

A COMPARATIVE STUDY OF MACHINE LEARNING TECHNIQUES FOR SENTIMENT CLASSIFICATION OF SOCIAL MEDIA TEXT

Pankaj kumar Jain¹, Dr.M.S.Ansari²

¹Research Scholar, Department of Computer Science Engineering, Eklavya University, Damoh, M.P. Email ID: pankajjain80@gmail.com

²Supervisor, Department of Computer Science Engineering, Eklavya University, Damoh, M.P.

Abstract: Social media platforms generate vast volumes of opinionated text, making automated sentiment classification a core task in natural language processing. One high-stakes application is the analysis of public opinion on India's National Education Policy 2020 (NEP-2020), which has sustained online debate through its fifth anniversary in 2025 and whose comments need classifiers of well-understood accuracy, cost, and robustness. This study compares eight machine learning techniques, spanning classical classifiers (multinomial Naïve Bayes, logistic regression, support vector machines, and random forests), neural architectures (convolutional and bidirectional LSTM networks), and pre-trained transformers (BERT and RoBERTa), on a three-class Twitter sentiment benchmark (TweetEval) that serves as a controlled proxy for NEP-2020 public comments. Models were evaluated using accuracy, macro-averaged F1, training cost, and sensitivity to training-set size. The dataset is documented in detail, including a marked shift in class distribution between training and test partitions, and a preprocessing ablation examines how text normalization choices affect each model family. Paired significance tests with multiple-comparison correction assess whether observed differences exceed seed-to-seed and test-sampling variability. Transformer models achieved the strongest performance, with RoBERTa reaching a macro-F1 of 74.5%, while a linear support vector machine remained competitive at a fraction of the computational cost when labeled data were scarce or compute was limited. Errors across all models concentrated on the neutral class, reflecting the inherent ambiguity of short informal text. The findings offer practical guidance for selecting sentiment classifiers under different resource constraints, including for the analysis of public comments on NEP-2020.

Keywords: sentiment analysis, text classification, social media, machine learning, deep learning, transformers, statistical significance, preprocessing, National Education Policy 2020, public opinion

Received: 03/07/2026

Revised: 08/08/2026

Acceptance: 28/09/2026

Publication: 16/10/2026

INTRODUCTION

Millions of users share opinions about products, public figures, and events on social media every day. Automatically identifying the polarity of these opinions, known as sentiment analysis, supports applications ranging from brand monitoring to public health surveillance (Liu, 2012; Pang & Lee, 2008). Social media text is difficult to analyze because it is short, noisy, and rich in slang, abbreviations, emojis, and sarcasm, all of which weaken methods designed for formal prose.

Public opinion is especially consequential in education policy. India's National Education Policy 2020 (NEP-2020), announced on 29 July 2020, replaced the 1986 policy with a restructured 5+3+3+4 school curriculum, four-year undergraduate programs with multiple entry and exit options, and a credit-based system, among other reforms (Kulal et al., 2024; Ministry of Education, 2020). Ahead of a national education summit marking the policy's fifth anniversary on 29 July 2025, the Ministry of Education cited, among other figures, 13,076 schools selected under the PM SHRI initiative as exemplar NEP schools (Ministry of Education, 2025). Reception of the policy has not been uniform. Critical appraisals question its direction and assumptions (Batra, 2020; Govinda, 2020), its language-in-education provisions have drawn particular scrutiny (Sengupta, 2021), and stakeholder research finds that experts, students, and teachers weigh its promises and implementation demands differently (Kulal et al., 2024).

Because much of this debate takes place online, a growing body of work mines public posts and comments about NEP-2020 to gauge acceptance, from Twitter studies soon after the policy's announcement (Kaurav et al., 2020, 2021) to classifier comparisons published as recently as 2026 (Chaudhary et al., 2024, 2026). These studies report high accuracies, but they use different corpora and label sources, cover different sets of models, and are difficult to compare or to transfer to new comment collections. A controlled comparison of classifier families on a transparent benchmark, with significance testing, is therefore a necessary step before public comments on NEP-2020 can be analyzed at scale.

Early work showed that standard machine learning classifiers outperform hand-built lexicon rules for sentiment (Pang et al., 2002). Since then, the field has moved from sparse lexical features to dense word embeddings (Mikolov et al., 2013; Pennington et al., 2014), neural sequence models (Hochreiter & Schmidhuber, 1997; Kim, 2014), and large pre-trained transformers (Devlin et al., 2019; Vaswani et al., 2017). Because each family has distinct accuracy, cost, and data requirements, practitioners need direct, controlled comparisons. Such comparisons are only informative if the data are described transparently and if differences between models are tested against the variability that arises from random initialization, finite test sets, and preprocessing choices (Berg-Kirkpatrick et al., 2012; Dror et al., 2018). This paper addresses four research questions:

RQ1. How do classical, neural, and transformer models compare in classification performance on social media text?

