International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(15s) 2026 pp. 431-446
https://doi.org/10.70917/ijcisim-2026-4427
Article

An Enhanced Hybrid BERT–BiLSTM–GRU Model for Text-Based Emotion Detection

Rakesh Sahani[1]*, Firos A[2]

[1,2] Department of Computer Science & Engineering, Rajiv Gandhi University, Doimukh, India–791112
[1]rakesh.sahani@rgu.ac.in [2]firos.a@rgu.ac.in

Abstract: Fine-grained emotion detection in user-generated text remains a challenging task because of severe class imbalance and the semantic overlap between related affective categories. This paper investigates whether stacking recurrent layers on top of a pre-trained transformer encoder can improve text-based emotion classification, and under what architectural conditions such stacking is beneficial. We propose an enhanced hybrid model that augments BERT-base with a Bidirectional Long Short-Term Memory (BiLSTM) layer followed by a Bidirectional Gated Recurrent Unit (BiGRU), with classification performed from the [CLS] representation. The model is evaluated on the GoEmotions corpus mapped to the six basic Ekman emotions, and is compared against twelve baselines spanning classical, recurrent, transformer, and hybrid architectures using three random seeds, McNemar's significance tests, and mean ± standard deviation reporting. The proposed model attains the highest mean macro-F1 (0.7233 ± 0.0042), exceeding BERT-base by 0.93 percentage points and matching RoBERTa-base, which uses substantially more pre-training data, despite a 9.5% smaller parameter budget. Architectural ablation shows that all three design choices—the BiLSTM layer, the BiGRU layer, and CLS-based pooling—are individually necessary: removing any one of them reduces macro-F1. Per-class analysis reveals heterogeneous trade-offs, with significant gains on the majority class and on surprise but a regression on sadness relative to several transformer baselines. The model also incurs a 2.7× single-sample inference latency penalty over BERT-base, which is largely amortised at batch inference. The findings clarify when and why recurrent stacking on top of transformers is useful, and provide an honest, fully reproducible empirical baseline for future work on transformer–RNN hybrids in emotion classification.

Keywords: Emotion Detection, Text Classification, BERT, BiLSTM, GRU, Hybrid Models, GoEmotions, Ekman Emotions, Transformer Fine-Tuning, Natural Language Processing

1. Introduction

Automatic emotion detection from text is a foundational problem in affective computing and natural language processing, with applications spanning mental-health screening, customer-experience analytics, content moderation, and human–computer interaction [1], [2]. Unlike coarse sentiment analysis, fine-grained emotion classification requires distinguishing among semantically adjacent affective states such as anger and disgust, or sadness and disappointment, which often share lexical and syntactic cues.

The introduction of large pre-trained transformer encoders [3], [4], [5] has substantially advanced the state of the art on text-classification benchmarks, including emotion-labelled corpora. At the same time, an extensive body of work has demonstrated that recurrent architectures—LSTM [6] and GRU [7], particularly in their bidirectional forms—remain competitive on sequence-classification tasks where local order information is informative. A natural research question, therefore, is whether stacking recurrent layers on top of a transformer encoder can yield further improvements, and under what conditions this is the case. Several recent works have proposed BERT–BiLSTM, BERT–GRU, and BERT–BiLSTM–GRU hybrids for emotion classification [8], [9], [10], typically reporting modest gains over plain transformer baselines.

Despite these reports, the empirical picture is unclear for three reasons. First, many published comparisons rely on a single random seed, making it difficult to distinguish architectural improvements from initialisation noise [11]. Second, the choice of pooling strategy on top of the recurrent stack—whether to use the [CLS] token, mean pooling,

max pooling, or a combination—is rarely treated as a controlled variable, even though pooling is known to interact strongly with how the encoder's pre-trained representations are used [12]. Third, accuracy and macro-F1 can diverge under class imbalance, so a model that improves macro-F1 may not improve overall accuracy, and vice versa; without per-class analysis, the locus of improvement remains opaque.

This paper addresses these gaps by conducting a large, controlled empirical study of transformer–RNN hybrids for fine-grained emotion classification on the GoEmotions corpus [13], mapped to the six basic Ekman emotions [14]. Our central architectural proposal is an enhanced hybrid model that stacks a bidirectional LSTM and a bidirectional GRU on top of a fully fine-tuned BERT-base encoder, with classification performed from the [CLS] representation rather than from a mean+max pool of the recurrent outputs. We compare this model against twelve baselines drawn from four families—classical, recurrent, transformer, and hybrid—and report mean and standard deviation across three random seeds, McNemar's tests with Bonferroni correction at both the overall and per-class levels, and parameter, training-time, and inference-latency efficiency measurements.

The contributions of this paper are as follows.

(1) We propose an enhanced hybrid BERT–BiLSTM–BiGRU architecture for text-based emotion detection that achieves the highest mean macro-F1 (0.7233 ± 0.0042) among thirteen compared model variants on GoEmotions Ekman-6, statistically tied with the larger RoBERTa-base model (macro-F1 0.7226 ± 0.0024) while using approximately 9.5% fewer trainable parameters.
(2) We provide a complete architectural ablation that isolates the contribution of each component. Removing the BiGRU layer reduces macro-F1 to 0.7117; removing the BiLSTM layer reduces it to 0.7147; and replacing CLS-based pooling with the more common mean+max pool reduces it to 0.7036, below the BERT-base baseline. All three design choices are individually necessary.
(3) We conduct a per-class significance analysis using McNemar's test that reveals heterogeneous architectural effects across emotion classes. The proposed model significantly improves recall on surprise relative to BERT-base, and on the majority class joy relative to several baselines, but exhibits a statistically significant regression on sadness relative to RoBERTa-base, DistilBERT, and the BERT–BiGRU ablation. We discuss these trade-offs explicitly rather than reporting only aggregate metrics.
(4) We report a full efficiency profile—trainable parameters, training time per epoch, and median inference latency at batch sizes 1 and 32—showing that the proposed model incurs a 2.7× single-sample latency penalty over BERT-base, but only a marginal latency increase at batch inference. This frames the model as well suited to batch-oriented deployments and identifies single-sample inference as the principal limitation.
(5) We adopt a strict reproducibility protocol—frozen test indices, fixed seed sets, deterministic mode in cuDNN and PyTorch, and assertion-based verification that the held-out test labels match a master reference at the end of every run. All thirty-nine main runs (thirteen model variants × three seeds) and the additional hyperparameter-investigation run satisfy these checks.

The remainder of the paper is organised as follows. Section 2 reviews related work on emotion detection and transformer–RNN hybrid architectures. Section 3 describes the dataset and the Ekman-6 mapping procedure. Section 4 details the proposed architecture and the twelve baselines. Section 5 specifies the experimental protocol. Section 6 reports the results and statistical analyses. Section 7 discusses architectural implications and limitations. Section 8 concludes.

