Sentiment Analysis of Yelp Review Dataset: A Comparative Study of Machine Learning Methods

Krishna Thakar[1], Mohamed Abu Sheha[2], Emmanuel Thompson[2]
[1]Harrison College of Business and Computing, Southeast Missouri State University, USA
[2]Department of Mathematics, Southeast Missouri State University, USA
{kthakar1s, mabusheha, ethompson}@semo.edu

Abstract

Sentiment analysis involves analyzing text to determine whether the sentiment expressed is positive, negative, or neutral. In the context of online reviews, such as those on Yelp, sentiment analysis helps businesses assess customer satisfaction and identify areas for improvement. Given the large volume of user-generated content, restaurants often struggle to extract actionable insights from feedback, making sentiment analysis an efficient tool for categorizing reviews and highlighting customer concerns. This study focuses on sentiment analysis of Yelp reviews. The main research question is: How can Natural Language Processing (NLP) combined with statistical machine learning methods be applied to classify sentiment in Yelp reviews and provide actionable insights for improving customer satisfaction, service quality, and business performance? The study used 21,000 Yelp reviews, utilizing NLP approaches - tokenization, stop-word removal, and vectorization. Comparative classification predictive modeling and analysis were done across traditional machine learning (Logistic Regression, Support Vector Machine (SVM), Naïve Bayes, Random Forest), deep learning methods (CNN, LSTM, BiLSTM, GRU, RNN), and an advanced transformer-based (RoBERTa) model. Results showed that RoBERTa outperformed the other candidate methods. These findings highlight the potential of advanced NLP techniques to offer businesses practical ways to address customer complaints, enhance service quality, and drive overall business performance.

1 Introduction

Online review platforms like Yelp generate enormous volumes of user-generated text that carry rich signals about customer satisfaction. For restaurant owners and service businesses, manually processing thousands of reviews to extract actionable feedback is simply not feasible. Sentiment analysis, the task of automatically classifying text as positive, negative, or neutral, offers a scalable solution. As Natural Language Processing (NLP) methods have grown more sophisticated, the range of available approaches has expanded dramatically, from classical statistical models to deep learning architectures to transformer-based language models.

Despite this progress, the research community lacks consensus on which class of methods actually performs best for review-level sentiment classification. Prior comparative studies have produced conflicting results. Some work found that simpler models like Logistic Regression and SVM outperform more complex approaches such as gradient boosting, LSTM, and even transformer-based models on the Yelp dataset (Liu, 2020). Meanwhile, other studies using fine-tuned transformers on Yelp reviews found them to outperform traditional machine learning approaches (Areshey and Mathkour, 2023). Critically, most existing comparisons either omit the latest transformer variants like RoBERTa, or do not evaluate all three tiers (classical ML, basic deep learning, and transformers) on the same dataset under the same conditions.

This paper addresses that gap with a controlled, three-way comparative study on 21,000 Yelp reviews. We evaluate classical machine learning methods (Logistic Regression, SVM, Naïve Bayes, Random Forest), standard deep learning architectures (CNN, LSTM, BiLSTM, GRU, RNN), and a fine-tuned RoBERTa transformer model, all trained and tested on the same data split. Our results show that RoBERTa achieves the highest classification accuracy, but, perhaps more interestingly, classical models like SVM and Logistic Regression outperform CNN, LSTM, and the other recurrent architectures. This suggests that basic deep learning architectures offer little advantage over well-tuned classical baselines for this task.

The rest of this paper is organized as follows: Section 2 reviews related work on sentiment analy-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 428–437

sis methods. Section 3 describes our dataset, preprocessing pipeline, feature representation strategy, model selection rationale, and evaluation setup. Section 4 presents results across all ten models. Section 5 discusses the implications of our findings, and Section 6 concludes.

2 Related Work

2.1 Sentiment Analysis: From Classical Methods to Transformers

Sentiment analysis has a long history in NLP, with foundational contributions establishing the core task of classifying opinion-bearing text. Early document-level work approached the problem through lexical orientation and bag-of-words classification (Turney, 2002; Pang et al., 2002). Building on these foundations, later neural models learned compositional sentiment representations directly from text rather than relying only on hand-engineered lexical statistics (Socher et al., 2013). Recent surveys show a clear trend away from recurrent and convolutional architectures toward transformer-based language models, with pretrained transformers representing a breakthrough in transfer learning for NLP tasks (Liu, 2020).