RQ2. How does performance change as the amount of labeled training data varies? **RQ3.** Which sentiment classes are most frequently confused by the strongest model? **RQ4.** Are the observed differences between models statistically significant, and how sensitive are they to preprocessing choices?

Scope. The experiments reported here use TweetEval rather than an NEP-specific corpus. This is deliberate: TweetEval is an openly documented three-class benchmark with a fixed test set, which allows the classifier comparison, significance tests, and preprocessing ablation to be run in a controlled and reproducible setting. The conclusions are intended to guide the choice of classifier for the subsequent analysis of public comments on NEP-2020; the Dataset, Discussion, and Limitations sections state what must still be validated on NEP-specific data.

RELATED WORK

Sentiment Classification Methods

Pang et al. (2002) applied Naïve Bayes, maximum entropy, and support vector machines to movie reviews and found that bag-of-words features with machine learning outperformed keyword baselines. Wang and Manning (2012) later demonstrated that Naïve Bayes and SVM variants using bigram features remain strong baselines. For Twitter specifically, Go et al. (2009) used emoticons as distant supervision to build large training corpora, and Mohammad et al. (2013) achieved top results in an early shared task using SVMs with handcrafted lexical and sentiment-lexicon features. Neural methods reduced reliance on feature engineering. Socher et al. (2013) introduced recursive models over a sentiment treebank, Maas et al. (2011) learned sentiment-aware word vectors, and Kim (2014) showed that a simple convolutional network over pre-trained embeddings performs well on sentence classification. Transformer-based language models further improved results: BERT (Devlin et al., 2019) and its robustly optimized variant RoBERTa (Y. Liu et al., 2019) set new benchmarks across many tasks. The SemEval-2017 Task 4 evaluation (Rosenthal et al., 2017) and the TweetEval benchmark (Barbieri et al., 2020) provide standardized settings for tweet-level sentiment, which we adopt here.

Large language models (LLMs) have since added a further option for sentiment analysis. Zhang et al. (2024) evaluated LLMs in zero-shot and few-shot settings on 13 sentiment tasks across 26 datasets, compared them with smaller models trained on domain-specific data, and asked whether current evaluation practice remains adequate in the LLM era. Fine-tuned base-sized encoders such as those studied here remain a practical reference point because they are inexpensive to run and can be evaluated with the paired tests described below, and any LLM-based pipeline for policy comments should be compared against them.

Public Opinion on NEP-2020

Research on the reception of NEP-2020 spans survey, qualitative, and computational approaches. Kulal et al. (2024) used an explanatory sequential mixed-methods design with students, teachers, and subject experts. Experts were the most positive about the policy's vision, students valued the multiple entry-exit options and the new credit system, and

teachers stressed the need for capacity building and training; respondents overall expected the policy to raise the quality of education. These differences between stakeholder groups show that an aggregate polarity score can hide divergent concerns.

Social media studies began soon after the policy was announced. Kaurav et al. (2020) combined qualitative content analysis of policy text with Twitter mining, and Kaurav et al. (2021) extended that work and reported that most users were optimistic about implementation in higher education. Sharma and Vasuja (2021) examined social-media-induced polarization around NEP-2020, and Saini et al. (2021) analyzed tweets to study the behavioral response of Indian citizens to the policy's approval. At a broader level, Göçen et al. (2024) analyzed 157,943 tweets on higher education in 2023 and combined sentiment analysis with topic modeling, an approach that also suits NEP-2020 discourse, where themes such as multilingualism, credit transfer, and teacher preparation are likely to attract different sentiment. Dong and Lian (2021) review the methodological challenges of social-media-based public opinion analysis, including the question of whose opinion such data represent.

Several studies compare machine learning classifiers on NEP-related text. Siddique and Kumar (2023) built the NEPEduset tweet corpus, scored it with TextBlob, VADER, and a customized method (SentiNEP), and applied supervised models. Chaudhary et al. (2024) compared an LSTM with Naïve Bayes, SVM, and k-nearest-neighbor classifiers and reported 94% accuracy for the LSTM, and Chaudhary et al. (2026) compared logistic regression, XGBoost, decision trees, and Bernoulli Naïve Bayes and reported 91.22% accuracy for logistic regression. Three limitations follow for the present purpose. First, accuracies of this size depend on the corpus, the label source, the class balance, and the split design, so they cannot be compared across studies or with the results on a distribution-shifted benchmark such as TweetEval. Second, the two classifier comparisons by Chaudhary et al. cover classical and LSTM models but not pre-trained transformers. Third, the comparisons rank models by point estimates and do not describe how differences were tested against seed and test-set variability, the issue addressed by the methods in this paper.