2. Related Work

2.1 Emotion Detection in Text

Text-based emotion detection has been studied for over two decades, with early systems relying on lexicons of affect-laden words [1]. Modern approaches treat the task as supervised classification using a labelled corpus and either feature-engineered classifiers or learned representations. Recent surveys have catalogued the rapid progress of deep-learning approaches and the persistent challenges of class imbalance, label ambiguity, and dataset noise [24], [26], [27]. Fine-grained emotion taxonomies—commonly Ekman's six basic emotions [14] or Plutchik's eight [15]—have largely supplanted binary positive/negative sentiment formulations in research, although the choice of label space materially affects difficulty: more classes increase the chance of inter-class confusion and amplify the impact of class imbalance.

The GoEmotions corpus [13] provides one of the largest publicly available fine-grained emotion datasets, comprising 58,009 Reddit comments annotated with 27 emotion categories plus a neutral label. Demszky et al. [13]

established baselines using BERT-base and reported a macro-F1 of approximately 0.46 on the full 27-emotion task and substantially higher scores on coarser groupings, including the Ekman-6 mapping used in the present study.

2.2 Recurrent Models for Sequence Classification

LSTM [6] and GRU [7] units, especially in bidirectional configurations, have long served as strong baselines for text classification. Attention mechanisms [16], [17] further improved their ability to focus on salient tokens. Yang et al. [18] showed that hierarchical attention over BiGRU encodings could outperform earlier recurrent classifiers on document-level tasks. Despite the rise of transformers, recurrent layers remain useful as feature aggregators because their inductive bias differs from self-attention in ways that can be complementary.

2.3 Pre-trained Transformer Encoders

BERT [3] introduced a bidirectional masked-language-model pre-training objective and a [CLS]-based fine-tuning protocol that has since become a dominant paradigm. RoBERTa [4] removed the next-sentence-prediction objective and trained on substantially more data, yielding consistent improvements. DistilBERT [5] applied knowledge distillation to obtain a 40% smaller model that retains approximately 97% of BERT's downstream performance. All three are widely used as baselines and as backbones for hybrid architectures.

2.4 Transformer–RNN Hybrid Architectures

Several studies have proposed augmenting transformer encoders with recurrent layers. Pota et al. [19] reported gains on Italian emotion classification using a BERT–BiLSTM hybrid. Acheampong et al. [8] proposed a two-stage architecture that feeds BERT outputs into a BiLSTM classifier for emotion detection on the ISEAR corpus, reporting modest gains over single-model transformer baselines. Tan et al. [9] introduced a RoBERTa–LSTM hybrid for sentiment analysis on imbalanced Twitter datasets, combining data augmentation and oversampling with the recurrent stack to address class imbalance and lexical sparsity. More recently, Rahman et al. [28] proposed a context-aware RoBERTa–BiLSTM hybrid for sentiment analysis on multiple benchmarks, while Jahin et al. [29] combined RoBERTa with attention-augmented BiLSTM for interpretable Twitter sentiment analysis. Talaat [10] proposed hybrid stacks combining BERT and DistilBERT/RoBERTa with BiLSTM and BiGRU layers for Twitter sentiment classification, reporting modest accuracy gains over the plain transformer baselines. As with most hybrid-stack reports in this literature, the experiments use a single random seed and do not isolate the effect of pooling strategy from the recurrent layer choice — a gap that the present work addresses.

The present work differs from prior hybrid studies in three respects. First, our architectural ablation isolates the individual contribution of each recurrent layer and of the pooling strategy—two factors that are typically conflated. Second, we report results across three random seeds with formal significance testing, both at the overall level and per emotion class. Third, we situate the architectural comparison within an efficiency frame that includes parameter counts, training time, and inference latency.

2.5 Reproducibility and Evaluation Practices

Recent work has highlighted methodological concerns in NLP evaluation, including single-seed reporting [11], inadequate statistical testing [20], and dataset artefacts that inflate reported gains. McNemar's test [21] is a standard tool for paired-sample comparison of classifiers and is applicable to per-sample correctness vectors derived from a held-out test set. We adopt these practices throughout.

3. Dataset

3.1 GoEmotions

We use the GoEmotions corpus [13], which contains 58,009 English Reddit comments labelled with one or more of 27 emotion categories plus a neutral category. The corpus is publicly distributed via the Hugging Face datasets library [22] and is split into 43,410 training, 5,426 validation, and 5,427 test examples. Annotation was performed by trained workers, and the dataset has been widely used as a benchmark for fine-grained emotion classification, alongside earlier benchmarks such as SemEval-2018 Task 1 [25] which targets affect intensity and valence on Twitter.

3.2 Ekman-6 Mapping

Following the mapping released with GoEmotions [13], we project the 27 emotion labels onto the six basic Ekman emotions [14]: anger, disgust, fear, joy, sadness, and surprise. The mapping is many-to-one and is documented in the dataset card. We adopt the following preprocessing decisions explicitly:

(i) Neutral examples are removed. We treat the task as six-way Ekman classification rather than seven-way, on the grounds that 'neutral' lacks a coherent emotional content and substantially dilutes per-class signal in the minority categories.

(ii) Multi-label examples that span more than one Ekman category after the mapping are removed, converting the task to single-label classification. This is consistent with prior work that uses GoEmotions as a single-label benchmark and prevents ambiguous gradient signal during training.

After preprocessing, the dataset comprises 28,104 training examples, 3,527 validation examples, and 3,539 test examples. From the original splits, 16,021 examples were dropped for being neutral-only, and a further 3,072 were dropped for being multi-Ekman. The resulting class distribution is reported in Table 1; it is heavily skewed toward joy, with fear and disgust each accounting for fewer than 600 training examples.

Table 1. Class distribution after Ekman-6 mapping with neutral and multi-label examples removed.

Split	Anger	Disgust	Fear	Joy	Sadness	Surprise
Train	4,590	524	566	15,809	2,455	4,160
Validation	596	62	77	2,017	278	497
Test	609	77	85	1,937	298	533

The class imbalance is severe: joy outnumbers disgust by approximately 30:1 in the training set. This imbalance is preserved across all splits and substantially shapes the difficulty profile of the task. We do not apply class re-weighting, oversampling, or focal loss in the main experiments, in order to maintain comparability with prior work that uses the same dataset under standard cross-entropy training.

4. Methodology

4.1 Proposed Architecture

The proposed model, denoted *BERT–BiLSTM–BiGRU (CLS)*, consists of four components arranged sequentially:

(1) A pre-trained BERT-base encoder (bert-base-uncased, 109.5M parameters [3]) that maps tokenised input to a sequence of contextualised representations of dimension 768.

(2) A bidirectional LSTM with hidden size 256 per direction (512 total), one layer, applied to the BERT output sequence. The BiLSTM provides an explicit recurrent modelling stage on top of the self-attention representations.

