Automation and Machine Learning (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2026.070214
ISSN 2516-5003 Vol. 7 Num. 2

Fine-Tuned BERT for Sentiment Classification of Online Reviews

Hongru Li[a], Zhitao Wu[b,*]

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China
[a]lihongru415@outlook.com,[b]aswzt@163.com
**Corresponding author*

Keywords: Online reviews; Sentiment classification; BERT; Fine tuning; Contextual representation

Abstract: Online review sentiment classification requires models to interpret contextual meaning, negation, and evaluative expressions that may be distributed across a document. This paper presents a fine-tuning study of BERT and compares it with TextCNN and LSTM classifiers on the public IMDB movie review dataset. The design preserves the official test partition, derives validation data from the training partition, and controls the review passage supplied to each model. BERT is adapted with a binary classification head, while the baselines use pretrained word embeddings and task-specific encoders. Learning rate, batch size, and maximum sequence length are examined alongside classification accuracy, positive-class precision, recall, F1, and inference latency. Training duration is included to expose the computational cost of contextual fine-tuning. In the experiments, BERT achieves an accuracy of 93.8% and an F1 of 93.8%, accompanied by greater training and inference costs than the baseline systems. The analysis evaluates gains associated with each complete pretrained system and examines how truncation can remove sentiment-bearing evidence. Subsequent work will address long reviews, mixed sentiment, domain transfer, and probability calibration to support customer-opinion summarization.

1. Introduction

Online reviews often combine factual description with explicit and implicit evaluation. A positive adjective may be negated, qualified, or contradicted by a later sentence. Convolutional sentence classifiers can detect useful local patterns [1], while LSTM networks model information across a sequence through gated recurrence [2]. These approaches provide informative baselines for evaluating contextual pretraining on a supervised task with limited training data.

The Transformer uses attention to relate sequence positions without recurrent state propagation [3]. BERT builds pretrained bidirectional representations that can be adapted to downstream language tasks through fine-tuning [4]. This provides a plausible route to better review classification, because a token representation can reflect surrounding expressions. The computational cost of pretrained context is evaluated alongside accuracy using consistent baselines

and a clearly defined text budget.

The present study asks whether a fine-tuned BERT system offers a useful accuracy and F1 advantage for binary movie review sentiment classification, and what training and inference costs accompany that advantage. It also examines the sensitivity of fine-tuning to learning rate and batch size and the effect of maximum sequence length.

2. Related Theories and Technologies

2.1. Contextual Representation and BERT Adaptation

Self-attention forms token representations using weighted information from other positions [3]. BERT pretraining conditions representations on both left and right context, and downstream fine-tuning can add a small output component while updating the pretrained encoder [4]. The model uses the uncased BERT base configuration with 12 encoder layers, hidden width 768, and 12 attention heads.

Each review passage is encoded with the model’s own subword tokenizer, including the required initial classification token and terminal separator. The final hidden state of the classification token passes through dropout and a two-class linear layer. Softmax produces class probabilities, and cross entropy provides supervision. All encoder layers are updated during fine-tuning. All runs use one fixed tokenizer vocabulary, checkpoint revision, and preprocessing configuration to avoid silent differences between runs.

2.2. TextCNN and LSTM Baselines

TextCNN applies convolutional filters over token embeddings and pools the strongest responses to form a document representation [1]. The baseline uses filter widths three, four, and five, with 100 filters per width, followed by global maximum pooling and a binary classifier. The LSTM control uses one unidirectional 128-unit recurrent layer and its final nonpadding hidden state [2]. Both baselines receive 300-dimensional GloVe embeddings, which are based on global word co-occurrence information [5]. Embeddings are updated during supervised training.

2.3. Regularization and Sequence Length

Dropout provides a regularization mechanism by suppressing selected activations during training [6]. The BERT classification head uses rate 0.10, while the TextCNN and LSTM output representations use rate 0.50. These declared settings reflect different model designs, and they are fixed throughout the architecture comparison.

Maximum sequence length limits both retained evidence and computation. With a fixed number of attention heads, attention-score storage grows quadratically with the number of positions, while other components contribute additional costs. Actual latency also depends on model components and implementation. Shorter passages can remove concluding judgments or the resolution of a negation, whereas longer passages can include irrelevant plot summaries. Maximum length is therefore evaluated as an empirical quality-cost tradeoff.

