Evaluating TF-IDF With Logistic Regression and BERT Fine-Tuning for Movie Review Sentiment Analysis

Bowen Guo

School of Engineering, Architecture and Information Technology, University of Queensland, Brisbane, Australia

* Corresponding Author Email: bowen.guo1@student.uq.edu.au

Abstract. Understanding how people express opinions online is crucial for application such as public sentiment analysis social media monitoring and opinion-based decision making, yet sentiment analysis still faces difficulties due to the flexible, diverse, and often ambiguous use of language in user-generated content. In this study, we compare two distinct approaches: a conventional TF-IDF model combined with Logistic Regression, and a neural approach that fine-tunes BERT. Experiments were performed on the IMDb dataset containing 50,000 movie reviews, which is evenly split between positive and negative samples. We compared a simple TF-IDF model with logistic regression against a fine-tuned BERT using the AdamW optimiser and early stopping. The BERT model gave a slightly higher Macro-F1 (0.9146 ± 0.0041) than the baseline (0.9084). The improvement seems to come from its better handling of subtle or implied sentiment. Still, the traditional model remains surprisingly strong on a balanced dataset, suggesting that simpler models can still be valuable for sentiment analysis tasks.

Keywords: Sentiment Analysis; TF-IDF; Logistic Regression; BERT.

1. Introduction

These days, user-generated content shapes much of how people express themselves online. Every day, users share opinions through comments and reviews on social media and other platforms. As the rapid increase in this content, it has led to sentiment analysis becoming an essential tool for understanding public opinion, guiding business, and monitoring societal well-being [1]. A significant example is movie reviews where sentiment analysis not only influences consumer choices but also help studios and researchers understand trends in the film industry.

Research on opinion analysis in online content has grown steadily over the years. Early studies focus on rule-based approaches [2], but soon moving through traditional machine learning models [3], More recently, pre-trained language models like BERT [4] have brought another leap forward. Even though these approaches are based on different methods, and each showing its own advantages and drawbacks.

To some extent, it is important to highlight how each method performs differently in sentiment analysis. Rule-based systems rely heavily on pre-defined sentiment lexicons and expert knowledge [2], so their performance often drops when the vocabulary changes over time. Traditional machine learning methods, including logistic regression and support vector machines, address some of these limitations but remain highly dependent on manual feature engineering [3]. As a result, their performance declines when applied to large-scale and diverse datasets. In contrast, pre-trained language models such as BERT generate contextualized embeddings learned from massive corpora, enabling substantial improvements in sentiment classification without extensive feature design [5]. With the advent of pre-trained language models, particularly BERT, sentiment analysis has entered a new stage [5]. BERT leverages bidirectional contextual information and transfer learning, allowing the model to reach competitive performance across numerous natural languages processing tasks, including sentiment classification.

In this paper we employed the IMDb movie review dataset, a widely used resource containing fifty thousand movie review samples equally divided into positive and negative categories [6]. Building

on this resource, to assess effectiveness in sentiment classification, we contrast BERT fine-tuning against a conventional machine learning baseline.

2. Methods

2.1. TF-IDF and Logistic Regression

The Term Frequency-Inverse Document Frequency (TF-IDF) approach is a widely adopted technique for converting text into numerical representations suitable for computational analysis. It quantifies how important a word is within a particular document compared with its frequency in a collection of documents. Specifically, TF measures how frequently a word occurs in a given text, while IDF penalizes words that appear in many documents by taking the logarithm of the ratio between the total number of documents and the number containing that word. Through this weighting scheme, common words receive smaller scores and unique words are assigned higher significance [7].

The TF-IDF score for a term t in a document d is computed as [7]:

$$\text{TF-IDF}(t,d) = \text{tf}(t,d) \cdot \log\left(\frac{N}{\text{df}(t)}\right) \tag{1}$$

In this formulation, $\text{tf}(t,d)$ represents the occurrence count of t in document d; N denotes the total number of documents corpus; $\text{df}(t)$ denotes the number of documents containing term t[7]. Together, these components capture both the local and global significance of each word in the dataset.

Due to its ability to highlight informative terms, The TF-IDF method has become a fundamental feature extraction technique in numerous natural languages processing applications, including information retrieval and various forms of text classification. It is particularly effective when the discriminative power of individual words or n-grams can be directly exploited, such as in sentiment classification of short texts.