(3) A bidirectional GRU with hidden size 256 per direction (512 total), one layer, stacked on the BiLSTM output. The two recurrent stages are intended to capture complementary aspects of sequential structure.

(4) A classification head consisting of dropout ($p = 0.3$) followed by a linear projection to six emotion classes. Crucially, the head receives the representation at the [CLS] position—that is, the first time step of the BiGRU output—rather than a pooled aggregate over the whole sequence. As we show in Section 6.2, this pooling choice is essential.

The model has 112,769,286 trainable parameters, approximately 3% more than plain BERT-base.

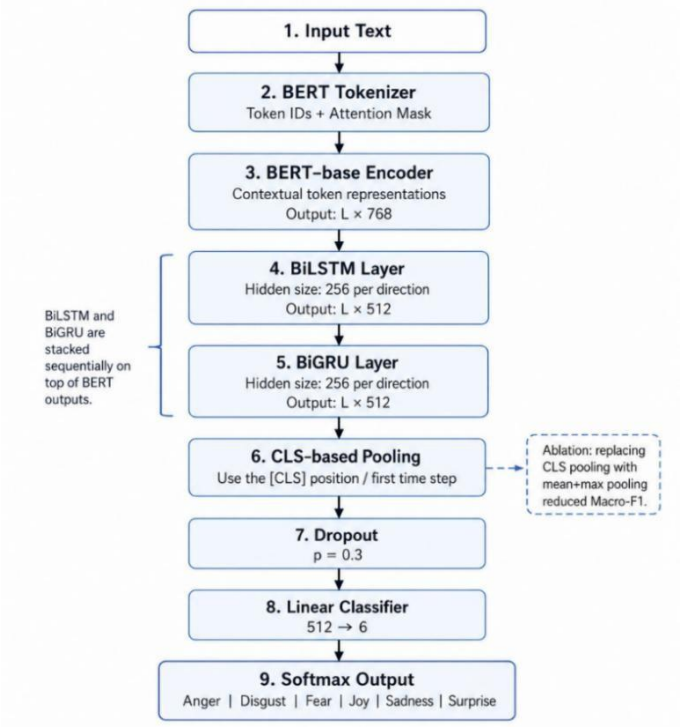

Fig. 1. Architecture of the proposed BERT+BiLSTM+BiGRU (CLS) model. Input tokens are embedded by a pre-trained BERT-base encoder, producing a sequence of 768-dimensional contextualised representations. These are processed by a bidirectional LSTM (hidden size 256 per direction), then by a bidirectional GRU (hidden size 256 per direction). The representation at the [CLS] position of the BiGRU output is passed through dropout (p = 0.3) and a linear classifier to produce probabilities over the six Ekman emotion classes.

4.2 Training Objective and Optimisation

All transformer-based models are trained with cross-entropy loss using the AdamW optimiser [23]. The proposed model and its hybrid variants use two parameter groups with distinct learning rates: 2×10^{-5} for the BERT encoder and 1×10^{-3} for the recurrent and head parameters. This separation reflects the empirical observation that randomly initialised heads benefit from a larger learning rate than that suitable for a pre-trained encoder; using a single small learning rate slows head learning, while a single large rate damages BERT's pre-trained weights.

Other training hyperparameters are: batch size 16, weight decay 0.01, linear learning-rate schedule with 10% warmup, gradient norm clipping at 1.0, and mixed-precision training (fp16). Maximum sequence length is 128 tokens. We train for up to five epochs with early stopping on validation macro-F1 (patience 3). The classical and recurrent baselines use task-appropriate optimisers and schedules described in Section 4.4.

4.3 Pooling Strategy

We use the [CLS] representation as the input to the classification head. The alternative—mean+max pooling over time, with the mean and max vectors concatenated—is a common choice in published BERT–RNN hybrids and was our initial choice. As reported in Section 6.2, this yields a macro-F1 of 0.7036, below plain BERT-base. Substituting [CLS] pooling raises macro-F1 to 0.7233 with no other change to the recurrent layers; note that [CLS] pooling yields a 512-dimensional head input whereas mean+max pooling yields a 1,024-dimensional input (512 mean + 512 max concatenated), so the linear classifier input dimension also changes between variants. We treat this pooling sensitivity as a primary methodological finding.

4.4 Baselines

We compare the proposed model against twelve baselines spanning four families.

Classical baseline. TF-IDF features over unigrams and bigrams (min_df = 2, sublinear TF, English lowercasing) followed by a linear support-vector classifier (LinearSVC, C = 1.0). This serves as a non-neural reference.

Recurrent baselines. Five word-level recurrent classifiers, each with a 200-dimensional embedding layer held at its random initialisation and not updated during training, vocabulary size 10,055, and dropout 0.3 on the embedding and head: (i) unidirectional LSTM, (ii) unidirectional GRU, (iii) BiLSTM, (iv) BiLSTM with additive (Bahdanau-style) attention, and (v) CNN-LSTM with parallel 1D convolutions of kernel sizes 3, 4, and 5 followed by an LSTM. All recurrent baselines use Adam with learning rate 1
$\times 10^{-3}$, batch size 64, and up to 15 epochs with early stopping.

Transformer baselines. Three plain pre-trained encoders fine-tuned with a linear classification head: BERT-base [3] (single LR 2×10^{-5}), RoBERTa-base [4] (single LR 1×10^{-5}, lower per the standard recommendation), and DistilBERT-base [5] (single LR 5×10^{-5}, batch size 32 to fit the smaller model).

Hybrid ablations. Two component ablations of the proposed model: BERT–BiLSTM (CLS), which removes the BiGRU stage, and BERT–BiGRU (CLS), which removes the BiLSTM stage. Both use [CLS] pooling and the same two-LR-group training recipe as the proposed model. We also include BERT–BiLSTM–BiGRU (mean+max), which retains both recurrent layers but replaces [CLS] pooling with the more common mean+max pool, to isolate the effect of pooling.

5. Experimental Protocol

5.1 Reproducibility

All experiments use three random seeds {42, 1337, 2024}. For each (model, seed) pair, we set Python, NumPy, PyTorch CPU, PyTorch CUDA, and Hugging Face transformers seeds; we enable cuDNN deterministic mode (cudnn.benchmark = False) and request deterministic algorithms (warn-only). The test-set indices and labels are computed once by a deterministic preprocessing script and frozen as a NumPy array; the training script asserts equality of test labels against this frozen reference at the end of every run, which guards against any silent non-determinism in the data pipeline. All thirty-nine main runs (thirteen model variants × three seeds) and the additional hyperparameter-investigation run pass this assertion.

5.2 Evaluation Metrics

We report accuracy, macro precision, macro recall, and macro F1, with mean and standard deviation across three seeds. We additionally report per-class F1 to expose heterogeneous architectural effects. Statistical comparisons use McNemar's test [21] with continuity correction, applied to paired correctness vectors over the test set. To increase test power and reduce seed-luck artefacts, we pool correctness vectors across the three seeds for each comparison, yielding $3 \times |\text{test}| = 10{,}617$ paired observations per test. We apply Bonferroni correction across the family of comparisons (twelve baseline tests for the overall analysis; thirty per-class tests for the per-class analysis).