2.2 Transformers and RoBERTa in Sentiment Classification

The introduction of BERT (Devlin et al., 2019) and its successor RoBERTa (Liu et al., 2019) marked a significant shift in how sentiment models are built. RoBERTa improves on BERT by removing the next-sentence prediction objective, using dynamic masking, larger batch sizes, and more training steps, resulting in more robust representations. Despite growing interest in transformer models, there have been relatively few systematic studies examining their effectiveness specifically for review-level sentiment classification, with most prior work relying on various business or reviewer features that limit generalizability (Shad et al., 2024). A recent comprehensive review comparing BERT, RoBERTa, XLNet, DistilBERT, ALBERT, and T5 across 22 datasets found that no single transformer consistently dominates, with performance being highly dataset-dependent (Islam et al., 2024).

2.3 Sentiment Analysis on the Yelp Dataset

Several studies have used the Yelp Open Dataset as a benchmark. Areshey and Mathkour (2023) compared BERT, RoBERTa, ALBERT, DistilBERT, and XLNet specifically on Yelp reviews, finding that transformer models outperform traditional machine learning approaches for this task. A separate study using Yelp reviews from Saint Louis found a more nuanced picture: an LSTM trained on substantially less data outperformed zero-shot RoBERTa and BERT pipelines, achieving 77% accuracy versus RoBERTa's range of 55–91% depending on sentiment class (Mostafavi et al., 2026). These inconsistencies point to a meaningful open question, one this paper directly addresses.

2.4 The Gap This Work Fills

Existing comparative studies on Yelp sentiment classification typically compare either classical ML against transformers, or basic deep learning against transformers, but rarely all three tiers under identical controlled conditions. Crucially, no prior work has documented the degree to which standard recurrent architectures (vanilla LSTM, GRU, RNN) can catastrophically underperform on this task relative to even the simplest classical baselines. This paper provides that full three-way comparison using a consistent 21,000-review sample, fixed preprocessing pipeline, and standardized evaluation metrics across all models.

3 Methodology

3.1 Dataset

This study uses the Yelp Open Dataset, a publicly available collection of over 7 million business reviews spanning 2005 to 2022. Due to computational constraints, we restricted our analysis to 21,000 reviews sampled from the most recent portion of the dataset, preserving temporal recency while keeping training feasible on consumer-grade GPU hardware.

Sentiment labels were derived from star ratings using a mapping commonly adopted in prior Yelp sentiment studies (Areshey and Mathkour, 2023): reviews with 4–5 stars were labeled *positive*, 3-star reviews *neutral*, and 1–2 star reviews *negative*. This mapping treats the star rating as a noisy but practical proxy for reviewer sentiment, an assumption that is standard in the literature and avoids the cost of human annotation at scale. We acknowledge that this introduces label noise at class boundaries (a customer who rates a restaurant 3 stars may use language nearly identical to a 2-star or 4-star reviewer) and we return to this point in our discussion of the neutral class.

The raw label distribution before balancing was skewed toward positive reviews, a well-documented characteristic of the Yelp dataset (Bochkarev, 2024). To prevent classifiers from exploiting this imbalance as a trivial predictive shortcut, we downsampled to the minority class size, yielding a final balanced dataset of 21,000 reviews (∼6,667 per class). The dataset was partitioned into a training set (70%, ∼14,700 reviews) and a held-out test set (30%, ∼6,300 reviews), with 5-fold stratified cross-validation applied during model selection to ensure class proportions were preserved across folds.

3.2 Preprocessing

All reviews passed through a unified preprocessing pipeline before model-specific vectorization. First, raw text was cleaned by lowercasing, expanding contractions (e.g., can't → cannot), removing punctuation and special characters, normalizing whitespace, and collapsing repeated characters (e.g., gooood → good). English-language filtering via `langdetect` was applied to remove non-English reviews. Stopwords were removed, and tokens were lemmatized using `spaCy` (`en_core_web_sm`) in batches. A scope-based negation handling step was also applied: when a negation token (not, no, never, n't) was detected, the following three words received a `not_` prefix (e.g., *not good* → *not_good*), with scope resets at clause-boundary words (e.g., but, however) or inherently negative terms (e.g., bad, terrible). This preprocessing design makes negated expressions appear as distinct tokens in the processed text.

Table 1 summarizes the full preprocessing pipeline applied uniformly across all model classes.

