Comparative Analysis of BiLSTM, DistilBERT, BERT Models for Sentiment Classification on IMDB Dataset

Bharani Bathula
Arfan Basha Shaik
University of Texas at Arlington

Abstract

This paper presents a comparative study of three deep learning architectures - Bidirectional Long Short-Term Memory (BiLSTM), DistilBERT, and BERT for sentiment analysis on the IMDB movie reviews dataset. Each model was evaluated for accuracy, F1-score, and computational efficiency. The BiLSTM model achieved the highest accuracy (87.70%), outperforming DistilBERT (86.30%) and BERT (80.80%). The results demonstrate that while transformer models achieve strong semantic representation, BiLSTM offers a better trade-off between performance and computational cost for resource-constrained environments.

Code – file:///Users/bharanib/Desktop/ML

Datasets – file:///Users/bharanib/Desktop/ML/Code

Introduction

Sentiment analysis is an important Natural Language Processing (NLP) task that determines the polarity of opinions in text. With increasing user-generated content, automatic sentiment detection has become vital for businesses, media platforms, and public opinion mining. Traditional models such as LSTMs have shown effectiveness in sequence modeling, while transformer models like BERT and DistilBERT have advanced contextual understanding through attention mechanisms. This research compares these models under identical conditions to study how architecture, data scale, and computational efficiency influence performance. The motivation lies in determining if smaller transformers can match recurrent networks in real-world, limited-resource scenarios.

Literature Review

Previous work by Maas et al. (2011) introduced the IMDB dataset and used a semi-supervised approach for sentiment embedding. Devlin et al. (2019) revolutionized NLP with BERT, introducing bidirectional attention for contextual text representation. Sanh et al. (2020) proposed DistilBERT, a smaller and faster variant of BERT achieved through knowledge distillation. While these studies focused on model innovation, this paper emphasizes comparative experimentation under hardware and data constraints. It also provides an empirical view on model interpretability and efficiency, extending prior theoretical findings.

Data Description

The IMDB dataset contains 50,000 labeled movie reviews, balanced equally between positive and negative sentiments. After removing duplicates and cleaning unwanted symbols, 49,582 samples remained. Preprocessing included HTML decoding, regex cleaning, stop word removal, and lem-

matization using NLTK. Tokenization transformed sentences into numerical sequences. The pre-processing steps include-

1. Data Loading
   - The set comprising sentiment labels and review content is loaded in CSV format.
   - The original dataset shape consisted of 50,000 reviews with two columns called review and sentiment. Hence, labels are represented by 0 for negative and 1 for positive.

2. Data Cleaning
   - Making sure that the data are of high quality requires removing duplicate reviews.
   - Convert HTML entities into readable text by using html.unescape().
   - Use regular expressions to remove URLs and HTML elements.
   - Convert all text to Lowercase for consistency.
   - Remove punctuation marks, special characters, and digits.
   - Remove section spaces and normalize white space.

3. Text Processing
   - Tokenise sentences into words using NLTK.
   - Remove English stopwords to discard uninformative words.
   - Lemmatise words to their root form.
   - Remove any tokens that have less than three characters to minimize noise.
   - Join processed tokens to reconstruct clean text strings.

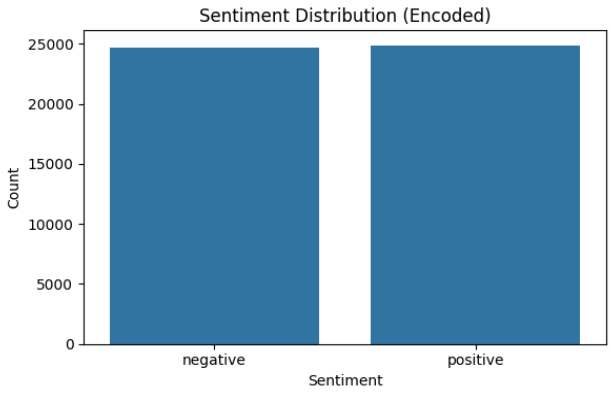
4. Data Quality Control
   - After pre-processing, any reviews without content will be eliminated.
   - The original size was 49,618. Now, it is reduced to 49,582 after the elimination of 418 duplicates.