5.3 Efficiency Measurements

Inference latency is measured on a single NVIDIA RTX 4050 Laptop GPU (6 GB VRAM, CUDA 12.1, cuDNN 9). For each model, we use the seed-42 best checkpoint and run 200 warm-up forward passes followed by 1,000 timed forward passes at batch sizes 1 and 32. We report the median milliseconds per sample. CUDA synchronisation is performed before and after each timed call. cuDNN benchmark mode is left disabled for reproducibility, which applies a uniform pessimistic bias across all models. Training time is reported as mean ± standard deviation of seconds per epoch across all observed epochs and seeds.

5.4 Hardware and Software

All experiments are conducted on a single NVIDIA RTX 4050 Laptop GPU with 6 GB VRAM. The software stack is Python 3.11, PyTorch 2.5.1+cu121, transformers 4.38–4.45, datasets $\geq$ 2.16, scikit-learn $\geq$ 1.3, and statsmodels $\geq$ 0.14.

6. Results

6.1 Main Results

Table 2 reports overall metrics for all thirteen model variants, grouped by family (classical, recurrent, transformer, hybrid). The proposed BERT–BiLSTM–BiGRU (CLS) model attains the highest mean macro-F1 of

0.7233 ± 0.0042, narrowly exceeding RoBERTa-base (0.7226 ± 0.0024) and improving on BERT-base by 0.93 percentage points. The pooling-ablation variant—the same architecture with mean+max pooling instead of CLS pooling—falls to 0.7036, below all six other transformer-family models, including plain BERT-base. The non-transformer baselines (classical and recurrent) form a clearly separated tier at macro-F1 between 0.6182 and 0.6518, approximately 6–10 percentage points below the transformer family.

Table 2. Main results on GoEmotions Ekman-6 test set, grouped by model family (classical, recurrent, transformer, hybrid). Mean ± standard deviation over three random seeds. Bold row indicates the proposed model. Significance markers indicate McNemar's test against the proposed model with Bonferroni correction across twelve comparisons (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$).

Model	**Accuracy**	**Macro-P**	**Macro-R**	**Macro-F1**
TF-IDF + LinearSVC ***	0.7318 ± 0.0000	0.6992 ± 0.0000	0.5683 ± 0.0000	0.6182 ± 0.0000
LSTM ***	0.7556 ± 0.0035	0.6903 ± 0.0068	0.5981 ± 0.0112	0.6355 ± 0.0055
GRU ***	0.7560 ± 0.0056	0.6814 ± 0.0145	0.6182 ± 0.0124	0.6450 ± 0.0095
BiLSTM ***	0.7443 ± 0.0022	0.6766 ± 0.0022	0.5895 ± 0.0183	0.6235 ± 0.0126
BiLSTM + Attention ***	0.7642 ± 0.0038	0.6926 ± 0.0102	0.6220 ± 0.0038	0.6518 ± 0.0047
CNN-LSTM ***	0.7521 ± 0.0036	0.6631 ± 0.0086	0.6085 ± 0.0033	0.6311 ± 0.0031
BERT-base	0.8192 ± 0.0038	0.7450 ± 0.0215	0.6978 ± 0.0129	0.7140 ± 0.0043
RoBERTa-base	0.8226 ± 0.0007	0.7316 ± 0.0096	0.7161 ± 0.0061	0.7226 ± 0.0024
DistilBERT-base	0.8146 ± 0.0017	0.7288 ± 0.0064	0.7022 ± 0.0059	0.7109 ± 0.0035
BERT + BiLSTM (CLS)	0.8147 ± 0.0011	0.7185 ± 0.0072	0.7102 ± 0.0083	0.7117 ± 0.0043
BERT + BiGRU (CLS)	0.8192 ± 0.0009	0.7331 ± 0.0150	0.7021 ± 0.0093	0.7147 ± 0.0048
BERT + BiLSTM + BiGRU (CLS) [proposed]	**0.8209 ± 0.0012**	**0.7524 ± 0.0081**	**0.7036 ± 0.0014**	**0.7233 ± 0.0042**
BERT+BiLSTM+BiGRU (mean+max) *	0.8119 ± 0.0049	0.7283 ± 0.0099	0.6964 ± 0.0043	0.7036 ± 0.0032

McNemar's test on aggregated test predictions across the three seeds shows that the proposed model significantly outperforms all six non-transformer baselines ($p < 0.001$ in every case after Bonferroni correction across the twelve comparisons), and significantly outperforms its pooling ablation ($p = 1.3 \times 10^{-2}$, Bonferroni-corrected). However, the test does not find significant differences in overall accuracy between the proposed model and the five strong transformer baselines (BERT, RoBERTa, DistilBERT, BERT+BiLSTM, BERT+BiGRU); raw p-values for these comparisons range from 0.026 to 0.561, and all become non-significant under Bonferroni correction.

This pattern is consistent with the observation that the proposed model's macro-F1 advantage is concentrated in the per-class structure rather than in overall accuracy. Macro-F1 weights each class equally, while accuracy weights each sample equally; under the strong class imbalance of GoEmotions (joy comprises 55% of test samples), the same change in per-class behaviour can yield different signals on the two metrics. We pursue this in Section 6.3.

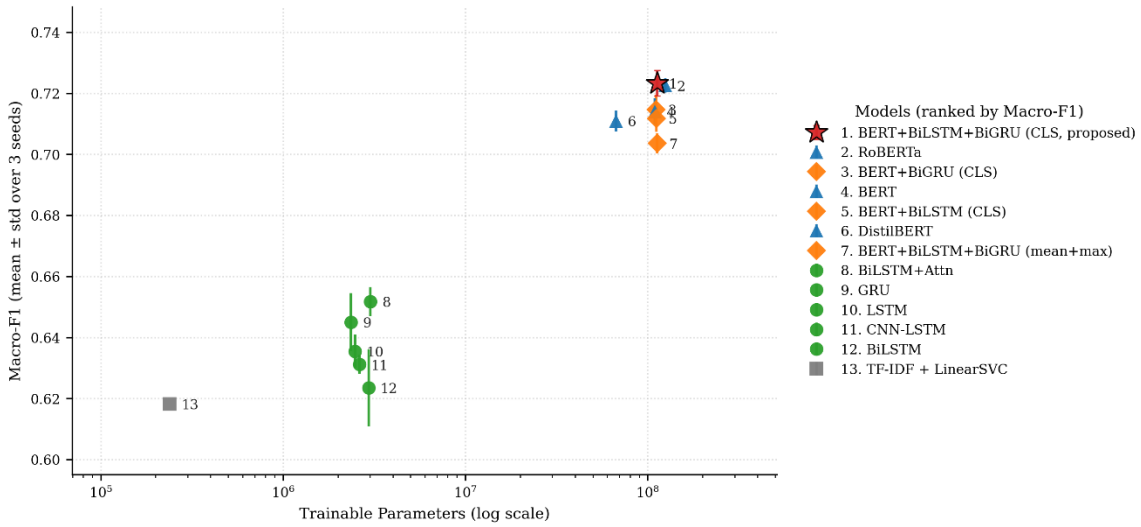

Fig. 2. Trainable parameters (log scale) versus macro-F1 (mean ± standard deviation over three seeds) for all thirteen model variants. Markers are coloured by family: classical (grey square), recurrent (green circle), transformer (blue triangle), hybrid (orange diamond), and the proposed model (red star). The legend ranks models by macro-F1. Three tiers are visible: a transformer cluster (~0.71–0.72) led by the proposed model, a recurrent tier (~0.62–0.65), and the classical baseline (0.6182).