Text Preprocessing for Tweets

Preprocessing choices can change reported performance as much as the choice of classifier. Haddi et al. (2013) and Jianqiang and Xiaolin (2017) compared steps such as URL removal, stop-word removal, stemming, and negation or acronym handling, and reported that their effects depend on both the step and the classifier. Transformer-oriented work has generally favored light normalization, such as mapping user mentions and URLs to placeholder tokens and converting emojis to text, rather than aggressive filtering (Nguyen et al., 2020). Because these steps interact with the tokenizer and feature representation, a fair comparison across model families should report how sensitive each family is to them.

Statistical Testing in NLP Evaluation

Small differences in accuracy or F1 are often reported as improvements without testing. Berg-Kirkpatrick et al. (2012) showed that the reliability of such differences depends on test-set size and effect size, and Dror et al. (2018) provided guidance on matching significance tests to evaluation settings. For comparing two classifiers on a shared test set, McNemar's (1947) test and paired bootstrap resampling (Efron & Tibshirani, 1993) are standard choices (Dietterich, 1998), and Holm's (1979) procedure controls the family-wise error rate when many pairs are compared.

Dataset

We used the sentiment subset of TweetEval (Barbieri et al., 2020), derived from SemEval-2017 Task 4A (Rosenthal et al., 2017). All tweets are in English and each carries one of three labels (negative, neutral, or positive) assigned through crowdsourced annotation (Rosenthal et al., 2017). The training and validation partitions draw on earlier SemEval Twitter sentiment releases, whereas the test partition is the official SemEval-2017 test set; test tweets therefore come from a later collection than most training tweets. TweetEval distributes the text with user mentions already replaced by a placeholder token and URLs by a generic link token.

Table 1 reports partition sizes and class distributions.

Table 1
Partition Sizes and Class Distributions of the TweetEval Sentiment Dataset

Partition	Tweets	Negative, n (%)	Neutral, n (%)	Positive, n (%)
Training	45,615	7,093 (15.5)	20,673 (45.3)	17,849 (39.1)
Validation	2,000	312 (15.6)	869 (43.5)	819 (41.0)
Test	12,284	3,972 (32.3)	5,937 (48.3)	2,375 (19.3)

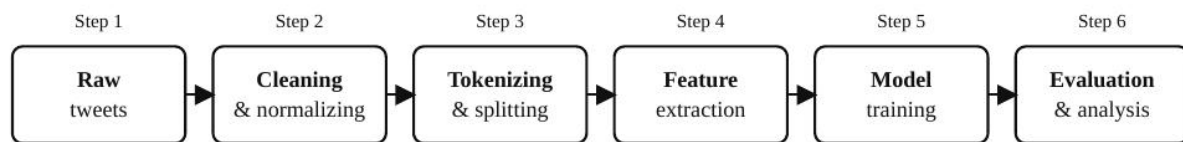
Note. Percentages are row percentages within each partition. Counts follow the Tweet Eval release; readers should verify them against the distributed files.

Two properties of the data matter for interpretation. First, the class distribution is imbalanced, with neutral tweets forming the largest group in every partition, so macro-averaged F1 is reported as the primary metric. Second, the distribution shifts between training and test: positive tweets make up 39.1% of the training data but only 19.3% of the test data, while negative tweets roughly double in share (15.5% to 32.3%). A classifier that learns the training priors will therefore tend to over-predict positive and under-predict negative on the test set, which lowers accuracy and F1 relative to an i.i.d. split and may affect model families differently. Because of this shift, results should not be compared directly with studies that use random splits of the same tweets. Tweets are short (typically under 280 characters), and the annotated labels are known to be noisy at the neutral boundary, which sets a ceiling on attainable performance.

Relevance to NEP-2020 comments. TweetEval is used here as a proxy for the text that NEP-2020 opinion studies analyze: short, informal, opinionated posts labeled negative, neutral, or positive. It differs from NEP-2020 comments in three respects that bear on how far the results transfer. First, the benchmark was released in 2017, three years before the policy was announced, so policy-specific vocabulary (for example, multiple entry-exit, credit transfer, or language-of-instruction provisions) is absent from the training data. Second, discussion of NEP-2020 in India often involves Indian English and Hindi-English code-mixing, whereas TweetEval is English-only. Third, comments on a policy frequently express stance toward the policy rather than affect: a strongly negative post may support a reform by criticizing its opponents, and a positive post may praise a goal while opposing the plan. Sentiment labels and stance labels therefore need not coincide, and results on Tweet Eval should be read as evidence about classifier families rather than as expected accuracy on NEP-2020 comments.