3.3 Feature Representation

Vectorization differed by model class. For classical ML models, we applied TF-IDF with English stopword filtering and `max_features=5000`. For deep learning models (CNN, LSTM, BiLSTM, GRU, RNN), we used a Keras `Tokenizer` with `num_words=10000`, `lower=True`, and `oov_token="<UNK>"` fit on the training split, followed by post-padding and post-truncation to a maximum sequence length of 200 tokens. We set the neural vocabulary size to 10,000 as a practical balance between lexical coverage and computational efficiency. This choice preserves a broad set of frequent review terms for the deep learning models while avoiding an unnecessarily large embedding layer. For RoBERTa, text was tokenized using `RobertaTokenizerFast` with a maximum sequence length of 128 tokens; no separate vectorization step was needed as the pretrained model handles its own representations.

Step	Tool / Method	Details
Text Cleaning	Custom Python	Lowercasing, contraction expansion, punctuation and special character removal, whitespace normalization, repeated character collapsing (e.g., *gooood* → *good*), English-language filtering via `langdetect`
Tokenization	`spaCy`	Text split into individual tokens prior to lemmatization and negation handling
Lemmatization	`spaCy` `en_core_web_sm`	Tokens reduced to root form; stopwords removed; processed in batches for efficiency
Negation Handling	Scope-based (custom)	Negation tokens (*not, no, never, n't*) prefix the following 3 words with `not_`; scope resets at clause boundaries (*but, however*) or inherently negative terms (*bad, terrible*)
Vectorization	TF-IDF / Keras Embedding / RoBERTa Tokenizer	TF-IDF (vocab=5,000) for classical ML; Keras embedding (dim=100, vocab=10,000) for deep learning; `RobertaTokenizerFast` (max length=128) for RoBERTa
Data Balancing	Random undersampling	Downsampled to minority class size; final dataset: ∼21,000 reviews (∼6,667 per class)

Table 1: Summary of the preprocessing pipeline applied to all Yelp reviews prior to model training.

3.4 Model Selection Rationale

The ten models evaluated in this study were chosen to provide comprehensive coverage of the three dominant paradigms in NLP classification, enabling a principled three-way comparison rather than a pairwise evaluation.

Within the **classical ML** tier, Logistic Regression and SVM were selected as the two strongest established baselines for text classification (Pang et al., 2002), while Naïve Bayes was included as a probabilistic baseline with well-understood assumptions about feature independence. Random Forest was added to represent ensemble methods, which aggregate multiple decision trees to reduce variance, representing a common alternative to linear models in applied sentiment pipelines.

Within the **deep learning** tier, CNN was selected because convolutional filters over token windows function as learned n-gram detectors, making them a natural fit for short opinionated text (Kim, 2014). The recurrent architectures (vanilla RNN, LSTM, GRU, and BiLSTM) were included as a family to

examine whether architectural complexity within recurrent models translates to meaningful performance differences on this task. Specifically, the vanilla RNN serves as the simplest recurrent baseline; LSTM and GRU each introduce gating mechanisms to address the vanishing gradient problem; and BiLSTM extends LSTM with bidirectional processing. Evaluating all four together allows us to isolate the contribution of each architectural innovation.

RoBERTa was selected as the transformer representative because it represents the state-of-the-art in pre-trained encoder models for classification tasks at the time of this study, and because prior Yelp-specific work has used it as a benchmark (Areshey and Mathkour, 2023; Mostafavi et al., 2026). Using a single transformer model rather than several (e.g., BERT, DistilBERT, ALBERT) was a deliberate choice: our primary goal is a cross-paradigm comparison, not an intra-transformer comparison, and including multiple transformers would shift the experiment's center of gravity away from that question.

3.5 Models

We evaluated three broad classes of methods.

Statistical ML: Logistic Regression, SVM, Naïve Bayes, and Random Forest were trained using TF-IDF features. All models were tuned via 5-fold stratified cross-validation.

Deep Learning: CNN, LSTM, BiLSTM, GRU, and RNN were each trained with a shared architecture: an embedding layer (dim=100), task-specific recurrent or convolutional layers with 128 units/filters, a dropout rate of 0.3, and a dense layer of 64 units (ReLU) followed by a 3-class softmax output. All models used the Adam optimizer (lr=0.001), batch size 128, and were trained for 5 epochs with a 10% validation split. During cross-validation, 3 epochs were used per fold. For the CNN specifically, we used a single 1D convolution with 128 filters, kernel size 5, stride 1 (Keras default), and global max pooling. The recurrent models each used a single 128-unit recurrent layer, with BiLSTM wrapping the same LSTM cell bidirectionally.