6.2 Architectural Ablation

Table 2 supports a clean architectural ablation. Removing the BiGRU stage from the proposed model (yielding BERT–BiLSTM with CLS pooling) reduces macro-F1 from 0.7233 to 0.7117. Removing the BiLSTM stage instead (yielding BERT–BiGRU with CLS pooling) reduces it to 0.7147. Both single-recurrent variants are statistically tied with plain BERT-base (0.7140) under McNemar's test on overall accuracy. This indicates that neither recurrent layer in isolation provides a measurable improvement over BERT-base on this dataset; the gain emerges only from their combination.

The pooling ablation is more striking. Replacing CLS pooling with mean+max pooling—while keeping the BiLSTM and BiGRU layers, all hyperparameters, and the training recipe identical—reduces macro-F1 from 0.7233 to 0.7036, a drop of 1.98 percentage points that places the architecture below plain BERT-base. McNemar's test confirms this regression at $p < 0.05$ after Bonferroni correction. We interpret this finding as evidence that BERT's pre-trained [CLS] representation is a critical anchor for downstream classification: pooling strategies that ignore [CLS] in favour of aggregating recurrent outputs over time discard signal that the encoder was specifically pre-trained to provide. The recurrent stack functions best as a refinement of [CLS] rather than as an alternative aggregation.

The internal hyperparameter investigation we conducted prior to settling on this architecture confirmed the same sensitivity. A variant with the head learning rate halved from 1×10^{-3} to 5×10^{-4} (mean+max pooling retained) yielded macro-F1 = 0.6997 on seed 42, marginally lower than the original. Switching pooling to [CLS] (head learning rate retained at 1×10^{-3}) yielded macro-F1 = 0.7194 on the same seed—the change responsible for the architectural recovery. Pooling, not learning rate, accounts for the difference.

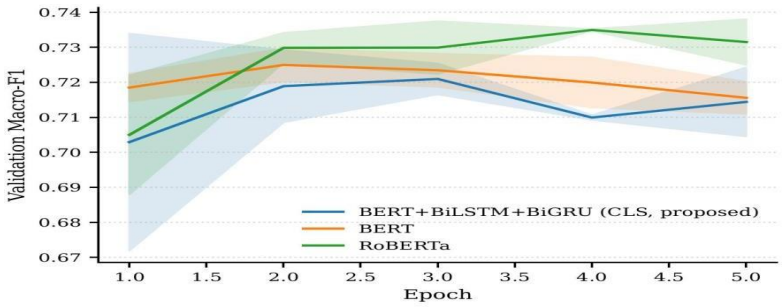

Fig. 3. Validation macro-F1 over training epochs for the proposed model and two strong transformer baselines. Solid lines show the mean across three seeds; shaded bands show ± one standard deviation. BERT and the proposed model reach peak validation performance within the first three epochs and then plateau, while RoBERTa-base continues to improve through epoch 4. All three models converge into the 0.71–0.74 range; the test-set ranking reported in Table 2, where the proposed model leads narrowly, is determined by the early-stopping checkpoint rather than the final-epoch state.

6.3 Per-Class Analysis

Table 3 reports per-class F1 for all thirteen model variants, with the proposed model in bold. The proposed model achieves the highest F1 on disgust (0.592), exceeding the next-best baseline (BERT–BiLSTM at 0.554) by 3.8 percentage points—the largest single-class margin in the table. On sadness it is the second-best (0.665), trailing only RoBERTa-base (0.675). On the remaining four classes the proposed model is competitive but not the leader: RoBERTa-base attains the highest F1 on anger (0.718) and surprise (0.748) and ties with BERT–BiGRU on joy (0.911), while BERT–BiGRU leads on fear (0.739). The proposed model's deficit on these four classes ranges from 0.001 (joy) to 0.015 (fear).

Table 3. Per-class F1 on GoEmotions Ekman-6 test set, mean over three random seeds. Bold row indicates the proposed model.

Model	**Anger**	**Disgust**	**Fear**	**Joy**	**Sadness**	**Surprise**
TF-IDF + LinearSVC	0.581	0.492	0.657	0.839	0.534	0.606
LSTM	0.592	0.498	0.619	0.860	0.565	0.679
GRU	0.592	0.535	0.624	0.864	0.590	0.665
BiLSTM	0.589	0.485	0.605	0.856	0.553	0.653
BiLSTM + Attention	0.606	0.506	0.643	0.865	0.602	0.688
CNN-LSTM	0.586	0.479	0.627	0.864	0.559	0.672
BERT-base	0.707	0.529	0.738	0.910	0.663	0.737
RoBERTa-base	0.718	0.552	0.731	0.911	0.675	0.748
DistilBERT-base	0.704	0.530	0.726	0.907	0.659	0.740
BERT + BiLSTM (CLS)	0.704	0.554	0.713	0.908	0.653	0.740
BERT + BiGRU (CLS)	0.711	0.526	0.739	0.911	0.654	0.747
BERT + BiLSTM + BiGRU (CLS) [proposed]	**0.707**	**0.592**	**0.723**	**0.910**	**0.665**	**0.743**
BERT + BiLSTM + BiGRU (mean+max)	0.700	0.533	0.703	0.907	0.649	0.730

To assess the statistical significance of these per-class differences, we apply McNemar's test to per-class correctness vectors—that is, for each true class c, we test whether the proposed model and a baseline differ in their probability of correct classification on samples whose true label is c. We perform this test for each of the six classes against five strong transformer baselines, yielding thirty tests, with Bonferroni correction across all thirty. The significant findings are summarised in Table 4.

Table 4. Significant per-class McNemar tests between the proposed model and transformer baselines, after Bonferroni correction across thirty tests. Δ recall is the proposed model's per-class accuracy minus the baseline's.