3. Experimental Design and Implementation

3.1. Dataset Preparation

The source is the public IMDB Large Movie Review Dataset for positive-versus-negative review

classification. Its official labeled training and test partitions each contain 25,000 reviews. The protocol reserves a stratified 2,500 reviews from the training partition for validation and uses the remaining 22,500 for fitting. The official test partition remained untouched during configuration selection, and only labeled reviews are included.

HTML break markers were replaced with spaces, and redundant whitespace was normalized. Stop words, punctuation, and negation terms are retained because their removal can alter sentiment. Case handling follows the uncased checkpoint, and baseline text is lowercased consistently. Exact duplicates were grouped before the internal training-validation split. Potential duplicates across the official training and test partitions were audited.

To reduce differences in available evidence, the reference comparison supplies all models with the same raw review prefix. Specifically, the largest whole-word prefix fitting within 254 BERT subword positions is selected, leaving two positions for special tokens within a maximum length of 256. Each system starts from the same retained passage while using its own tokenization procedure. Baseline padding is excluded from pooling and recurrent state selection.

For the sequence-length sweep, the corresponding BERT text budgets are 126, 254, and 510 subword positions, producing total limits of 128, 256, and 512. Each setting may expose a different review prefix and is treated as an information-and-cost comparison.

3.2. Experimental Environment

The implementation uses Python, PyTorch, and a Transformer-model library on one CUDA-capable GPU; all models were trained and timed on the same machine.

Training duration includes fitting and validation over the stated epoch budget, excluding model download, tokenizer construction, and source-file parsing. Model-only inference is measured at batch size one with fixed padding to the configured maximum length, after 100 warm-up passes and across at least 1,000 timed passes with accelerator synchronization. Tokenization and text cleaning are excluded from this metric.

3.3. Experimental Parameter Setting

The BERT reference uses Adam [7] at learning rate 0.00002, momentum coefficients 0.9 and 0.999, and zero weight decay. The study uses this Adam configuration throughout the BERT experiments. Learning rate increases linearly during the first 10% of updates and then decreases linearly to zero. Global gradient norm is clipped at 1.0. Batch size is 16, and the training budget is four epochs.

TextCNN and LSTM use Adam at rate 0.001 with batch size 64 and a maximum of 15 epochs. Their training stops after three epochs without a validation-F1 improvement of at least 0.001. The BERT checkpoint with highest validation F1 is selected from the fixed four-epoch run. Model-specific budgets support practical training comparisons, and seconds per update and processed tokens per second provide complementary throughput measures.

The fine-tuning rate candidates are 0.00001, 0.00002, and 0.00005 at batch size 16. Batch sizes eight and 32 are additional controls at the reference rate, with gradient accumulation disabled. The sequence-length comparison fixes rate 0.00002 and batch size 16. Validation F1 is the primary selection criterion, followed by validation accuracy. Test labels are excluded from checkpoint, learning-rate, batch-size, and text-budget selection.

3.4. Experimental Indicators

Positive sentiment is the designated positive class. Precision is TP divided by TP plus FP, recall

is TP divided by TP plus FN, and F1 is twice precision times recall divided by their sum. Accuracy is the number of correctly labeled reviews divided by the total number of reviews. A zero denominator receives zero for the affected precision, recall, or F1 calculation. All percentages use this binary positive-class convention.

The operating rule assigns the class with greatest softmax probability, equivalent to a positive probability threshold of 0.5 for the two-class output. The threshold was fixed before final testing. Subsequent analyses will use paired review-level bootstrap intervals to quantify uncertainty in performance differences, with resampling preserved within identifiable review groups.

Inference latency is reported in milliseconds per review under the fixed-padding benchmark, and total training duration is reported in minutes. These quantities answer different computational questions. Training cost affects the feasibility of adaptation and repeated tuning; latency affects interactive or bulk scoring.

4. Experimental Results and Analysis

4.1. Classification Quality and Model Cost

Table 1 reports the comparison on the reference passage budget. Precision, recall, and F1 refer to positive sentiment.

Table 1. Review classification models and computational cost

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Latency (ms)	Training (min)
TextCNN	88.4	89.3	87.2	88.3	1.8	7.6
LSTM	89.5	88.8	90.4	89.6	4.7	18.2
BERT	93.8	93.3	94.4	93.8	15.9	54.3

BERT exceeds TextCNN by 5.4 percentage points in accuracy and LSTM by 4.3 points. Its training duration is about three times the LSTM duration, and its model-only latency is more than three times as high. The additional computation is evaluated according to the review-processing workload.