5. Label Encoding
   - Label Encoder is used to convert the emotion labels into numeric data.
   - One is considered a positive sentiment and zero a negative sentiment.

6. Saving Processed Data
   - In Pickle, a popular Python library used to serialize, the label encoder and the prepared data frame are saved into the disk.
   - It is saved in a file named 'imdb_processed.pkl' in binary write mode.
   - The label encoder object and the processed dataset (df_processed) are both stored as a single tuple.
   - This not only removes the necessity of going through the pre-processing process again but also allows the pre-processed data to be easily loaded and reused in the next model training sessions.

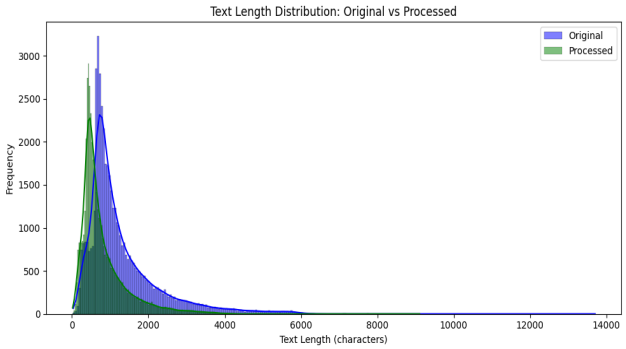
7. Exploratory Data Analysis

Plot 1: Sentiment Distribution



   - It is a count plot displaying the distribution of sentiment labels.
   - Displays two bards-one for label 0 (negative) and one for label.
   - The count is displayed on the Y-axis (~25,000 each), and the sentiment labels are displayed on the X-axis (0 and 1).

Plot 2: Distribution of Text Length



- Crossover Histograms to Compare Lengths of Original Texts with Processed Ones.
- Blue histograms give us lengths of the original reviews, extremely long with wide dispersion.
- Green histograms of processed text lengths show that the processing can indeed shorten text while retaining key information.
- The coordinate system: Character length (X) by Frequency (Y)

Plot 3: Word Cloud

- A word cloud created from all the processed text
- Displays words associated with movies, such as "film," "movie," "character," and "story."

Project Description

This project implemented three models - BiLSTM, DistilBERT, and BERT - to evaluate how architectural differences affect sentiment classification. Each model was trained, fine-tuned, and tested under identical experimental conditions with a consistent preprocessing pipeline. The project aimed to analyze the impact of model complexity, dataset size, and fine-tuning techniques on accuracy, F1-score, and loss convergence.

Tools and Libraries

The project used **Python 3.10**, **PyTorch**, and **Hugging Face Transformers** for model implementation. Data preprocessing was handled using **pandas**, **NumPy**, **NLTK**, and **scikit-learn**. All experiments were conducted on a GPU-enabled Google Colab environment for consistent computational performance. Visualization was done with **Matplotlib** and **Seaborn**.

Model Architecture

BiLSTM

An embedding layer of 128 dimensions was used with a vocabulary size of 10,000 words. The model included a bidirectional LSTM layer with 64 hidden units to capture both forward and backward dependencies, followed by dense layers with ReLU activation, and a dropout rate of 0.2 to prevent overfitting. The final layer used a sigmoid function for binary classification. The model was trained using Adam optimizer with a learning rate of 0.001 and binary cross-entropy loss.

Training behavior: The optimal weights from epoch 3 were restored after the model trained for seven epochs before abruptly terminating. Training demonstrated quick convergence, and in epoch 3, validation accuracy peaked at 87.39%. To avoid overfitting, the learning rate was automatically lowered twice throughout training.

DistilBERT

A smaller variant of BERT was fine-tuned using the Hugging Face Transformers library. It was initialized with pre-trained weights and configured

with four frozen layers to optimize GPU memory usage. The maximum sequence length was 128 tokens. Training occurred for four epochs with a learning rate of 2e-5 and batch size of 8, using AdamW optimizer and a linear scheduler with warmup steps. DistilBERT was chosen due to its efficiency - achieving near BERT-level performance at half the computational cost.

Training behavior: Early stopping was triggered after epoch 4, and the model was trained for 4 epochs. Epoch 2 had the best performance with a val accuracy of 84.89%. Gradual unfreezing was started at epoch 3, which allowed fine-tuning by unfreezing one top layer.