Baseline	**Class**	**Δ recall**	**p (Bonferroni)**	**Direction**
BERT-base	Surprise	+0.034	1.6×10^{-4}	Proposed wins
DistilBERT	Joy	+0.017	2.9×10^{-7}	Proposed wins
DistilBERT	Sadness	−0.060	3.9×10^{-5}	Proposed loses
BERT + BiLSTM (CLS)	Joy	+0.017	2.9×10^{-7}	Proposed wins
BERT + BiLSTM (CLS)	Surprise	−0.027	2.4×10^{-2}	Proposed loses
BERT + BiGRU (CLS)	Joy	+0.010	1.0×10^{-2}	Proposed wins
BERT + BiGRU (CLS)	Sadness	−0.059	6.9×10^{-6}	Proposed loses
RoBERTa-base	Sadness	−0.086	8.3×10^{-9}	Proposed loses

The per-class significance pattern reveals the architecture's true profile. The proposed model achieves significantly better recall on surprise relative to BERT-base (+3.4 percentage points) and on joy relative to DistilBERT, BERT–BiLSTM, and BERT–BiGRU (+1.0 to +1.7 points). However, it shows a statistically significant regression on sadness against three baselines: RoBERTa-base (−8.6 percentage points), BERT–BiGRU (−5.9 points), and DistilBERT (−6.0 points). It also shows a smaller but significant regression on surprise against BERT–BiLSTM (−2.7 points). On the smallest classes, fear (n = 85 in test) and disgust (n = 77), no comparison reaches significance after correction; the apparent trends from Table 3 are consistent with sampling variation.

These findings indicate that the proposed model's macro-F1 advantage is principally driven by the F1 gain on disgust, which more than offsets small deficits (0.001–0.015) on the remaining five classes (Table 3) when macro-F1 weights each class equally. Because disgust has only n = 77 test samples, the McNemar test in Table 4 cannot confirm this advantage at the 5% level after Bonferroni correction; it should therefore be regarded as suggestive. We discuss the implications in Section 7.

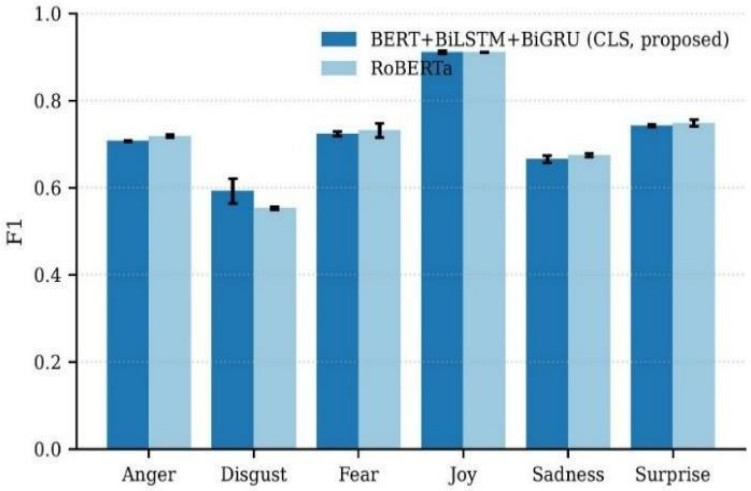

Fig. 4. Per-class F1 of the proposed model and the strongest transformer baseline (RoBERTa-base) on the six Ekman emotion classes. Error bars indicate ± one standard deviation across three seeds. The proposed model is stronger on disgust by approximately 4 percentage points. On the other five classes RoBERTa-base leads on F1 by at most 1.2 percentage points (largest gap on anger), with the joy gap being negligible at less than 0.1 percentage points. As Table 4 shows, RoBERTa-base's per-class recall on sadness is substantially higher than the proposed model's despite the small F1 gap on this class.

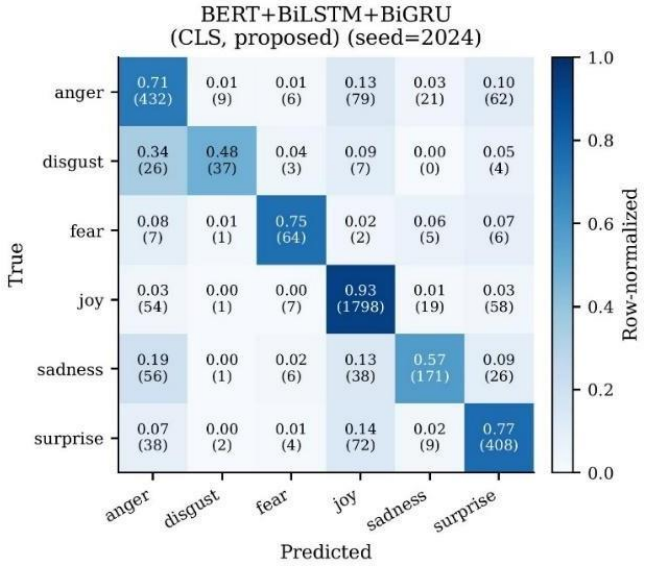

Fig. 5. Row-normalised confusion matrix for the proposed model on the GoEmotions Ekman-6 test set, computed on the median-F1 seed. Each cell shows both the row-normalised proportion and the absolute count. Diagonal entries indicate correct classifications. Off-diagonal mass concentrates in two patterns: confusion of disgust and sadness with anger (disgust→anger = 0.34; sadness→anger = 0.19), and a tendency for minority-class samples to be misclassified as joy, the dominant class (anger→joy = 0.13; sadness→joy = 0.13; surprise→joy = 0.14).

6.4 Efficiency

Table 5 reports parameter counts, training time per epoch, and inference latency for all thirteen model variants. The proposed model has 112.8M trainable parameters, approximately 3% more than BERT-base and 9.5% fewer than RoBERTa-base. Training time per epoch is the highest among all models at 371.4 ± 28.8 seconds on the RTX 4050 Laptop GPU, attributable to the additional gradient computation through the recurrent layers.

Table 5. Efficiency comparison. Parameters in millions; training time as mean ± standard deviation over all observed epochs; inference latency as median ms per sample over 1,000 timed iterations on a single NVIDIA RTX 4050 Laptop GPU.