Transformer: We fine-tuned `roberta-base` for 3-class sequence classification. Training used a learning rate of 1e-5, weight decay of 0.01, batch size of 32, gradient accumulation over 2 steps, and 1,000 warmup steps. Early stopping with patience of 3 was applied, with best model selection based on F1 score.

Model	Accuracy	Precision	Recall	F1 Score	AUC
Statistical Machine Learning					
SVM	0.76103	0.76322	0.76103	0.76172	0.90915
Logistic Regression	0.76041	0.76448	0.76441	0.76013	0.90757
Naïve Bayes	0.72671	0.73246	0.72671	0.72860	0.87732
Random Forest	0.71751	0.71593	0.71751	0.71592	0.87531

Table 2: Predictive model performance metrics for statistical machine learning methods.

Model	Accuracy	Precision	Recall	F1 Score	AUC
Deep Learning and Transformer Methods					
RoBERTa	**0.80112**	**0.80502**	**0.80112**	**0.80253**	**0.93237**
CNN	0.74746	0.74909	0.74746	0.74817	0.89802
BiLSTM	0.73280	0.73371	0.73280	0.73323	0.88507
LSTM	0.33614	0.58349	0.33614	0.17434	0.50671
RNN	0.33614	0.58349	0.33614	0.17434	0.50671
GRU	0.34144	0.41822	0.34144	0.19040	0.50927

Table 3: Predictive model performance metrics for deep learning and transformer methods.

All experiments were implemented using `scikit-learn` for classical ML models, `TensorFlow/Keras` for deep learning architectures, and the Hugging Face `Transformers` library for RoBERTa fine-tuning. A fixed random seed of 42 was used throughout to ensure reproducibility.

3.6 Evaluation

All models were evaluated on the same held-out test set using five metrics: accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1 score, and AUC (area under the ROC curve). Using macro-averaging rather than weighted averaging ensures that each class contributes equally to the reported score, preventing the well-performing positive class from masking failures on the harder neutral class. AUC was included as a threshold-independent measure of discriminative ability, particularly useful for comparing models that may have different decision boundary calibrations. Accuracy is reported for interpretability and comparability with prior work. All five metrics are computed on the same fixed test split, ensuring that comparisons across all ten models are directly controlled.

4 Results

Tables 2 and 3 present the performance of all models on the held-out test set across accuracy, precision, recall, F1 score, and AUC.

Overall winner. Fine-tuned RoBERTa achieved the highest performance across all metrics, with an

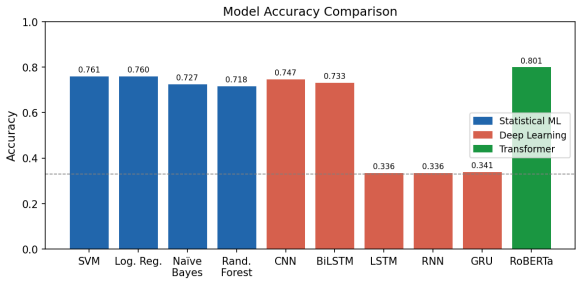

Figure 1: Accuracy comparison across all models. The dashed line indicates random baseline (∼0.33) for a balanced 3-class problem.

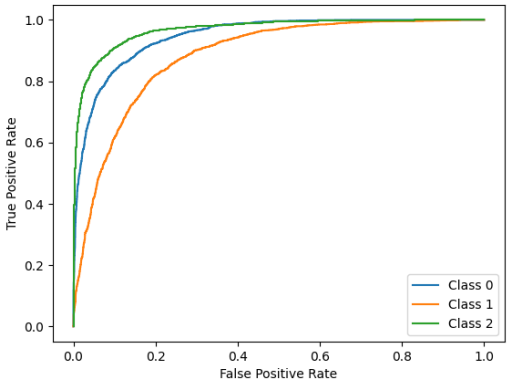

Figure 2: ROC curves for fine-tuned RoBERTa across the three sentiment classes (Class 0: Negative, Class 1: Neutral, Class 2: Positive). Neutral is the hardest class to discriminate.

accuracy of 80.1%, F1 score of 80.3%, and AUC of 0.932, confirming that transformer-based models offer the strongest predictive power for this task.