After vectorizing the documents using TF-IDF, this paper applies a Logistic Regression model for binary classification. In Logistic Regression, the likelihood of an instance belonging to a particular class is estimated through a sigmoid transformation of a linear combination of its features:

$$\hat{y} = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{2}$$

In this model, $\mathbf{x} \in R^n$ represents the TF-IDF features extracted from each document. The parameter vector $\mathbf{w} \in R^n$ contains the learned coefficients that determine the contribution of each feature, while the scalar b acts as a bias term adjusting the model's output. The predicted probability $\hat{y} \in (0,1)$ reflects the likelihood that a given sample belongs to the positive category, and $\sigma(\cdot)$ represents the logistic sigmoid function.

In this paper for sentiment analysis task, each movie review is transformed into a high-dimensional TF-IDF vector based on unigrams and bigrams. These vectors are then fed into an L2-regularized Logistic Regression model, where L2-regularization helps prevent overfitting by discouraging excessively large weight values and thereby improves the model's generalization ability. The model trained using the liblinear solver, this optimization method is widely applied in Logistic Regression and performs efficiently in high-dimensional text classification tasks. This traditional pipeline is used as a strong baseline when comparing with neural models such as BERT. It is particularly efficient and easy to interpretable, showing strong performance on datasets like IMDb, where sentiment is often lexically explicit.

2.2. BERT-based Sentiment Classifier

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep language model which uses a multi-layer bidirectional Transformer architecture to generate contextualized word representations [4]. Unlike early embedding methods such as Word2Vec, BERT represents each

word in relation to its context, allowing it to capture more nuanced semantics across various tasks in natural language processing.

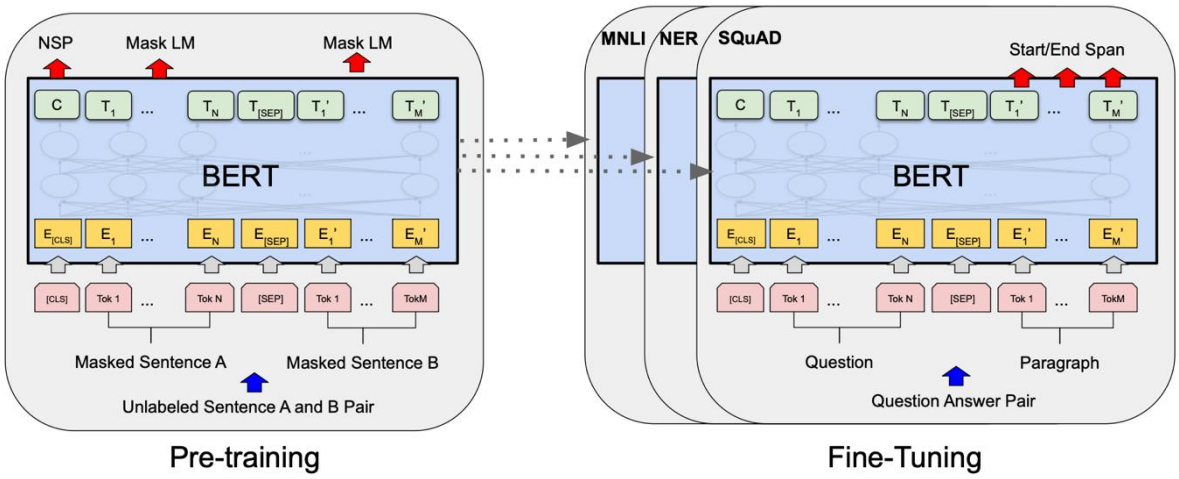

Fig. 1 Architecture of BERT in pre-training (left) and fine-tuning for text classification (right). In this configuration, the [CLS] token is used for binary sentiment prediction [4].

Figure 1 illustrates the overall framework of BERT, including its pre-training and fine-tuning stages. In the pre-training phase (shown on the left), the model processes pairs of sentences—denoted as Sentence A and Sentence B—augmented with the special tokens [CLS] and [SEP]. It learns through two self-supervised objectives: Masked Language Modeling (MLM), which reconstructs words that have been randomly hidden within the text, and Next Sentence Prediction (NSP), which trains the model to assess whether Sentence B naturally follows Sentence A based on the contextual embedding of [CLS][4]. In the fine-tuning phase (right side), BERT adapts to specific downstream applications such as question answering, where the model receives a question–context pair as input and predicts the token indices corresponding to the beginning and end of the correct answer within the passage [4].

For sentiment analysis, BERT takes the input sentence and first tokenizes it using the WordPiece tokenizer. The sequence begins with a special token [CLS], whose final hidden representation acts as a holistic embedding of the input and is subsequently employed for classification purposes [4]. Each token is then mapped to an embedding that integrates three components: a token embedding that encodes the meaning of the word, a segment embedding that distinguishes different sentences when the input consists of a pair, and a position embedding that provides information about the token's order in the sequence. The model then processes these embeddings through multiple layers of Transformer encoders with self-attention, allowing the model to learn contextual dependencies among tokens [8].