Model	**Params**	**Train s/epoch**	**Latency b=1 (ms)**	**Latency b=32 (ms)**
TF-IDF + LinearSVC	0.24M	1.4 ± 0.1	0.31	0.03
LSTM	2.5M	13.4 ± 0.8	4.38	0.15
GRU	2.4M	14.1 ± 0.3	4.44	0.15
BiLSTM	3.0M	18.3 ± 0.7	8.29	0.29
BiLSTM + Attention	3.0M	18.2 ± 0.1	8.59	0.29
CNN-LSTM	2.6M	14.2 ± 0.1	5.19	0.17
BERT-base	109.5M	287.7 ± 0.6	11.15	5.44
RoBERTa-base	124.7M	351.4 ± 19.6	15.24	5.91
DistilBERT-base	67.0M	142.9 ± 6.4	8.19	2.97
BERT + BiLSTM (CLS)	111.6M	335.7 ± 19.0	20.99	5.73
BERT + BiGRU (CLS)	111.1M	315.1 ± 3.3	20.30	5.71
BERT + BiLSTM + BiGRU (CLS) [proposed]	**112.8M**	**371.4 ± 28.8**	**29.80**	**6.00**
BERT + BiLSTM + BiGRU (mean+max)	112.8M	365.0 ± 22.2	27.19	5.97

Inference latency exhibits a notable batch-size sensitivity. At batch size 1—the relevant regime for real-time, single-sample inference—the proposed model takes 29.8 ms per sample, approximately 2.7× the latency of plain BERT-base (11.1 ms) and 2.0× the latency of RoBERTa-base (15.2 ms). At batch size 32—the relevant regime for batch processing—the gap narrows substantially: the proposed model takes 6.00 ms per sample versus 5.44 ms for BERT-base, a 10% increase. The single-sample penalty arises because cuDNN's recurrent kernels incur substantial fixed launch overhead per call, which dominates when the per-step computation is small. At batch size 32, this overhead is amortised and the recurrent layers add only modestly to total cost. The proposed model is therefore well suited to batch-oriented deployment scenarios such as offline content moderation or batched stream processing, but less well suited to per-message real-time inference.

7. Discussion

7.1 What the Architecture Does

The empirical pattern in Sections 6.1–6.3 supports a coherent interpretation of the proposed architecture. Stacking a BiLSTM and a BiGRU on top of BERT, with classification from [CLS], achieves the highest mean macro-F1 on GoEmotions Ekman-6, but the gain over BERT-base is modest in absolute terms and the architecture is statistically tied with RoBERTa-base on overall accuracy. The architectural ablation is the more discriminating finding: neither BiLSTM nor BiGRU alone improves over BERT-base, and replacing CLS pooling with mean+max pooling causes the architecture to underperform BERT-base. The benefit of the architecture therefore arises from the interaction of three design choices, not from any one of them in isolation.

We hypothesise that the [CLS] representation, having been pre-trained as a sentence-level summary, provides a strong inductive prior for classification. Mean+max pooling over BiGRU outputs averages over a sequence of post-recurrent representations that are not directly tied to that pre-trained prior, weakening the connection between the encoder's pre-training objective and the downstream classification task. Stacking the BiLSTM and BiGRU as a refinement of [CLS]—rather than as an alternative aggregation—preserves that prior while adding a recurrent

processing stage that captures sequential dependencies the self-attention mechanism may underweight. The fact that single-recurrent variants do not improve over BERT suggests that the BiLSTM and BiGRU encode complementary aspects of the sequence: the BiLSTM's gated memory and the BiGRU's update-gate dynamics may capture different temporal regularities, and only their composition is sufficient to outperform plain BERT.

7.2 Per-Class Trade-offs

The per-class significance analysis (Section 6.3) reveals that the proposed model is not a uniform improvement over its baselines. It achieves significantly higher recall on surprise relative to BERT-base and on joy relative to several baselines, but exhibits a significant regression on sadness relative to RoBERTa-base, DistilBERT, and BERT–BiGRU. The sadness regression is the most pronounced, with an 8.6 percentage-point gap against RoBERTa-base. We do not have a definitive explanation for this asymmetry. One conjecture is that sadness in GoEmotions overlaps lexically with anger and the remorse–disappointment cluster that maps to the Ekman 'sadness' category, and that the additional recurrent processing amplifies anger-related signals at the expense of sadness, while RoBERTa's larger pre-training corpus provides more nuanced lexical disambiguation. A second possibility is that sadness benefits more than other classes from RoBERTa's specific pre-training data distribution. Distinguishing these explanations is beyond the scope of this paper but suggests a direction for future investigation.

This per-class heterogeneity has methodological implications. A macro-F1 improvement of 0.93 pp can mask a structure in which gains and losses are unevenly distributed across classes. We recommend that future work on fine-grained emotion classification routinely report per-class F1 with significance tests, in addition to overall metrics.

7.3 Efficiency Considerations

The proposed model carries an inference-latency cost that scales differently across deployment regimes. In single-sample mode, it is 2.7× slower than BERT-base; in batch mode at batch size 32, it is only 10% slower. For applications that can tolerate batched inference—offline analytics, content moderation, periodic ingestion of social-media streams—the additional cost is small and the macro-F1 advantage may be worth the trade-off. For real-time, per-message inference, the latency penalty is significant and the modest macro-F1 gain may not justify the architectural complexity over plain BERT-base. The choice between the two should be made with the deployment regime in mind.

7.4 Limitations

Several limitations of the present study should be acknowledged.

Single dataset. All experiments are conducted on GoEmotions Ekman-6. The architectural conclusions, particularly the importance of [CLS] pooling and the BiLSTM+BiGRU synergy—may not transfer to other emotion corpora with different label distributions, text lengths, or domain characteristics. Generalisation across datasets such as ISEAR, SemEval-2018 [25], or EmotionLines remains future work.

Single backbone. The proposed hybrid uses BERT-base as the encoder. We did not investigate the same hybrid stack on top of RoBERTa or DistilBERT, which could plausibly yield different macro-F1 levels and per-class profiles. The interaction between encoder choice and recurrent stacking is an open question.

Sadness regression. As reported in Section 6.3, the proposed model significantly underperforms three baselines on sadness recall. We did not pursue corrective measures (class-weighted loss, focal loss, or sadness-specific data augmentation), as these would change the comparison conditions relative to prior work. Addressing this regression while maintaining the gains on joy and surprise is a natural next step.

Single-sample latency. The 2.7× single-sample inference latency relative to BERT-base limits the proposed model's suitability for real-time deployments. Techniques such as recurrent layer fusion, kernel-level optimisation, or distillation into a non-recurrent student could mitigate this cost but were not investigated.

Hyperparameter scope. We selected a single set of hyperparameters per model based on validation performance, then ran three seeds. Broader hyperparameter sweeps (learning-rate schedules, dropout schedules, recurrent hidden sizes, depth) might yield different conclusions. Our two-LR-group recipe (encoder at 2×10^{-5}, recurrent and head at 1×10^{-3}) was found to be necessary for stable training but was not exhaustively tuned.

Frozen baseline embeddings. The embedding tables of the five word-level recurrent baselines were held at their random initialisation and were not updated during training.

Hardware. Latency measurements are obtained on a single consumer-grade laptop GPU and may not reflect data-centre deployment behaviour. Relative comparisons between models are nevertheless informative because all models are evaluated on the same hardware under the same software stack.

8. Conclusion

This paper presented a controlled empirical study of an enhanced hybrid BERT–BiLSTM–BiGRU architecture for text-based emotion detection on GoEmotions Ekman-6. Across twelve baselines, three random seeds, and rigorous statistical testing, the proposed model attained the highest mean macro-F1 of 0.7233 ± 0.0042. The improvement over BERT-base is modest (+0.93 percentage points), and the model is statistically tied on overall accuracy with RoBERTa-base, despite using approximately 9.5% fewer trainable parameters.