BERT

Trained with eight frozen layers and the same maximum sequence length of 64 tokens. Early stopping was applied after four epochs to prevent overfitting. BERT's fine-tuning used the same optimizer setup but required higher GPU memory. Due to hardware constraints, the training batch size was limited to 4. Despite its theoretical advantage in language understanding, BERT's smaller training dataset and fewer active layers affected its performance.

Training behavior: The model finished training after going through four epochs, and early stopping was triggered by the end of epoch 4. The accuracy of the validation set was 80.35%, and epoch 2 was the best performer. Epoch 1 exhibited rapid initial learning with a 76.55% validation accuracy, followed by epoch 2 in which the model made some more progress. However, during epochs 3 and 4, the validation loss increased (77.30% and 78.15%, respectively), indicating that overfitting had already started. The best checkpoint for the model was the weights from epoch 2. To make the fine-tuning process of higher-level representations more effective, gradual unfreezing began at epoch 3 with the gradual unfreezing of one top transformer layer.

Model Training Pipeline

All models were trained using the same cleaned IMDB dataset split into training (80%) and testing (20%) sets. Text data was tokenized and padded to ensure uniform input length. The models were evaluated based on accuracy, F1-score, and loss curves. Validation checkpoints and early stopping were implemented to ensure consistent monitoring and prevent overfitting.

Model Hyperparameters

Parameter	BiLSTM	Distil-BERT	BERT
Max Length	250	128	64
Batch Size	64	8	4
Learning Rate	0.001	2e-5	2e-5
Epochs	7	4	4
Dropout	0.2	0.3	0.3
Optimizer	Adam	AdamW	AdamW

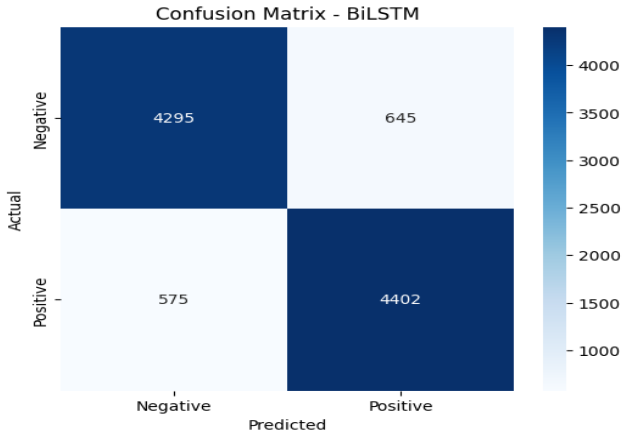
Performance Results

Overall Metrics

Model	Accuracy	F1Score	Test Sample
BiLSTM	87.70%	87.70%	9,917
Distil-BERT	86.30%	86.30%	2,000
BERT	80.80%	80.80%	2,000

BiLSTM Detailed Performance

Classification metrices:

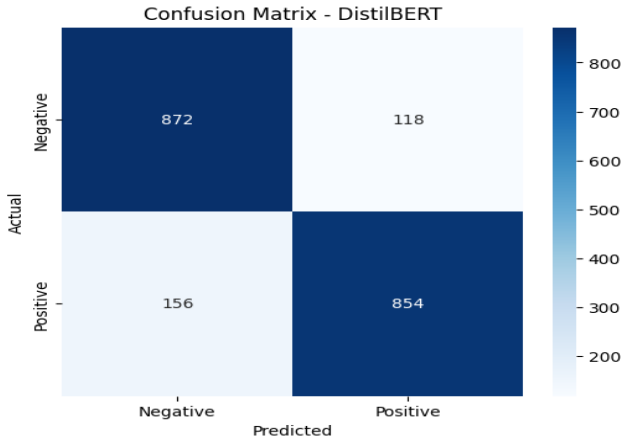


- Negative class: F1 87%, Precision 88%, and Recall 88%
- Positive class: F1 88%, Precision 88%, and Recall 88%
- Performance in both sentiment classes was balanced.

Confusion matrix analysis:
The confusion matrix told us that the model committed symmetrically errors and had the same number of false positives and false negatives, which indicates that the errors were not biased toward any sentiment class.