In this experiment, we fine-tuned the bert-base-uncased model on the IMDb dataset. The model is optimized using AdamW [9] with early stopping based on validation F1 score. Fine-tuned approach has demonstrated competitive performance in sentiment analysis tasks, particularly in identifying subtle or implicit emotional expressions

3. Experiments

3.1. Dataset

All experiments are carried out using the IMDb movie review dataset, having 50,000 balanced samples of positive and negative sentiments [6]. The dataset is split into two equal halves: 25,000 instances are used for model training and the other 25,000 instance are used for testing. In addition, 10% of the training portion is further set aside as a validation subset to support hyperparameter optimization and early stopping.

3.2. Preprocessing

For the traditional machine learning baseline, reviews are lowercased, tokenized, and vectorized using TF-IDF with uni-grams and bi-grams [10]. We apply logarithmic term frequency scaling, set $\text{min\_df} = 2$ to remove rare tokens and cap the vocabulary size at 100k features to control sparsity.

For BERT-based models, the WordPiece tokenizer from bert-base-uncased is applied. Reviews are truncated or padded to a maximum length of 256 tokens, and the corresponding input_ids and attention_mask are generated [4][11].

3.3. Models

We compare two categories of models:

- TF-IDF + Logistic Regression: A traditional machine learning baseline. We train L2-regularized logistic regression with liblinear solver [10].
- BERT Fine-Tuning: The [CLS] token embedding is passed to a linear classification layer with softmax activation [4].

3.4. Training Setup

- For the TF-IDF baseline

The regularization strength C is tuned on the validation set via a small grid search [0.25, 0.5, 1.0, 2.0, 4.0].

The final model is retrained on the combined training and validation sets with the best C before reporting performance on the test dataset.

- For BERT Fine-Tuning

In fine-tuning, the model was optimize using Adam [9] with a weight decay of 0.01. The learning rates was set to 2×10^{-5} for BERT encoder, and 1×10^{-3} for classifier head [12]. Training was performed with a batch size of 16 and evaluation with a batch size of 32, for 5-6 epochs with early stopping based on validation F1 score. A dropout rate of 0.1 was applied to mitigate overfitting, and mixed-precision training fp16 was employed to improve computational efficiency [12].

3.5. Evaluation Metrics

Performance is evaluated using Accuracy, Precision, Recall, and Macro-F1 score [13]. Each experiment is repeated three times with different random seeds, and the mean and standard deviation are reported.

4. Experiments

4.1. TF-IDF Logistic Regression Baseline

For the traditional machine learning baseline, we trained an L2-regularized logistic regression model on TF-IDF features [10],[14]. Table 1 summarizes the validation performance of the TF-IDF + Logistic Regression model under different values of the regularization parameter C [15]. A consistent improvement in Macro-F1 was observed as C increased, with the best result obtained at C = 4.0.

Table 1. Validation performance of TF-IDF + Logistic Regression with different C values

c	Accuracy	Precison	Recall	Macro-F1
0.25	0.8808	0.8819	0.8808	0.8807
0.5	0.8944	0.8952	0.8944	0.8943
1.0	0.9024	0.9030	0.9024	0.9024
2.0	0.9076	0.9081	0.9076	0.9076
4.0	0.9148	0.9153	0.9148	0.9148

With the regularization parameter set to $c = 4.0$, the model achieved 0.9084 accuracy and Macro-F1 on test set, which we adopt as the baseline for subsequent comparisons.

Figure 2 illustrates the effect of varying the regularization parameter c on the performance of the TF-IDF + Logistic Regression model using the validation set [15]. As c increases from 0.25 to 4.0, all four-evaluation metrics—Accuracy, Precision, Recall, and Macro-F1—show a consistent upward trend. The improvement is most notable when moving from the default setting $\mathrm{C} = 1.0$ to the best configuration $\mathrm{C} = 4.0$, where the model achieves its highest Macro-F1. This trend confirms that appropriate tuning of the regularization parameter can yield measurable gains across multiple metrics, particularly in overall predictive balance as captured by Macro-F1.

