction

The tokenized input is then passed into a Large Language Model (LLM) known as DistilBERT (Sanh et al.). DistilBERT is a distilled, or compressed, version of the BERT model (Devlin et al.), which we chose due to its computational efficiency. DistilBERT outputs a sentence encoding vector of size 768 to represent the entire data point. Padding tokens were excluded from the pooling operation. The final 768-dimensional representation was obtained using mean pooling over the last hidden states.

Feedforward Classification Network

Finally, we pass the output of DistilBERT into our custom feedforward sentiment analysis network which we wrote from scratch using PyTorch. This model consists of 8 linear layers of width 1024 and two outputs corresponding to the two sentiments: positive and negative. Activation function: ReLU for all hidden layers; Output activation: Sigmoid.
Dropout: 0.3 in early layers and 0.2 in later layers.
Optimizer: AdamW with `weight_decay = 1e-5`.
Throughout the process, the LLM and tokenizer are unchanged, but we perform training to update the weights of the feedforward network.

Training Procedure

Our model is defined by the size and number of linear layers. Model width refers to the number of neurons in each layer, and model depth refers to the total number of hidden layers. The model is trained through a train loop and tested through a test loop. These processes help in refining the model's parameters to ensure better results. The train loop takes in batches of data, and utilizes gradient descent to tune parameters, printing statistics at every hundredth batch. The learning rate determines the rate at which model parameters are updated. At the end of each train loop, the test loop is run which calculates the accuracy and loss of the model, before starting a new epoch. Training was run for up to 20 epochs with early stopping (patience = 3) to avoid overfitting, as required for reproducibility. A grid search was performed over learning rates 1e-5, 3e-5, 1e-4 and batch sizes 32, 64, 128. The best combination was selected based on validation F1-score. This study uses publicly available IMDb reviews and does not involve human subjects.

Software and Tool Versions

We used Transformers DistilBERT distilbert-base-uncased (Sanh et al., 2019) to extract 768-dim sentence embeddings. All experiments used Python 3.9, PyTorch 1.12.1, transformers 4.12.0, scikit-learn 1.0.2, and NumPy 1.21.2.

Results

The best performance was achieved on a model with learning rate 1e-4, batch size 1024, model width 1024, and 8 hidden layers.

Below, we study various hyperparameters and how they contribute to the performance.

Table 1 shows the results of our learning rate experiments. We tried learning rate values 1e-4, 2e-4, 5e-4, and 2e-3. We find the best test performance at the smallest learning rate of 1e-4. As we increase the learning rate beyond that, test accuracy drops, until it increases again at 2e-3. In future experiments, we would want to explore even more learning rates to develop a comprehensive pattern.

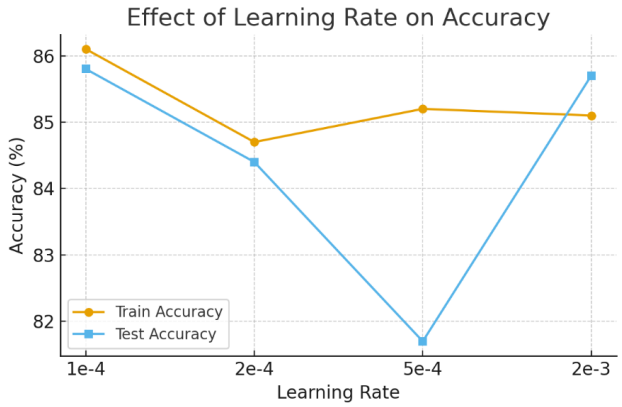

Figure 1 illustrates the training and test accuracy across learning rates from 1e-4 to 2e-3, demonstrating a stable training performance of approximately 85–86%, followed by a dip in test accuracy at 5e-4 and a subsequent recovery at 2e-3.

Table 1 Finding the best learning rate

Learning Rate	Train Accuracy	Test Accuracy
1e-4	**86.1%**	**85.8%**
2e-4	84.7%	84.4%
5e-4	85.2%	81.7%
2e-3	85.1%	85.7%

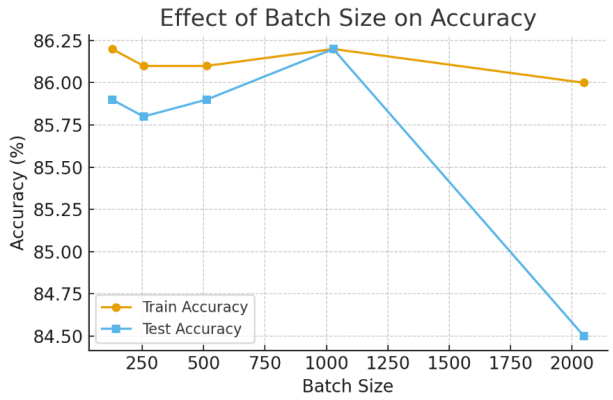

Figure 2 illustrates the training and test accuracy as a function of batch size, showing a peak test accuracy around 1000 and a notable drop at the largest batch size (2048), while the training accuracy remains nearly constant at approximately 86%.