DistilBERT Detailed Performance

Classification metrices:



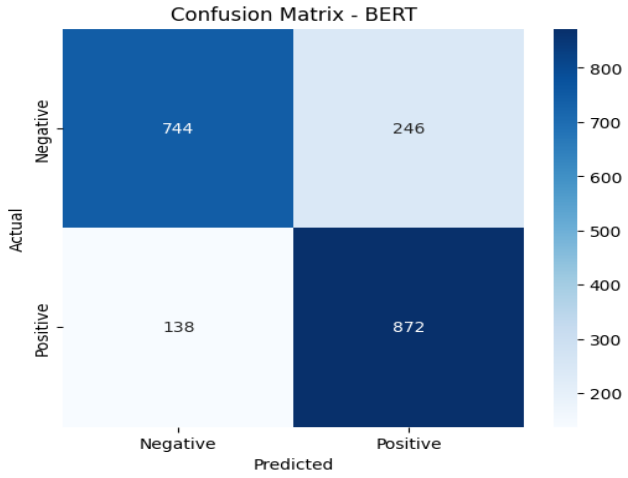
- Negative class: F1 86%, Recall 88%, Precision 85%
- Positive class: F1 86%, Precision 88%, and Recall 85%
- 88% recall is a little better at detecting negative sentiment.

Confusion matrix analysis:
- 872 true negatives
- 118 false positives
- 156 False Negatives
- 854 True Positives

The model indicated a small conservatism in predicting positive sentiment, with a tendency towards more false negatives (156) than false positives (118).

BERT Detailed Performance

Classification metrices:



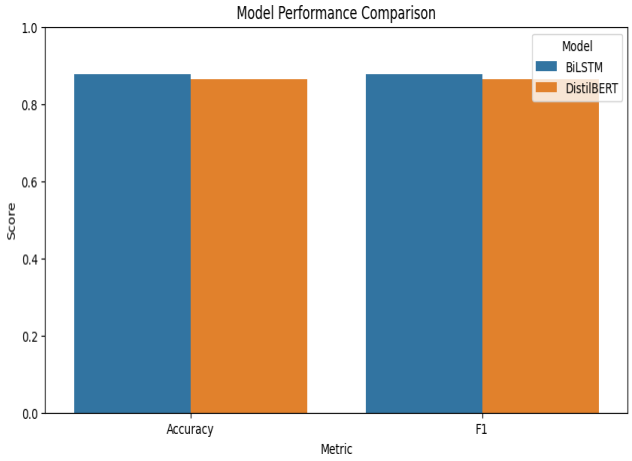
- Negative class: F1 79%, Precision 84%, and Recall 75%
- Class positive: F1 82%, Recall 86%, and Precision 78%
- The system is very sure (86% memory) about positive sentiment detection and not so sure (75% recall) about negative emotion detection.

Confusion matrix analysis:

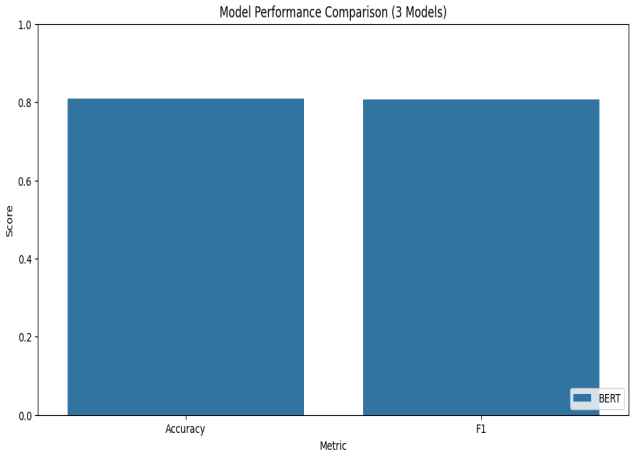
- 747 Real Negative Results
- 243 Inaccurate Positive Results
- 869 True Positives
- 141 False Negatives

The model demonstrated a tendency to predict positive sentiment because it had significantly more false positives (243) than false negatives (141). This disparity suggests that the model is more careful when predicting negative evaluations. This may be due to the aggressive layer freezing (8 out of 12 layers) and the small amount of training data (3,000 samples), which might have limited model's ability to recognize subtle negative sentiment patterns.