The architectural ablation is the principal methodological contribution. We showed that none of the three design choices—the BiLSTM stage, the BiGRU stage, or [CLS] pooling—is sufficient on its own. The two single-recurrent variants are statistically tied with plain BERT-base, and replacing [CLS] pooling with the more common mean+max pool reduces macro-F1 below the BERT-base baseline. We also reported per-class trade-offs: gains on the majority class joy and on surprise are partially offset by a significant regression on sadness against several transformer baselines, including a notably large gap against RoBERTa-base. The proposed model thus represents a careful architectural recipe that improves the average across emotion classes, but does not uniformly dominate prior work; we report this trade-off explicitly.

From a deployment perspective, the model carries a 2.7× single-sample inference latency penalty over BERT-base that is largely amortised at batch size 32, making it well suited to batch-oriented deployments and less suited to real-time single-message inference.

Future work should investigate (i) the transfer of the architectural recipe to other emotion corpora and to other transformer backbones, (ii) targeted measures to address the sadness regression without reverting the gains on joy and surprise, and (iii) latency-reduction techniques such as kernel fusion or distillation that preserve the macro-F1 advantage at lower inference cost. More broadly, the pooling-strategy sensitivity we observe suggests that hybrid transformer–RNN architectures should treat pooling as a primary design variable rather than a default.

9. Declarations

Funding: The authors received no specific funding for this work.

Clinical Trial Registration: Not applicable.

Ethics Approval and Consent to Participate: Not applicable.

Consent for Publication: Not applicable.

References

1. Mohammad SM, Turney PD (2013) Crowdsourcing a word–emotion association lexicon. Computational Intelligence 29(3):436–465. https://doi.org/10.1111/j.1467-8640.2012.00460.x
2. Mohammad SM, Bravo-Marquez F (2017) WASSA-2017 shared task on emotion intensity. In: Proceedings of the 8th workshop on computational approaches to subjectivity, sentiment and social media analysis (WASSA). Association for Computational Linguistics, Copenhagen, pp 34–49. https://doi.org/10.18653/v1/W17-5205
3. Devlin J, Chang M-W, Lee K, Toutanova K (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 (NAACL-HLT). Association for Computational Linguistics, Minneapolis, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423
4. 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 preprint arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692
5. Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108
6. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
7. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference

on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, pp 1724–1734. https://doi.org/10.3115/v1/D14-1179
8. Acheampong FA, Nunoo-Mensah H, Chen W, Rubungo AN (2020) Recognizing emotions from texts using a BERT-based approach. In: Proceedings of the 17th international computer conference on wavelet active media technology and information processing (ICCWAMTIP). IEEE, Chengdu, pp 117–121. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317523
9. Tan KL, Lee CP, Anbananthen KSM, Lim KM (2022) RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access 10:21517–21525. https://doi.org/10.1109/ACCESS.2022.3152828
10. Talaat AS (2023) Sentiment analysis classification system using hybrid BERT models. Journal of Big Data 10:110. https://doi.org/10.1186/s40537-023-00781-w
11. Dodge J, Gururangan S, Card D, Schwartz R, Smith NA (2019) Show your work: improved reporting of experimental results. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, pp 2185–2194. https://doi.org/10.18653/v1/D19-1224
12. Reimers N, Gurevych I (2019) Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, pp 3982–3992. https://doi.org/10.18653/v1/D19-1410
13. Demszky D, Movshovitz-Attias D, Ko J, Cowen A, Nemade G, Ravi S (2020) GoEmotions: a dataset of fine-grained emotions. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics, online, pp 4040–4054. https://doi.org/10.18653/v1/2020.acl-main.372
14. Ekman P (1992) An argument for basic emotions. Cognition and Emotion 6(3–4):169–200. https://doi.org/10.1080/02699939208411068
15. Plutchik R (1980) A general psychoevolutionary theory of emotion. In: Plutchik R, Kellerman H (eds) Theories of emotion. Academic Press, New York, pp 3–33. https://doi.org/10.1016/B978-0-12-558701-3.50007-7
16. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd international conference on learning representations (ICLR), San Diego. https://doi.org/10.48550/arXiv.1409.0473
17. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30. Curran Associates, Long Beach, pp 5998–6008. https://doi.org/10.48550/arXiv.1706.03762
18. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: human language technologies (NAACL-HLT). Association for Computational Linguistics, San Diego, pp 1480–1489. https://doi.org/10.18653/v1/N16-1174
19. Pota M, Ventura M, Catelli R, Esposito M (2021) An effective BERT-based pipeline for Twitter sentiment analysis: a case study in Italian. Sensors 21(1):133. https://doi.org/10.3390/s21010133
20. Dror R, Baumer G, Shlomov S, Reichart R (2018) The hitchhiker's guide to testing statistical significance in natural language processing. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics, Melbourne, pp 1383–1392. https://doi.org/10.18653/v1/P18-1128
21. McNemar Q (1947) Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12(2):153–157. https://doi.org/10.1007/BF02295996
22. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush AM (2020) Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics, online, pp 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6
23. Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: Proceedings of the 7th international conference on learning representations (ICLR), New Orleans. https://doi.org/10.48550/arXiv.1711.05101
24. Maruf AA, Khanam F, Haque MM, Jiyad ZM, Mridha MF, Aung Z (2024) Challenges and opportunities of text-based emotion detection: a survey. IEEE Access 12:18416–18450. https://doi.org/10.1109/ACCESS.2024.3356357
25. Mohammad SM, Bravo-Marquez F, Salameh M, Kiritchenko S (2018) SemEval-2018 task 1: affect in tweets. In: Proceedings of the 12th international workshop on semantic evaluation (SemEval). Association for Computational Linguistics, New Orleans, pp 1–17. https://doi.org/10.18653/v1/S18-1001
26. Chutia T, Baruah N (2024) A review on emotion detection by using deep learning techniques. Artificial Intelligence Review 57:202. https://doi.org/10.1007/s10462-024-10831-1
27. Kusal S, Patil S, Choudrie J, Kotecha K, Vora D, Pappas I (2023) A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection. Artificial Intelligence Review 56:15129–15215. https://doi.org/10.1007/s10462-023-10509-0

28. Rahman MM, Shiplu AI, Watanobe Y, Alam MA (2025) RoBERTa-BiLSTM: a context-aware hybrid model for sentiment analysis. IEEE Transactions on Emerging Topics in Computational Intelligence 9:3788–3805. https://doi.org/10.1109/TETCI.2025.3572150
29. Jahin MA, Shovon MSH, Mridha MF, Islam MR, Watanobe Y (2024) A hybrid transformer and attention-based recurrent neural network for robust and interpretable sentiment analysis of tweets. Scientific Reports 14:24882. https://doi.org/10.1038/s41598-024-76079-5