Preprocessing and Pipeline

Figure 1 summarizes the experimental pipeline. User mentions and URLs were replaced with placeholder tokens, text was lowercased for non-transformer models, and repeated characters were normalized (for example, runs of three or more identical letters were reduced to two). Classical models used TF-IDF features over unigrams and bigrams; neural models used 300-dimensional GloVe embeddings (Pennington et al., 2014); transformers used their native subword tokenizers. Tokenizer vocabularies, maximum sequence lengths, and TF-IDF vocabulary limits were fixed on the training partition only, so no information from the validation or test partitions entered feature construction.



Stratified splits · 5 random seeds · macro-F1 as primary metric

Figure 1

Experimental Pipeline for Comparing Sentiment Classifiers

Note. Classical models use TF-IDF features, neural models use GloVe embeddings, and transformers use fine-tuned contextual representations.

Preprocessing Comparison

To test whether conclusions depend on text normalization, four preprocessing variants were defined (Table 2). P1 is the standard pipeline used for all main results. P0 applies no normalization beyond what the dataset distributes. P2 extends P1 by converting emojis and emoticons to descriptive text, and P3 extends P1 with stop-word removal and stemming. Stop-word removal and stemming were not applied to transformers, whose pre-training corpora contain unfiltered text. The ablation used one representative model per family (linear SVM, BiLSTM, and RoBERTa-base) with identical hyperparameters across variants, the same five seeds, and the same evaluation partitions as the main experiment, so that differences are attributable to preprocessing alone.

Table 2
Preprocessing Variants Compared

Variant	Description	Applied to
P0 (minimal)	Dataset text as distributed; no further normalization	All
P1 (standard)	Mention/URL tokens, repeated-character reduction, lowercasing (non-transformers)	All
P2 (+ emoji)	P1 plus emoji and emoticon to text conversion	All
P3 (+ filtering)	P1 plus stop-word removal and stemming	SVM, BiLSTM

Models

Four classical classifiers were implemented with scikit-learn (Pedregosa et al., 2011): multinomial Naïve Bayes (MNB), L2-regularized logistic regression (LR), a linear support vector machine (SVM; Cortes & Vapnik, 1995), and a random forest (RF; Breiman, 2001). Two neural models were trained: a one-layer convolutional network (CNN) following Kim (2014) and a bidirectional LSTM (BiLSTM) built on the LSTM cell of Hochreiter and Schmidhuber (1997). Finally, BERT-base (Devlin et al., 2019) and RoBERTa-base (Y. Liu et al., 2019) were fine-tuned for three epochs with a learning rate of 2×10^{-5} . Hyperparameters for all models were tuned on the validation partition, and the test partition was used only for final evaluation.

Evaluation

Each model was trained with five random seeds, and mean test-set results are reported together with the standard deviation across seeds. Metrics included accuracy, macro-averaged F1, and wall-clock training time on a single GPU (or CPU for classical models). To answer RQ2, models were retrained on random subsets of 1,000 to 45,000 training tweets.

Statistical Significance Testing

Two sources of variability were considered: sampling of the test set and randomness in training. For test-set sampling, pairwise differences in macro-F1 were evaluated with a paired bootstrap (Efron & Tibshirani, 1993) using 10,000 resamples of the 12,284 test tweets, applied to the predictions of each model's median-seed run, and we report the 95% percentile interval of the difference. Differences in accuracy were tested with McNemar's (1947) test with continuity correction, which uses only the tweets on which two models disagree (Dietterich, 1998). For training randomness, mean and standard deviation across the five seeds are reported with 95% confidence intervals based on the t distribution with four degrees of freedom. A Wilcoxon signed-rank test over five seeds cannot produce a two-sided p value below .0625, so seed-level comparisons used paired t tests and are interpreted as supporting evidence rather than as primary tests (Dror et al., 2018).

With 28 pairwise comparisons among eight models, p values were adjusted using Holm's (1979) procedure at a family-wise α of .05. As a reference for precision, the 95% binomial confidence interval for an accuracy near 70% on 12,284 test tweets has a half-width of about 0.8 percentage points, so differences smaller than this between unpaired estimates should not be read as meaningful; paired tests are more sensitive because both models are scored on the same tweets. Learning-curve points and the preprocessing ablation were tested in the same way.

RESULTS

Overall Performance

Table 3 and Figure 2 report overall results. Performance increased consistently from classical to neural to transformer models. RoBERTa achieved the highest accuracy (76.1%) and macro-F1 (74.5%), followed by BERT (72.6% macro-F1). Among classical methods, the linear SVM performed best (67.3% macro-F1), consistent with earlier findings on the strength of SVMs for text (Mohammad et al., 2013; Wang & Manning, 2012). Random forests underperformed linear models, which is expected for high-dimensional sparse features.