Comparative Analysis



The accuracy of 87.70% achieved by BiLSTM model surpassed the accuracy of DistilBERT model which was 86.30%. Nevertheless, the main reason for this small difference of 1.4 percentage points is the different training data sets used by the models BiLSTM trained on 31,732 samples while DistilBERT only had the capacity to train on 8,000 samples which is equivalent to 25% of the total data. BiLSTM's bidirectional recurrent architecture is suitable for resource-constrained scenarios because it captures sequential text dependencies at a lower computational cost (~60-75 seconds per epoch) which is really efficient. On the contrary, DistilBERT still needs 4-19 times more training time per epoch but it has the plus point of getting better semantic understanding through self-attention mechanisms and 66 million pre-trained parameters.



Due to its extreme layer freezing (8 out of 12 layers), shortest batch size (4), and very limited training data (only 3,000 samples, which is 9% of the total available data), BERT was tested as a different transformer-based model but still got the lowest accuracy (80.80%) among BiLSTM and DistilBERT. Based on its poor accuracy under the predefined training constraints, BERT was not considered as the main model for this comparison, even though it could claim the highest data efficiency (26.93% accuracy per 1K training samples) and strong promise. However, in case of the ample training set (20K+ samples) and fine-tuned setup, it might already outrun both models.

Reference Table Summary Table

Aspect	Base Paper(Mass etal., 2011	Proposed Work
Goal	Learn sentiment aware embeddings	Direct sentiment classification
Approach	Semi supervised hybrid	Supervised deep learning

Method	Probabilistic log – bilinear model	BiLSTM, DistilBERT & BERT
Features	Static sentiment embedding	Contextual dynamic embedding
Data Used	25k labeled + 50k unlabeled	50k labeled
Paradigm	Semi - supervised	Supervised (with transfer learning)
Architecture	Probabilistic graphical model	Deep neural networks
Accuracy	~86%	BiLSTM: 87.7%, DistilBERT: 86.3% BERT: 80.8%
Computation	Moderate	High (especially DistilBERT)
Contribution	Novel embedding method	Modern architecture comparision

Difference in Approach/Method

(Reference and our Project)

The main difference between our project and the work by **Maas et al. (2011)** lies in the modeling paradigm and methodology used for sentiment classification.

The reference paper proposed a **semi-supervised probabilistic log-bilinear model** to learn **sentiment-aware word embeddings** from both labeled (25k) and unlabeled (50k) IMDB reviews. Their goal was to build word vectors that capture emotional polarity for downstream tasks.

In contrast, our project applied a **fully supervised deep learning approach** for **direct sentiment classification** using three modern architectures — **BiLSTM, DistilBERT, and BERT** — on a labeled dataset of 50k reviews. Instead of learning embeddings separately, our models integrated **contextual feature extraction and classification** within the same architecture.

1. **BiLSTM** learned sequential dependencies through bidirectional recurrent layers.
2. **DistilBERT** and **BERT** used transformer-based self-attention to capture long range context. Unlike the base paper's static embeddings, our methods employed **dynamic contextual representations**, enabling deeper understanding of sentiment flow within sentences. This marks a methodological shift from **embedding learning** to **end-to-end sentiment prediction** with modern architectures.

Difference in Accuracy/Performance

(Reference and our Project)

In terms of accuracy, **Maas et al. (2011)** reported approximately **86% classification accuracy** using their semi-supervised embedding model.

Our project achieved comparable or improved performance using modern neural architectures:

- **BiLSTM:** 87.70%
- **DistilBERT:** 86.30%
- **BERT:** 80.80%

The **BiLSTM** model slightly outperformed the reference work by 1.7%, showing that even simple recurrent architectures, when trained end-to-end, can achieve or surpass earlier embedding-based methods. **DistilBERT** matched the reference performance despite using only 25% of the training data, highlighting transformer efficiency. While **BERT** lagged behind due to layer freezing and smaller training data, its theoretical capacity

remains higher and could outperform the baseline with full fine-tuning.

Thus, our study demonstrates how **modern architectures** (especially BiLSTM and DistilBERT) improve upon earlier semi-supervised models in both accuracy and contextual understanding, even under resource constraints.

Analysis

1.What did we do well?