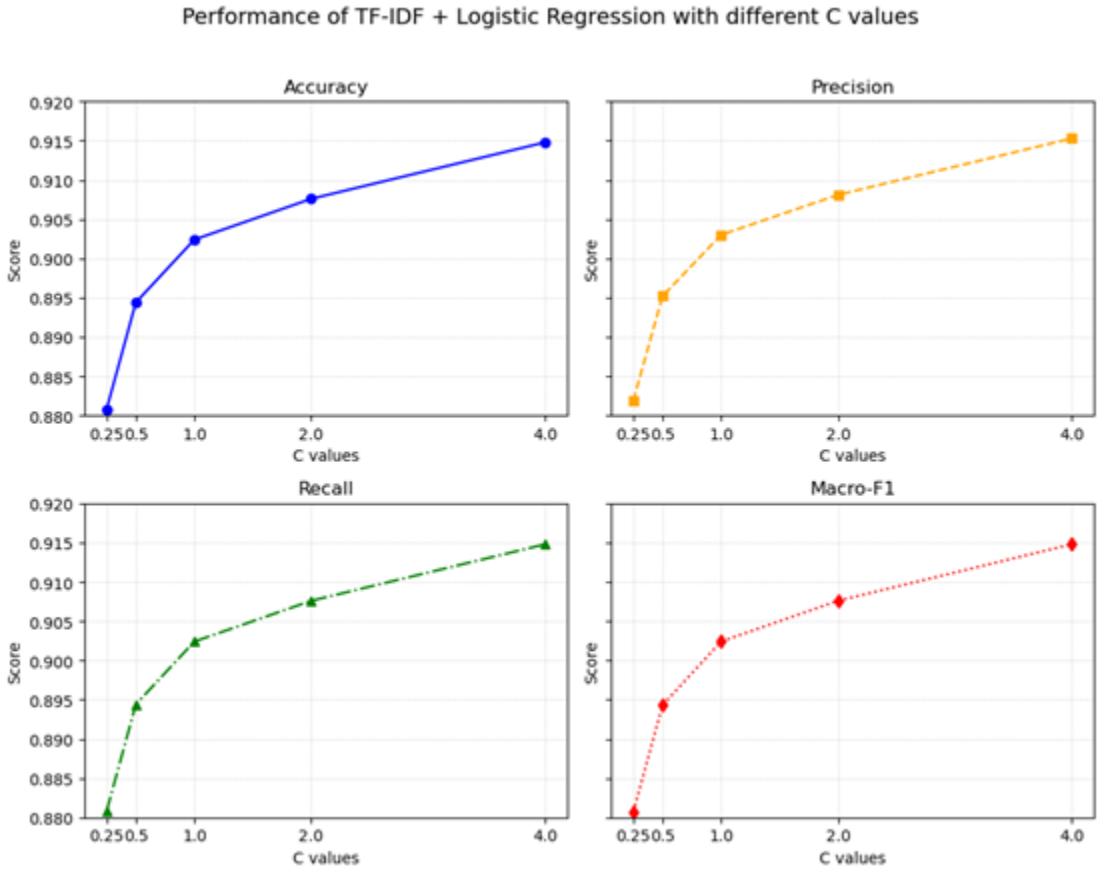

Fig. 2 Different C values performance in TF-IDF +Logistic Regression

4.2. BERT Fine-tuning

For the neural baseline, the bert-base-uncased model [4] was adapted to the IMDb dataset, where each input sequence was truncated to the maximum length of 256 tokens. Fine-tuning was performed using the AdamW optimizer [9] with discriminative learning rates (2×10^{-5} for the encoder and 1×10^{-3} for the classification head), a weight decay of 0.01, and a dropout probability of 0.1. Mixed-precision training was enabled when supported, and early stopping was applied based on the validation Macro-F1 score. To enhance reliability, the experiments were conducted three times using different random seeds (42, 2025, 7).

Table 2 reports the averaged test set performance of BERT Fine-Tuning, where each metric is presented as mean ± standard deviation across the three runs.

Table 2. Test set performance of BERT Fine-Tuning on IMDb

Model	Accuracy	Percision	Recall	Macro-F1
BERT Fine-Tuning	0.9147±0.0040	0.9161±0.0033	0.9147±0.0040	0.9146±0.0041

4.3. BERT Fine-tuning

We compare our sentiment classification results with both traditional and modern approaches reported in the literature. The TF-IDF + Logistic Regression baseline achieved a Macro-F1 of 0.9084 on the IMDb dataset, which is consistent with earlier findings that bag-of-words models coupled with strong linear classifiers remain highly competitive for text classification [10]. On the other hand, fine-tuning BERT produced a Macro-F1 of 0.9146 ± 0.0041, demonstrating a consistent improvement over the baseline and aligning with previous reports on the effectiveness of pre-trained transformers for sentiment analysis [5].