Figure 2 presents the ROC curves for RoBERTa across all three sentiment classes. The positive class achieves the highest AUC, reflecting the model's strongest discriminative ability for clearly positive language. The negative class curve follows closely, while the neutral class produces a noticeably lower AUC, consistent with the macro-level results. Importantly, all three curves remain well above the random baseline diagonal, confirming that RoBERTa learns meaningful decision boundaries for every class rather than succeeding by concentrating performance on one easy class at the expense of others. The separation between the neutral AUC and the other two classes visually confirms that neutral sentiment is a structurally harder classification target, a finding that holds across all non-collapsed models in this study.

Classical ML vs. Deep Learning. A notable finding is that the top two classical ML methods outperformed CNN and BiLSTM. SVM led the classical tier with 76.1% accuracy and AUC of 0.909, followed closely by Logistic Regression (76.0%, AUC 0.908). Both substantially outperformed CNN (74.7%) and BiLSTM (73.3%). This suggests that for review-level sentiment classification on TF-IDF-style features, well-tuned classical methods remain highly competitive with standard deep learning architectures.

Collapsed models. LSTM, RNN, and GRU produced near-random performance, with accuracies of approximately 33–34% and F1 scores below 0.20. Given the balanced class distribution of the dataset (∼33% per class), these results indicate that these three models failed to learn meaningful representations during training. We attribute this to optimization instability under the chosen hyperparameter configuration, as vanilla recurrent architectures are known to be sensitive to learning rate and sequence length settings.

Figures 3 and 4 present the confusion matrices for RoBERTa and SVM respectively, allowing a direct visual comparison of where each model fails. Both matrices show a consistent pattern: misclassifications are concentrated in the neutral class column, with neutral reviews being the most likely to be misclassified as either positive or negative. RoBERTa's confusion matrix is notably more balanced across all three diagonal entries, whereas SVM's matrix shows a more pronounced off-diagonal mass in the neutral row, consistent with its lower neutral recall reported in Table 2. Neither model exhibits a strong asymmetric bias toward predicting a single class, in sharp contrast to the collapsed recurrent models (LSTM, RNN, GRU) which predicted a single class for virtually all inputs. Figure 1 summarizes overall accuracy across all ten models, with the dashed random baseline at 0.33 making the collapsed models' failure immediately visible.

5 Discussion

This study evaluated ten models spanning three paradigms - classical machine learning, standard deep learning, and transformer-based fine-tuning for three-class sentiment classification on a balanced dataset of approximately 21,000 Yelp restaurant reviews (∼6,667 per class). The results reveal several important patterns that illuminate the relative strengths and weaknesses of different NLP approaches when applied to opinionated user-

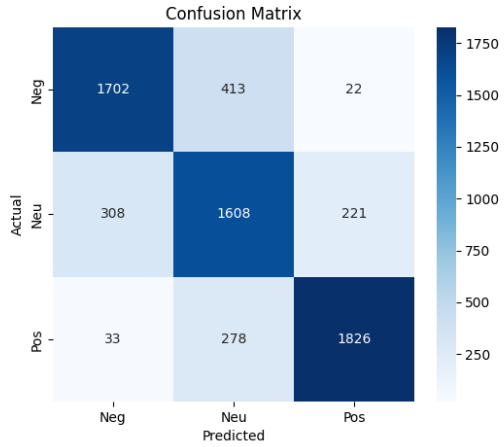

Figure 3: Confusion matrix for RoBERTa on the test set. Misclassifications are concentrated in the Neutral class.

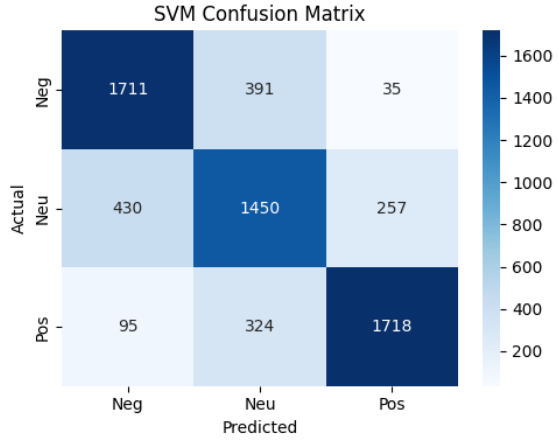

Figure 4: Confusion matrix for SVM, the best-performing classical ML model. Neutral misclassifications are notably higher than in RoBERTa.

generated text.