The LSTM has higher recall but lower precision than TextCNN, showing that accuracy alone can conceal a different balance between false-positive and false-negative predictions. BERT improves both quantities. Since all systems use pretrained information differently, these results characterize the complete classification systems and their respective pretrained representations.

4.2. Learning Rate and Batch Size Sensitivity

Table 2 changes learning rate at batch size 16 and then changes batch size at the reference rate. The reference row matches Table 1. This design provides controlled pairwise comparisons of learning rate and batch size.

Table 2. BERT learning rate and batch size settings

Learning rate	Batch size	Accuracy (%)	F1 (%)	Training time (min)
0.00001	16	93.1	93.1	56.8
0.00002	16	93.8	93.8	54.3
0.00005	16	92.6	92.6	53.9
0.00002	8	93.6	93.6	79.5
0.00002	32	93.4	93.4	39.7

The middle learning rate has the highest F1 among the three rates. The difference between

0.00002 and 0.00005 is 1.2 percentage points despite similar training durations, which suggests that the larger rate disrupts useful pretrained representations.

Batch size eight has a longer training duration because it performs more updates over the same number of epochs. Batch size 32 is faster but has an F1 lower by 0.4 points than the reference. Because the number of updates and the warm-up schedule also change, the comparison evaluates practical batch-size settings. Subsequent experiments will use gradient accumulation and an equal-update budget to test whether the pattern persists under a different computational constraint.

4.3. Maximum Sequence Length and Retained Context

Table 3 varies the maximum BERT input length, including special tokens. Rate and batch size remain fixed. The longer rows expose more of a review when the original text exceeds the corresponding limit; they also incur higher model computation under fixed padding.

Table 3. Sequence length and review classification efficiency

Maximum length	Accuracy (%)	F1 (%)	Latency (ms)	Training time (min)
128	92.6	92.5	8.6	30.2
256	93.8	93.8	15.9	54.3
512	94.0	94.0	36.8	119.6

F1 increases by 1.3 points from length 128 to 256, then by only 0.2 points from 256 to 512. The latter increment adds 20.9 milliseconds of latency and 65.3 training minutes. Given the small quality difference, the target workload is decisive when choosing the text budget. Length 256 provides the preferred quality-cost compromise for the stated workload, and repeated runs will further assess the stability of the 0.2-point difference.

Prefix truncation remains a limitation even at the larger budget. A review may reverse its evaluation near the end, and a retained prefix may omit that evidence. A subsequent comparison will evaluate head-only, head-and-tail, and segmented review aggregation with an explicit total token-processing budget. Performance will also be stratified by whether a review is truncated, because aggregate F1 can conceal a disproportionate loss on long documents.

5. Research Conclusions and Prospects

5.1. Research Conclusions

This paper presents a BERT fine-tuning study for online movie review sentiment classification, with TextCNN and LSTM baselines and controlled learning-rate, batch-size, and length comparisons. Accuracy, positive-class precision, recall, and F1 describe discrimination, while training duration and inference latency expose computational requirements. The BERT reference configuration achieves 93.8% accuracy and F1, 93.3% precision, and 94.4% recall, with a training duration of 54.3 minutes, outperforming both baselines on the official test partition.

The results indicate that BERT is preferable when its quality gain justifies the measured computational cost. Batch size and maximum length are selected from validation data under an explicit resource budget. Lower-cost baselines remain suitable for applications that prioritize faster adaptation or scoring and accept their error levels.

5.2. Research Limitations and Prospects

Binary movie review labels cover a narrower sentiment range than neutral, mixed, aspect-specific, or sarcastic opinion in general online commerce. Tokenization and prior training differ

across the pretrained systems. Prefix truncation can remove decisive judgments, and deployment class distributions may differ from the balanced benchmark. Downstream timing excludes pretraining expenditure.

Future work will evaluate product and service reviews using independently labeled target-domain sets, analyze long-document strategies at matched token budgets, and report probability calibration under class imbalance. Parameter-efficient adaptation will be considered after the full fine-tuning reference is measured. Efficiency comparisons will include search cost, peak memory, throughput, tokenization time, and model-only latency. These extensions will clarify when contextual classification provides a useful practical improvement beyond the results presented here.

References

[1] Kim Y. Convolutional neural networks for sentence classification[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014: 1746-1751.

[2] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.

[4] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of NAACL-HLT. 2019: 4171-4186.

[5] Pennington J, Socher R, Manning C D. GloVe: Global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[6] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[7] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//International Conference on Learning Representations. 2015.