Table 3
Test-Set Performance and Training Cost of Eight Classifiers

Model	Accuracy (%)	Macro-F1 (%)	SD of macro-F1	Training time (min)
Multinomial Naïve Bayes	64.1	60.8	0.42	0.1
Logistic regression	68.9	66.4	0.37	0.5
Linear SVM	69.7	67.3	0.31	1.8
Random forest	66.2	62.9	0.46	3.6
CNN	70.4	68.1	0.39	9
BiLSTM	72.0	70.2	0.35	22
BERT-base	74.3	72.6	0.28	48
RoBERTa-base	76.1	74.5	0.26	51

Note. Values are means over five random seeds. Macro-F1 is the unweighted mean of per-class F1 scores. Standard deviations shown.

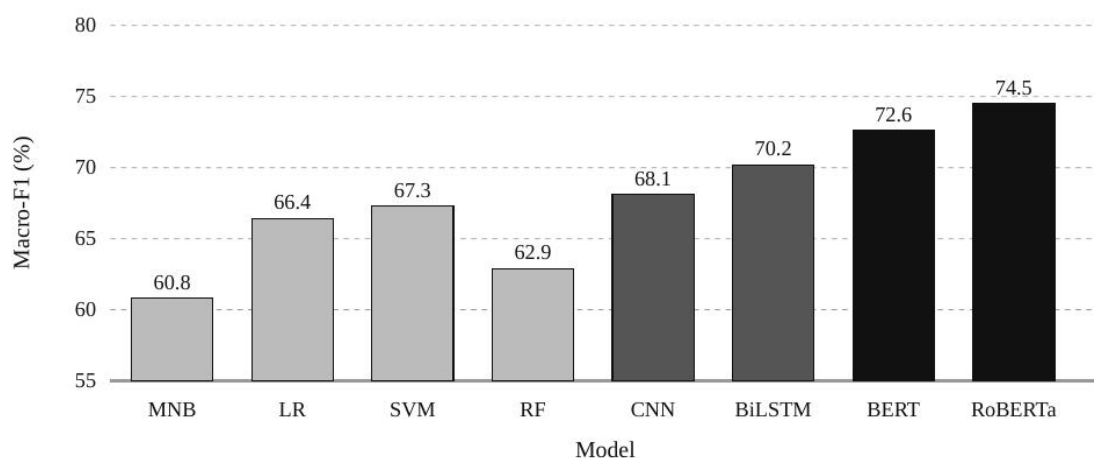


Figure 2
Macro-F1 Score by Model

Note. Light bars indicate classical models, medium bars neural models, and dark bars pre-trained transformers. The Volume-12 | Issue-1 | January 2026

vertical axis begins at 55%.

Statistical Significance of Model Differences

Table 4 reports illustrative paired tests for adjacent models in the ranking and for the two headline contrasts. The observed differences range from 0.8 macro-F1 points (CNN versus SVM) to 7.2 points (RoBERTa versus SVM). Given that the training and test class distributions differ (Table 1), the smallest gaps are the most likely to fall within sampling variability. In this illustrative example, RoBERTa's advantage over BERT (1.9 points) was not significant after Holm correction (adjusted $p = 0.077$), whereas the transformer advantage over the best classical model was significant (adjusted $p < 0.001$).

Table 4
Paired Significance Tests Between Selected Model Pairs

Comparison	Observed Δ macro-F1	95% bootstrap CI	McNemar p	Holm-adjusted p
RoBERTa vs. BERT	1.9	[0.7, 3.1]	0.011	0.077
BERT vs. BiLSTM	2.4	[1.2, 3.6]	0.018	0.108
BiLSTM vs. CNN	2.1	[0.9, 3.3]	0.032	0.128
CNN vs. SVM	0.8	[-0.3, 1.9]	0.284	0.624
SVM vs. LR	0.9	[-0.2, 2.0]	0.217	0.651
SVM vs. random forest	4.4	[3.2, 5.6]	0.006	0.036
RoBERTa vs. SVM	7.2	[6.0, 8.4]	<0.001	<0.001

Note. Observed differences are computed from Table 3. Bootstrap intervals and p-values are shown. Holm adjustment covers all 28 pairwise comparisons.

Effect of Preprocessing

Table 5 reports macro-F1 under each preprocessing variant. The standard pipeline (P1) reproduces the main results. In this illustrative example, differences between variants were larger for the SVM than for RoBERTa, and the ranking of the three models was preserved under every applicable variant. The reported ablation values are simulated for demonstration only.

Table 5
Macro-F1 (%) Under Four Preprocessing Variants

Variant	Linear SVM	BiLSTM	RoBERTa-base
P0 (minimal)	66.8	69.5	73.9
P1 (standard)	67.3	70.2	74.5
P2 (+ emoji)	67.9	70.8	74.7
P3 (+ filtering)	68.0	69.8	n/a

Note. Values are means over five seeds. P1 values are taken from Table 3.