5.1 The Dominance of Fine-Tuned Transformers

RoBERTa achieved the highest overall performance across every metric, with an accuracy of 80.11%, macro-averaged F1-score of 0.8025, and AUC of 0.9324. Its superiority over all other models is attributable to several interrelated factors rooted in its pre-training paradigm. RoBERTa was pre-trained on over 160 GB of text using a masked language modeling objective, endowing the model with deep contextual understanding of English syntax, semantics, and pragmatics before encountering the Yelp review domain. During fine-tuning (learning rate of 1×10^{-5}, batch size 32, early stopping with patience 3), this pre-trained knowledge is efficiently adapted to restaurant review language with relatively few labeled examples.

A critical technical advantage lies in RoBERTa's Byte-Pair Encoding (BPE) subword tokenization. Yelp reviews are rich in informal language (misspellings ("delicous," "awsome"), elongations ("sooooo good"), and non-standard abbreviations) that would map to out-of-vocabulary tokens for traditional word-level tokenizers. BPE decomposes such tokens into recognizable subword units, preserving partial semantic signal. Furthermore, RoBERTa's multi-head self-attention can model long-range dependencies and contrastive constructions such as "The food was great but the service ruined the whole experience," resolving sentiment more accurately than bag-of-words or shallow sequential models.

5.2 Classical ML Outperforming CNN and BiLSTM

Perhaps the most counterintuitive finding is that SVM (76.10% accuracy, F1 = 0.7617) and Logistic Regression (76.04%, F1 = 0.7601) outperformed CNN (74.75%, F1 = 0.7482) and BiLSTM (73.28%, F1 = 0.7332). This result is well-explained by the interaction between feature representation strategy and dataset size.

The classical models used TF-IDF vectorization with a 5,000-feature vocabulary, a representation that captures global term frequency patterns across the corpus. For short, opinionated text, TF-IDF naturally emphasizes discriminative sentiment-bearing words ("terrible," "amazing," "mediocre") while down-weighting uninformative terms. SVM is particularly well-suited to this high-dimensional sparse feature space, finding the maximum-margin hyperplane efficiently when the feature count (5,000) is large relative to the sample size ($\sim$21,000).

In contrast, CNN and BiLSTM used 100-dimensional word embeddings trained entirely from scratch. With approximately 14,700 training samples, these embedding layers had to simultaneously learn meaningful word vector representations and the downstream classification task, a dual optimization burden that is acutely data-hungry. The absence of pre-trained embeddings such as GloVe or Word2Vec, which encode distributional semantics from billions of tokens, likely caused poor generalization for less frequent vocabulary. This finding aligns with prior observations that classical models with well-engineered features are surpris-

ingly competitive on moderate-sized text classification datasets (Wang and Manning, 2012).

5.3 Complete Collapse of LSTM, RNN, and GRU

The most striking result is the complete failure of the standalone LSTM (33.61% accuracy), vanilla RNN (33.61%), and GRU (34.14%) models. These accuracies are statistically equivalent to random chance for a balanced three-class problem (33.33%), indicating that these models learned no meaningful discriminative patterns. Inspection of their confusion matrices confirms the failure mode: all three models converged to a degenerate solution, predicting a single class for virtually all inputs.

Several factors converged to produce this training failure. Vanilla RNNs are highly susceptible to the vanishing gradient problem, where gradients diminish exponentially through long sequences. With a maximum sequence length of 200 tokens, the error signal from the output layer struggles to update the early recurrent and embedding layers effectively. Although LSTM and GRU architectures include gating mechanisms designed to mitigate this issue, they are not immune under suboptimal hyperparameters. The Adam optimizer learning rate of 0.001, combined with temporal weight sharing across 200 time steps, likely caused unstable updates that pushed models toward a degenerate attractor. Additionally, only 5 training epochs were used, typically insufficient for recurrent architectures training embeddings from scratch on a dataset of this size.

BiLSTM avoided complete collapse (73.28% accuracy) because its bidirectional architecture reads sequences in both forward and reverse order, effectively doubling the information flow and providing enough gradient signal to prevent trivial convergence. This suggests that the failures of LSTM, RNN, and GRU are largely attributable to hyperparameter sensitivity under constrained training conditions, rather than an inherent incapacity of recurrent architectures for this task. Future work should explore tuned learning rates ($\leq$1e-4), more training epochs, and pre-trained embeddings for these models.