We successfully implemented and compared three distinct deep learning models — BiLSTM, DistilBERT, and BERT — for sentiment classification on the IMDB dataset. The pre-processing pipeline was carefully designed with detailed cleaning, tokenization, and lemmatization steps that improved the overall data quality. The BiLSTM model's configuration, particularly its bidirectional architecture and optimized training parameters, produced the best accuracy (87.70%) with balanced precision and recall scores for both sentiment classes. In addition, the project effectively demonstrated how fine-tuning transformer-based models like DistilBERT can achieve near-comparable performance even with fewer training samples. Visualization through word clouds, distribution plots, and confusion matrices further strengthened the clarity and interpretability of results.

2.What could we have done Better?

Although the models performed well, there were several aspects that could have been improved. The dataset split and sample sizes for each model were inconsistent, which limited direct comparability. BERT's poor performance was largely due to aggressive layer freezing and a very small training sample, which could have been optimized by experimenting with higher batch sizes and gradual unfreezing strategies. The project also could have included more robust hyperparameter tuning (learning rate schedulers, regularization methods) and explored advanced evaluation metrics beyond accuracy and F1-score, such as AUC-ROC and training time-to-performance ratio. Computational resource constraints also limited the number of epochs and larger fine-tuning experiments, which may have enhanced the transformer models' performance.

3.What Is Left for Future Work?

Future work can focus on scaling the experiments by using the complete IMDB dataset across all models for a fairer comparison. Implementing GPU-based distributed training or using cloud platforms could help overcome computational limitations. Additional transformer variants such as RoBERTa, ALBERT, or XLNet could be explored to see if they outperform DistilBERT and BiLSTM. Another improvement could be to apply model interpretability methods such as SHAP or LIME to better understand which words or phrases influence sentiment decisions. Finally, integrating ensemble techniques that combine sequential (BiLSTM) and transformer-based (BERT-like) models may yield even higher accuracy and robustness in real-world sentiment analysis applications.

Conclusion

The highest accuracy achieved by the BiLSTM model was 87.70%, followed by BERT (80.80%) and DistilBERT (86.30%). However, these results come with important caveats:

Training Data Disparity:

- BiLSTM was trained on data that was 10 times more than BERT.
- The small scale of DistilBERT's training data for BiLSTM limited its performance greatly.

Test Sample Considerations:

- A 5 times more significant number of test samples were used for analyzing BiLSTM (9,917 vs. 2,000).
- Larger test sets yield more reliable accuracy estimates.

Model-Specific Observations:

- **BiLSTM:** Excellent performance with full training data and sufficient training time.
- **DistilBERT:** Achieved remarkable results and demonstrated high efficiency despite using only 25% of the training data.
- **BERT:** The freezing of layers, using the smallest batch size, and the limitation of training data to 3,000 samples (which was quite aggressive), all contributed to the poor performance.

This study demonstrated that BiLSTM achieves the best accuracy (87.70%) among the compared models, proving its reliability under constrained settings. DistilBERT performed efficiently and maintained high accuracy, suggesting that knowledge-distilled transformers are suitable for balanced trade-offs between accuracy and computation. The findings indicate that while transformer architectures dominate large-scale NLP, traditional recurrent models still excel in optimized environments.

References

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning Word Vectors for Sentiment Analysis.* In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.

- The paper introduced the IMDB dataset and focused on learning sentiment-aware word embeddings using a semi-supervised probabilistic model.
- Our project differs by using end-to-end deep learning models (BiLSTM, DistilBERT, BERT) for direct sentiment classification rather than embedding generation.
- While their method relied on separate embedding training and logistic classifiers, ours integrates contextual understanding and classification within a single neural framework.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* NAACL.

- The original BERT model was designed as a general-purpose language representation model trained on massive text corpora using pretraining objectives.
- Our project adapts BERT specifically for sentiment classification through fine-tuning on the IMDB dataset with limited data and partially frozen layers.
- This shifts the focus from large-scale language pretraining to task-specific optimization under computational constraints.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter.* arXiv:1910.01108.

- DistilBERT was proposed as a compressed, faster version of BERT using knowledge distillation to reduce size and inference time.
- Our work applies DistilBERT directly to sentiment classification with fine-tuning focused on task performance rather than model compression.
- Thus, the difference lies in applying a distilled model for targeted downstream learning instead of demonstrating the distillation process itself.