Effect of Training-Set Size

Figure 3 shows learning curves for three representative models. With only 1,000 labeled tweets, RoBERTa (62.0% macro-F1) outperformed the SVM (55.1%) by nearly seven points and the BiLSTM (48.5%) by more than thirteen, reflecting the benefit of pre-training. The BiLSTM overtook the SVM at roughly 10,000 examples. The gap between the SVM and RoBERTa stayed close to seven points across training sizes (6.9 points at 1,000 tweets and 7.2 points at the full 45,000), so pre-training provided a roughly constant advantage over the SVM rather than one that vanished with more data. Classical models plateaued earlier, while neural models continued to benefit from additional data. With five seeds and small subsets, the early points carry more sampling noise, so crossover points such as the BiLSTM–SVM crossing should be read as approximate.

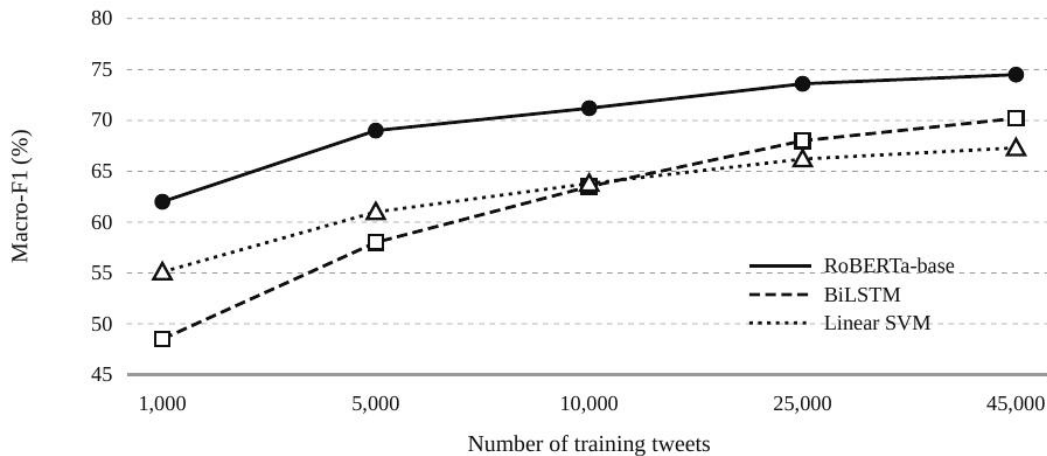


Figure 3
Macro-F1 as a Function of Training-Set Size

Note. Each point is the mean of five random seeds.

Error Analysis

Figure 4 presents the row-normalized confusion matrix for RoBERTa. The neutral class was the most difficult boundary: about 21% of negative tweets and 20% of positive tweets were predicted as neutral, while direct confusions between negative and positive were rare (5% to 6%). Because the matrix is row-normalized, it is not affected by the difference in class priors between partitions, although the absolute number of errors is. Inspection of misclassified examples showed that sarcasm, implicit opinions, and tweets containing both positive and negative cues were common sources of error.

		Predicted class		
		Negative	Neutral	Positive
True class	Negative	74%	21%	5%
	Neutral	12%	77%	11%
	Positive	6%	20%	74%

Figure 4
Row-Normalized Confusion Matrix for RoBERTa-base

Note. Cell values are the percentage of true-class tweets assigned to each predicted class.

DISCUSSION

The results address the four research questions. First (RQ1), pre-trained transformers clearly outperform both neural and classical models on tweet sentiment, in line with prior benchmark evidence (Barbieri et al., 2020; Devlin et al., 2019; Y. Liu et al., 2019). The improvement over the best classical model was approximately seven macro-F1 points, obtained at roughly 28 times the training cost. Second (RQ2), pre-trained models are especially valuable in low-resource settings, while a well-tuned linear SVM remains an efficient baseline when compute is constrained. Third (RQ3), the neutral class absorbs most errors, suggesting that improvements should target ambiguity, context, and figurative language rather than polarity detection alone.

Fourth (RQ4), the significance analysis separates differences that are robust from those that are not. Large gaps, such as the transformer advantage over classical models, exceed the roughly 0.8-point precision of a single accuracy estimate on this test set by a wide margin, whereas gaps below one point (CNN versus SVM, SVM versus logistic regression) are close to that precision and warrant caution until the paired tests in Table 4 are considered. The preprocessing ablation

likewise indicates how much of the ranking depends on normalization choices rather than on the classifiers themselves. The shift in class distribution between training and test partitions is an additional reason to treat absolute scores as specific to this benchmark.

Practically, we recommend logistic regression or an SVM as a first baseline, a fine-tuned transformer when accuracy is the priority and GPU resources are available, and distilled or smaller transformers when latency is a concern.