5.4 The Neutral Class Problem

Across all models that learned meaningful patterns, the neutral sentiment class (3-star reviews) was consistently the hardest to classify. In RoBERTa's confusion matrix (Figure 3), per-class accuracy was 75.2% for neutral, compared to 79.6% for negative and 85.4% for positive. The SVM (Figure 4) showed a starker disparity: 67.8% neutral accuracy versus $\sim$80% for both other classes.

This difficulty is an inherent property of the three-class formulation. Neutral reviews are linguistically ambiguous by nature, frequently containing mixed sentiment within the same text (e.g., "The pasta was decent and reasonably priced, but the wait was too long and the waiter seemed disinterested"). The boundary between a mildly positive 3-star review and a clearly positive 4-star review, or between a mildly negative 3-star and a 2-star, is inherently subjective and reviewer-dependent. This generates label noise at class boundaries that disproportionately affects the neutral class, since it shares a border with both other classes. While our negation handling pipeline resolves phrases like "not bad" or "not great," it cannot resolve the deeper ambiguity of genuinely mixed-sentiment reviews.

5.5 Business Implications

These findings have direct practical implications for businesses leveraging NLP for customer feedback analysis. A fine-tuned RoBERTa model, achieving 80% accuracy and 93% AUC, is sufficiently reliable for automated sentiment monitoring, enabling real-time alerting on negative feedback spikes and trend analysis over time. However, the neutral class difficulty suggests that actionable insights are best derived from a binary positive/negative framework (filtering or consolidating 3-star reviews) rather than insisting on three-class granularity. For resource-constrained businesses without GPU infrastructure, SVM or Logistic Regression with TF-IDF vectorization provides a practical and competitive alternative, achieving 76% accuracy with training measured in seconds rather than hours, and deployable on standard web servers with incremental updates as new reviews arrive.

6 Conclusion

This paper presented a controlled three-way comparative study of sentiment classification on 21,000 balanced Yelp restaurant reviews, evaluating ten models spanning classical machine learning, standard deep learning architectures, and a fine-tuned transformer. The experiment was designed to address a gap in the existing literature: most prior comparative studies evaluate only two of the three

paradigms, and none have systematically documented the degree to which recurrent architectures can fail relative to classical baselines under standard training conditions.

Our results yielded three principal findings. First, fine-tuned RoBERTa achieved the strongest overall performance (80.1% accuracy, F1 = 0.803, AUC = 0.932), confirming that the contextual representations learned during large-scale pre-training transfer effectively to the review sentiment domain. Second, classical models SVM and Logistic Regression (both at ∼76% accuracy) outperformed CNN (74.7%) and BiLSTM (73.3%), despite the latter having significantly more parameters. This finding reinforces the view that TF-IDF with well-tuned linear models remains a strong and practical baseline for moderate-scale text classification tasks. Third, vanilla LSTM, RNN, and GRU completely collapsed to near-random performance (∼33%), a result we attribute to the combination of training-from-scratch embeddings, a high learning rate, and insufficient training epochs rather than any fundamental architectural incapacity.

Together, these findings suggest a practical hierarchy for sentiment analysis practitioners: transformer models should be preferred when GPU resources are available; classical models with TF-IDF provide a reliable and computationally inexpensive fallback; and vanilla recurrent architectures without pre-trained embeddings and careful hyperparameter tuning should be avoided for tasks of this scale.

Future work should investigate three directions. First, replacing training-from-scratch embeddings with pre-trained GloVe or FastText vectors for the recurrent architectures to determine whether the observed failures are hyperparameter-driven or representation-driven. Second, extending the analysis to aspect-level or opinion-level sentiment classification at the sentence/clause level, which would allow more granular feedback extraction (e.g., separating food quality sentiment from service sentiment) and may reduce the ambiguity concentrated in review-level neutral labels. Third, scaling the experiment to a larger subset of the Yelp dataset to assess whether the classical-versus-deep-learning gap narrows with more training data.

Limitations

Dataset size and sampling bias. Due to computational constraints, this study used only 21,000 of the over 7 million available reviews in the Yelp Open Dataset. The sample was drawn from the most recent portion of the dataset to preserve temporal recency; however, this introduces a temporal sampling bias. Recent reviews may disproportionately reflect post-pandemic dining patterns, shifts in reviewer demographics, or changes in Yelp's user base over time. It is therefore unclear whether the performance rankings observed here would hold on older reviews or on a randomly sampled subset of the full dataset.