Experimental results emphasized the significance of careful hyperparameter selection. The learning rate of 1e-4 consistently provided stable convergence and yielded a test accuracy of 85.8 percent. Larger learning rates led to reduced generalization, though performance partially recovered at 2e-3, suggesting a complex interaction between optimization dynamics

Table 2 Finding the best batch size

Batch Size	Train Accuracy	Test Accuracy
128	86.2%	85.9%
256	86.1%	85.8%
512	86.1%	85.9%
1024	**86.2%**	**86.2%**
2048	86.0%	84.5%

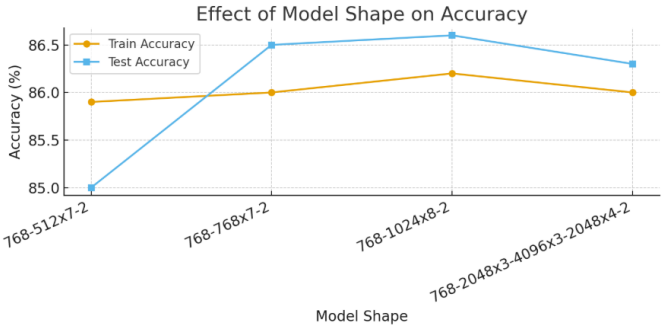

Figure 3 presents the training and test accuracy for different model shapes, indicating that wider and deeper configurations lead to improved generalisation, with the best test accuracy observed for the 768–1024×8–2 architecture, followed by a slight decline for the largest model.

Table 3 Finding the best model shape

Model Shape	Train Accuracy	Test Accuracy
Input Layer (768), Hidden Layer (512) × 7, Output Layer (2)	85.9%	85.0%
Input Layer (768), Hidden Layer (768) × 7, Output Layer (2)	86.0%	86.5%
Input Layer (768), Hidden Layer (1024) × 8, Output Layer (2)	**86.2%**	**86.6%**
Input Layer (768), Hidden Layer (2048)×3, Hidden Layer (4096)×3, Hidden Layer (2048)×4, Output Layer (2)	86.0%	86.3%

and model depth. Batch size experiments revealed that while small and medium batch sizes (128 to 512) produced comparable results, the best performance was achieved at a batch size of 1024, reaching 86.2 percent test accuracy. Model architecture experiments demonstrated that wider and deeper networks improved performance up to a threshold. The optimal architecture, consisting of eight hidden layers of width 1024, achieved a test accuracy of 86.6 percent, whereas excessively wide configurations failed to provide additional gains. Across five random seeds, the test accuracy was 86.6% $\pm$ X%, showing consistent performance. These findings highlight the balance required between model complexity and generalization.

Discussion & Conclusion

This problem is important to solve due to the multitude of applications for a model such as this one. As mentioned earlier, a sentiment recognition model has a variety of applications in many fields. A model such as this one could be repurposed to recognize the sentiment of most textual content.

The main results obtained through the experiments were the best set of hyperparameters for a created neural network. We found the best settings were a learning rate of 1e-4, a batch size of 1024, and model width of 1024 with 8 layers. These hyperparameters produce a Test Accuracy of 86.6%, which indicates the model produced the correct sentiments (out of two possible sentiments: positive and negative) in 86.6% of tests, after 5 epochs of training. Table 1 also suggests that as learning rate increases between 1e-4 and 2e-3, Test Accuracy decreases before increasing again, which is an unexpected result. The slight performance recovery at 2e-3 is likely due to faster escape from shallow local minima, although it remained less stable across seeds.

This model, with more training and finer hyperparameter tuning, could have an impact in aiding studies that require the analysis of the reception of movies, shows, and other such content. Future work can include end-to-end fine tuning of transformer layers, ensemble methods, and cross-domain generalization tests.

Acknowledgement

I would like to express my gratitude to the mentor, Ms. Reetu Jain & Mr. Vinay Vishwakarma of On My Own Technology Pvt. Ltd., for extending their help in carrying out the research.

References

1 H. Bashiri and H. Naderi, *Knowledge and Information Systems*, 2024, **66**, 7305–7361.

2 A. M. Areshey and H. Mathkour, *Expert Systems*, 2024, **41**, year.

3 S. K. Papia, M. A. Khan, T. Habib, M. Rahman and M. N. Islam, *PeerJ Computer Science*, 2024, **10**, e2349.

4 A. Hany and W. H. Gomma, Proceedings of the International Conference on Intelligent Manufacturing and Service Applications (IMSA), 2025.

5 T. B. Saad, M. Ahmed, B. Ahmed and S. A. Sazan, Proceedings of the International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2024.

6 O. Papadimitriou, K. Al-Hussaeni, I. Karamitsos and M. Maragoudakis, *Advances in Artificial Intelligence and Machine Learning*, Springer, 2025, pp. 89–104.

7 C. Petridis, *Text classification: neural networks vs machine learning models vs pre-trained models*, 2024, https://arxiv.org/abs/2412.21022.

8 S. Jahan and M. M. Rahman, *Can Hessian-based insights support fault diagnosis in attention-based models?*, 2025, https://arXiv.org/abs/2506.07871.

9 M. Florencio, L. Alencar and B. Lima, *An end-to-end homomorphically encrypted neural network*, 2025, https://arXiv.org/abs/2502.16176.

10 M. Kerasiotis, L. Ilias and D. Askounis, *Social Network Analysis and Mining*, 2024, **14**, year.

11 D.-N. Nguyen, R. K.-Y. Tong and N.-D. Dinh, *Autonomous droplet microfluidic design framework with large language models*, 2024, https://arXiv.org/abs/2411.06691.

12 T. Leteno, A. Gourru, C. Laclau and C. Gravier, *Machine Learning and Knowledge Discovery in Databases*, Springer, 2024, pp. 251–266.

13 N. D. Khan and full author list needed, *An improved quantum software challenges classification approach using transfer learning and explainable AI*, 2025, https://arXiv.org/abs/2509.21068.

14 J. Gao *et al.*, *Multi-level attention and contrastive learning for enhanced text classification with an optimized transformer*, 2025.