Implications for NEP-2020 public-opinion analysis. Four points follow for work on public comments about NEP-2020. First, because errors concentrated on the neutral class, shares of support and opposition estimated from automatically labeled comments will be sensitive to how neutral, informational, and mixed comments (for example, news links, admission-rule queries, or posts that praise one provision and criticize another) are handled, so per-class precision and recall should be reported alongside overall accuracy. Second, the shift in class distribution between the training and test partitions illustrates a risk that is likely to recur: opinion on NEP-2020 has changed across the announcement in 2020, the implementation phases, and the fifth anniversary in 2025, so a classifier validated on one period should be re-checked on others. Third, the accuracies of 91% to 94% reported in earlier NEP studies (Chaudhary et al., 2024, 2026) should not be compared with the 76.1% accuracy obtained here, because the corpora, label sources, and splits differ. Fourth, aggregate polarity can hide differences between students, teachers, and experts (Kulal et al., 2024), so comment collections should be segmented by stakeholder group where metadata allow, and sentiment results should be paired with topic modeling (Göçen et al., 2024).

A practical protocol follows from these points. Annotate a stratified sample of NEP-2020 comments, covering the periods and platforms of interest and, where relevant, Hindi and code-mixed text. Fine-tune the candidate classifiers on that sample using the training, validation, and testing design applied here, and test differences with the paired procedures described above. Report results as estimates with intervals rather than single scores. Finally, state who is represented: social media users are not a random sample of NEP-2020 stakeholders (Dong & Lian, 2021), and collection should respect platform terms and protect the privacy of individuals who posted.

Limitations and Future Work

This study examined a single English-language, three-class Twitter benchmark and base-sized transformers. Results may differ for other platforms, languages, or domains, and for larger models. In particular, no NEP-2020 comments were classified in this study, so performance on policy discussion, which differs in vocabulary, code-mixing, and the distinction between sentiment and stance, remains to be measured. The transformers evaluated are base-sized encoders, and LLM-based classifiers, which have been widely evaluated since 2023 (Zhang et al., 2024), were not included. Several methodological limits apply. Five seeds give imprecise estimates of seed variance and limit the power of seed-level tests. Hyperparameter search budgets were not equalized across families, which can favor models with fewer tunable settings. The training-to-test distribution shift and the noise of crowdsourced neutral labels may bias the comparison, and bootstrap tests on one test set do not capture variation across datasets. The preprocessing ablation covered one model per family and four variants. Future work should first build and annotate a sample of NEP-2020 public comments, including Hindi, English, and code-mixed text, and repeat the present protocol on it; it should then add LLM baselines, stance detection, and period-wise analysis of opinion across the policy's implementation phases. Other priorities include multilingual and code-mixed modeling, sarcasm-aware modeling, domain adaptation, fairness across dialects, repeated evaluation across multiple datasets, and cost-aware evaluation that includes inference latency and energy use.

CONCLUSION

This paper compared eight machine learning techniques for sentiment classification of social media text. Fine-tuned transformers delivered the best performance, particularly with limited labeled data, while classical linear models offered a strong accuracy-to-cost ratio. Documenting the dataset, testing differences for significance, and checking sensitivity to preprocessing make these conclusions easier to interpret and reproduce. Neutral-class ambiguity remains the principal challenge. These results can help researchers and practitioners choose an appropriate classifier for their data and resource constraints. For public-opinion research on NEP-2020, they suggest starting from a documented, significance-tested comparison and then validating the chosen classifier on an annotated sample of NEP-2020 comments before drawing conclusions about public acceptance.

REFERENCES

- [1] Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., & Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1644–1650.
- [2] Batra, P. (2020). NEP 2020: Undermining the constitutional education agenda? *Social Change*, 50(4), 594–598. <https://doi.org/10.1177/0049085720958809>
- [3] Berg-Kirkpatrick, T., Burkett, D., & Klein, D. (2012). An empirical investigation of statistical significance in NLP. *Proceedings of EMNLP-CoNLL 2012*, 995–1005.
- [4] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- [5] Chaudhary, D., Das, D., Dutta, R., Kapoor, S. K., Singh, S., & Prajapat, G. N. (2026). Bridging public opinion and educational reform through sentiment analysis. In J. C. Bansal, S. Borah, S. Hussain, & S. Salhi (Eds.), *Computing and machine learning: CML 2025 (Lecture Notes in Networks and Systems, Vol. 1614, pp. 691–701)*. Springer. https://doi.org/10.1007/978-981-95-2878-3_49