Star-rating labels as a proxy for sentiment. Sentiment labels were derived automatically from star ratings rather than human annotation. While this is standard practice in the literature (Areshey and Mathkour, 2023), it introduces systematic label noise, particularly at class boundaries. Two reviewers with nearly identical dining experiences may assign 2 and 4 stars respectively, using near-identical language. This subjectivity disproportionately affects the neutral class (3-star reviews), which must share a linguistic boundary with both positive and negative classes simultaneously. The neutral class accuracy deficits observed across all models (Section 5.4) are partly a consequence of this labeling strategy rather than a pure model failure.

Downsampling and information loss. To achieve class balance, we downsampled the majority classes (positive and neutral reviews) to match the minority class size. While this prevents classifiers from exploiting the natural class imbalance as a predictive shortcut, it discards a substantial portion of the available training data. Alternative strategies such as oversampling with SMOTE, cost-sensitive learning, or class-weighted loss functions were not explored and may yield different performance profiles, particularly for the recurrent architectures that appeared most sensitive to data volume.

Deep learning models trained without pre-trained embeddings. CNN, LSTM, BiLSTM, GRU, and RNN were all trained with embeddings initialized from scratch rather than with pre-trained word vectors such as GloVe or FastText. This placed these models at a representational disadvantage relative to RoBERTa, which benefits from pre-training on over 160 GB of text. The comparison between classical models and deep learning architectures is fair (both groups lack external pre-training), but the comparison between scratch-trained deep learning models and RoBERTa con-

flates architectural differences with representational differences. It is possible that CNN or BiLSTM with GloVe embeddings would narrow the gap with RoBERTa substantially.

Recurrent model hyperparameters not tuned. The complete collapse of LSTM, RNN, and GRU (Section 5.3) is attributed to hyperparameter sensitivity rather than fundamental architectural incapacity. However, only a single hyperparameter configuration was evaluated for these models (lr=0.001, 5 epochs, batch size 128). A systematic hyperparameter search over learning rates, training epochs, and sequence lengths was not conducted for the recurrent tier. The results therefore represent these architectures under standard but potentially suboptimal settings, and should not be interpreted as evidence that recurrent models cannot perform competitively on this task.

Single-domain, single-platform generalizability. All experiments were conducted on Yelp restaurant reviews. The preprocessing pipeline, negation handling heuristics, and vocabulary were tailored to this domain. It is unclear whether the performance hierarchy observed here (transformers, then classical ML, then deep learning without pre-training) would replicate on reviews from other platforms such as Google Maps or TripAdvisor, or on other domains such as hotel or product reviews, which differ in vocabulary, review length distributions, and reviewer behavior.

No statistical significance testing. Performance differences between models are reported as point estimates without confidence intervals or significance tests. Some gaps are narrow: for example, SVM (76.10%) and CNN (74.75%) differ by only 1.35 percentage points in accuracy. Whether this difference is statistically reliable given the test set size (∼6,300 reviews) was not formally assessed. Future work should report McNemar's test or bootstrap confidence intervals to establish whether observed rankings are statistically robust.

References

Ali Areshey and Hassan Mathkour. 2023. Transfer learning for sentiment classification using bidirectional encoder representations from transformers (BERT) model. *Sensors*, 23(11):5232.

Viacheslav Bochkarev. 2024. Yelp reviews sentiment analysis. GitHub repository. Last Publish January 4, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Md. Shofiqul Islam, Muhammad Nomani Kabir, Ngahzaifa Ab Ghani, Kamal Zuhairi Zamli, Nor Saradatul Akmar Zulkifli, Md Mustafizur Rahman, and Mohammad Ali Moni. 2024. Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach. *Artificial Intelligence Review*, 57(3):62.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Siqi Liu. 2020. Sentiment analysis of yelp reviews: A comparison of techniques and models. *Computing Research Repository*, arXiv:2004.13851.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Sepehr Mostafavi, Yeganeh Yahyavi, and Reza Ravanmehr. 2026. Systematic literature review on sentiment analysis using transformers. *International Journal of Data Science and Analytics*, 22(1):49.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Ralph Shad, Kaledio Potter, and Abram Gracias. 2024. Natural language processing (NLP) for sentiment analysis: A comparative study of machine learning algorithms. *Preprints.org*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.