From the results we can conclude that contextual embeddings derived from BERT yield robust and stable performance improvements, particularly by capturing semantic and syntactic nuances that n-gram features cannot [16]. However, classical feature-based methods still deliver impressive accuracy [17], especially on balanced datasets such as IMDb where the dataset contains an equal proportion of positive and negative reviews and sentiment cues are often lexically explicit [14]. In this scenario, the performance gap between TF-IDF and BERT is relatively narrow, which highlights the continuing strength of traditional baselines [10].

These results indicate that while models like BERT, which are pre-trained on large corpora, have led to notable gains in sentiment classification [4], simpler approaches remain valuable due to their efficiency and surprisingly competitive accuracy. This balance suggests that the choice of method should consider not only predictive performance but also computational resources and application context [9],[14].

5. Conclusion

In this work, we investigated sentiment analysis on the IMDb dataset by comparing a traditional TF-IDF + Logistic Regression baseline with BERT fine-tuning. Experiments showed that the baseline achieved a Macro-F1 of 0.9084, while BERT fine-tuning further improved performance to 0.9146 ± 0.0041. These results confirm the robustness of contextualized embeddings while also highlighting the competitiveness of traditional models. However, the performance gap between TF-IDF and BERT remains narrow, partly because sentiment cues in IMDb reviews are often lexically explicit, allowing bag-of-words methods to achieve strong accuracy. In the future, we intend to broaden our study with longer input sequences, advanced optimization strategies, and evaluations on more diverse datasets where sentiment is expressed more implicitly, in order to better highlight the advantages of pre-trained language models.

References

[1] K. Alahmadi, S. Alharbi, J. Chen, and X. Wang, “Generalizing sentiment analysis: a review of progress, challenges, and emerging directions,” *Soc. Netw. Anal. Min.*, vol. 15, no. 1, p. 45, Apr. 2025, doi: 10.1007/s13278-025-01461-8.

[2] K. Barik and S. Misra, “Analysis of customer reviews with an improved VADER lexicon classifier,” *J. Big Data*, vol. 11, no. 1, p. 10, Jan. 2024, doi: 10.1186/s40537-023-00861-x.

[3] A. Hussain and E. Cambria, “Semi-supervised learning for big social data analysis,” *Neurocomputing*, vol. 275, pp. 1662–1673, Jan. 2018, doi: 10.1016/j.neucom.2017.10.010.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. in Proc. NAACL-HLT (North American Chapter of the Association for Computational Linguistics: Human Language Technologies), Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[5] S. Alaparthi and M. Mishra, “BERT: a sentiment analysis odyssey,” *J. Mark. Anal.*, vol. 9, no. 2, pp. 118–126, June 2021, doi: 10.1057/s41270-021-00109-8.

[6] “IMDB Dataset of 50K Movie Reviews.” Accessed: Sept. 15, 2025. [Online]. Available: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[7] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.

[8] A. Vaswani *et al.*, “Attention is All you Need”. in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[9] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” Jan. 04, 2019, *arXiv*: arXiv:1711.05101. doi: 10.48550/arXiv.1711.05101.

[10] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” May 28, 2002, *arXiv*: arXiv:cs/0205070. doi: 10.48550/arXiv.cs/0205070.

[11] X. Lin, M. Kelly, J.-R. Park, K. Seki, and W. Ke, “Author’s Name: Sonia Manalo Pascua Submission Date: 06/19/2025”.

[12] U. Onyekpe, V. Palade, and M. A. Wani, *Recent Advances in Deep Learning Applications: New Techniques and Practical Examples*, 1st ed. Boca Raton: Chapman and Hall/CRC, 2025. doi: 10.1201/9781003570882.

[13] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, July 2009, doi: 10.1016/j.ipm.2009.03.002.

[14] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis”.

[15] P. K. Kumaresan, R. Ponnusamy, R. Priyadharshini, P. Buitelaar, and B. R. Chakravarthi, “Homophobia and transphobia detection for low-resourced languages in social media comments,” Nat. Lang. Process. J., vol. 5, p. 100041, 2023, doi: 10.1016/j.nlp.2023.100041.

[16] Z. Li, Y. Zou, C. Zhang, Q. Zhang, and Z. Wei, “Learning Implicit Sentiment in Aspect-based Sentiment Analysis with Supervised Contrastive Pre-Training,” Nov. 03, 2021, *arXiv*: arXiv:2111.02194. doi: 10.48550/arXiv.2111.02194.

[17] A. M. Van Der Veen and E. Bleich, “The advantages of lexicon-based sentiment analysis in an age of machine learning,” *PLOS ONE*, vol. 20, no. 1, p. e0313092, Jan. 2025, doi: 10.1371/journal.pone.0313092.