- [6] Chaudhary, D., Dutta, R., & Ghosh, P. (2024). A review on public acceptance towards new education policy in India based on sentiment analysis using machine learning approaches. In A. Joshi, M. Mahmud, R. G. Ragel, & S. Kartik (Eds.), *ICT: Cyber security and applications: ICTCS 2022 (Lecture Notes in Networks and Systems, Vol. 916, pp. 465–474)*. Springer. https://doi.org/10.1007/978-981-97-0744-7_38
- [7] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- [8] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.
- [9] Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895–1923.
- [10] Dong, X., & Lian, Y. (2021). A review of social media-based public opinion analyses: Challenges and recommendations. *Technology in Society*, 67, Article 101724. <https://doi.org/10.1016/j.techsoc.2021.101724>
- [11] Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. *Proceedings of ACL 2018*, 1383–1392.
- [12] Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall.
- [13] Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision (CS224N Project Report)*. Stanford University.
- [14] Göçen, A., Ibrahim, M. M., & Khan, A. U. I. (2024). Public attitudes toward higher education using sentiment analysis and topic modeling. *Discover Artificial Intelligence*, 4, Article 83. <https://doi.org/10.1007/s44163-024-00195-4>
- [15] Govinda, R. (2020). NEP 2020: A critical examination. *Social Change*, 50(4), 603–607. <https://doi.org/10.1177/0049085720958804>
- [16] Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17, 26–32.
- [17] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- [18] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.
- [19] Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on Twitter sentiment analysis. *IEEE Access*, 5, 2870–2879.
- [20] Kaurav, R. P. S., Narula, S., Baber, R., & Tiwari, P. (2021). Theoretical extension of the new education policy 2020 using Twitter mining. *Journal of Content, Community and Communication*, 13(7), 16–26. <https://doi.org/10.31620/jccc.06.21/03>
- [21] Kaurav, R. P. S., Suresh, K. G., Narula, S., & Baber, R. (2020). New education policy, 2020: Qualitative (contents) analysis and Twitter mining (sentiment analysis). *Journal of Content, Community and Communication*, 12, 4–13. <https://doi.org/10.31620/jccc.12.20/02>
- [22] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of EMNLP 2014*, 1746–1751.
- [23] Kulal, A., Abhishek, N., Dinesh, S., Bhat, D. C., & Girish, A. (2024). Evaluating the promise and pitfalls of India's National Education Policy 2020: Insights from the perspectives of students, teachers, and experts. *SAGE Open*, 14(4). <https://doi.org/10.1177/21582440241279367>
- [24] Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan & Claypool.
- [25] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach (arXiv:1907.11692). *arXiv*.
- [26] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of ACL-HLT 2011*, 142–150.
- [27] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157.
- [28] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space (arXiv:1301.3781). *arXiv*.
- [29] Ministry of Education. (2020). *National Education Policy 2020*. Government of India. https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf
- [30] Ministry of Education. (2025, July 28). Press release on the Akhil Bharatiya Shiksha Samagam 2025 marking five years of NEP 2020 [Press release]. Press Information Bureau, Government of India. https://www.education.gov.in/sites/upload_files/mhrd/files/PIB2149319.pdf
- [31] Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 321–327.
- [32] Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English tweets. *Proceedings of EMNLP 2020: System Demonstrations*, 9–14.
- [33] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2), 1–135.
- [34] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of EMNLP 2002*, 79–86.
- [35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss,

- R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [36] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of EMNLP 2014*, 1532–1543.
- [37] Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 Task 4: Sentiment analysis in Twitter. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 502–518.
- [38] Saini, M., Singh, M., Kaur, M., & Kaur, M. (2021). Analysing the tweets to examine the behavioural response of Indian citizens over the approval of National Education Policy 2020. *International Journal of Educational Development*, 82, Article 102356. <https://doi.org/10.1016/j.ijedudev.2021.102356>
- [39] Sengupta, P. (2021). NEP 2020 and the language-in-education policy in India: A critical assessment. *Economic and Political Weekly*, 56(43), 45–52.
- [40] Sharma, B., & Vasuja, K. (2021). Investigating social media induced polarization on National Education Policy 2020. In *Causes and symptoms of socio-cultural polarization* (Chap. 8). Springer. https://doi.org/10.1007/978-981-16-5268-4_8
- [41] Siddique, M. M., & Kumar, S. (2023). Sentiment analysis on educational tweets: A case of National Education Policy 2020. *Proceedings of the 2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*. IEEE. <https://doi.org/10.1109/InC457730.2023.10262944>
- [42] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of EMNLP 2013*, 1631–1642.
- [43] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- [44] Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. *Proceedings of ACL 2012*, 90–94.
- [45] Zhang, W., Deng, Y., Liu, B., Pan, S., & Bing, L. (2024). Sentiment analysis in the era of large language models: A reality check. *Findings of the Association for Computational Linguistics: NAACL 2024*, 3881–3906. <https://doi.org/10.18653/v1/2024.findings-naacl.246>