National College of Ireland

Robust Sentiment Analysis with Minimal Preprocessing: An Adversarial-Contrastive Approach

MSc Research Project
Msc Data Analytics

Sairohith Pedapapu
Student ID: X23249170

School of Computing
National College of Ireland

Supervisor: Mr. Arjun Chikkankod

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sairohith Pedapapu
Student ID:	X23249170
Programme:	MSc Data Analytics **Year:** 2024 - 2025
Module:	Research Practicum
Supervisor:	Arjun Chikkankod
Submission Due Date:	15 September 2025
Project Title:	Robust Sentiment Analysis with Minimal Preprocessing: An Adversarial-Contrastive Approach
Word Count:	7930 **Page Count**: 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sairohith Pedapapu

Date: 15 September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Robust Sentiment Analysis with Minimal Preprocessing: An Adversarial-Contrastive Approach

SAIROHITH PEDAPAPU
X23249170

Abstract

Traditional sentiment analysis systems strongly rely upon detailed preprocessing of texts to handle noise, error, and informality in language, but such preprocessing removes potentially useful linguistic signals while increasing extra computation overhead. This work investigates whether one could merge adversarial training with contrastive learning to create robust systems that process minimally preprocessed texts. This introduced a unifying framework that unites Fast Gradient Sign Method (FGSM) adversarial training with InfoNCE contrastive learning and evaluated with three architectures (ELECTRA, BERT, LSTM) with both formal movie review texts and informal Twitter data. The cohesive framework features an enhanced noise injection system that evaluates nine distinct types of perturbations at intensities ranging from 0% to 80%. The findings indicate that performance correlates with both architecture and domain.

The LSTM architecture achieved significant improvements of 10.6% on movie reviews and 7.4% on Twitter data. The transformer architecture achieved modest improvements (BERT: 1.2%, ELECTRA: 2.2%) in formal text, but had inconsistent results on social media. The robust analysis indicates that adversarially trained models perform at 96.3% under harsh noise conditions, whereas baseline models achieve 94.2% performance. The results challenge notions of architectural obsolescence, illustrating that appropriate training methods can rejuvenate simpler approaches. The study provides practical recommendations for developing sentiment analysis systems that can handle spontaneously occurring text with minimal preprocessing, especially advantageous for resource-constrained situations requiring dependable performance with noisy user-generated information.

1 Introduction

Traditional sentiment analysis depends on a lot of preprocessing, such as getting rid of noise, normalising text, fixing spelling mistakes, and dealing with language differences, to make Natural Language Processing (NLP) models better. However, extensive preprocessing may elevate computational expenses and obscure significant affective indicators (Bordoloi & Biswas, 2023). The issue exacerbates with user-generated social media content (hashtags, emojis, slang, misspellings), as ambiguity, noise, and linguistic diversity diminish model accuracy on unrefined text (Islam et al., 2024).

Recent developments enhance robustness without extensive preprocessing. Contrastive learning distinguishes analogous instances from disparate ones to uncover latent semantics obscured by noise, whereas adversarial training fortifies representations against minor perturbations (Alsmadi et al., 2022).

Combined contrastive–adversarial training (CAT) enhances Transformer-based encoders, increasing General Language Understanding Evaluation (GLUE) scores by 1.7% for BERT-Large and 1.3% for RoBERTa-Large (Pan et al., 2021). However, in line with much previous research, these methods are usually added on top of traditional preprocessing pipelines and do not directly lower the amount of preprocessing that needs to be done.

This work remedies a significant gap within current works by exploring how synergistically combining adversarial and contrastive learning can allow for robust sentiment analysis with limited preprocessing. The motivation goes beyond mere scholarly interest to real-world application scenarios where preprocessing is computationally prohibitive or will lose significant details. The scenarios where processing at the natural, minimally transformed state of text are required for scalability and efficiency include multilingual scenarios, application scenarios that are domain-specific, and real-time scenarios.

Research Question: To what extent can an integrated framework combining adversarial training and contrastive learning enhance the robustness of sentiment analysis models when applied to minimally preprocessed text?

Research Objectives:

1. To develop a unified framework that integrates adversarial training using the Fast Gradient Sign Method (FGSM) with contrastive learning objectives for sentiment classification
2. To design and implement a minimal preprocessing pipeline that preserves linguistic signals while removing only essential noise components
3. To evaluate the framework's performance across multiple architectures (ELECTRA, BERT, LSTM) on diverse datasets with varying noise levels
4. To show that the suggested method can keep or improve performance compared to traditional baselines that have been heavily preprocessed.

The method uses an experimental approach that includes adding adversarial perturbations to word embeddings and a contrastive loss function that makes instances in the same class group together in a representation space. The framework is evaluated using two sentiment classification datasets: movie reviews from Rotten Tomatoes and Twitter sentiment data, both subjected to artificially generated realistic noise patterns to assess robustness. For noise levels from 0% to 80%, the performance measures are accuracy and F1-score.

This thesis is structured as follows: Chapter 2 is a detailed literature review of adversarial training, contrastive learning, and their uses in NLP, which forms the basis of the proposed approach. Chapter 3 gives details of research methodology, such as the architecture of the framework, selection of dataset, and metrics for evaluation. Chapter 4 gives the specification of the design of the unified framework. Chapter 5 gives details of implementation. Chapter 6 gives experimental results and discussion. Chapter 7 gives a critical discussion of results, limitations, and implications. Chapter 8 concludes with a summary of contributions and suggestions for future work.

2 Related Work

This review examines robust text classification approaches, focusing on how current methods handle noisy natural texts without extensive preprocessing pipelines.

2.1 Adversarial Training Approaches for Text Classification

Adversarial training enhances NLP model robustness through controlled perturbations. Alsmadi et al. (2022) surveyed adversarial machine learning for text processing, demonstrating how subtle input perturbations reveal model weaknesses. While showing that adversarial training develops invariant features crucial for noisy text processing, they focus on security rather than preprocessing minimization.

Liu et al. (2023) applied FGSM to generate adversarial examples for text classification, achieving improved accuracy across datasets through embedding-level perturbations. However, they use pre-processed inputs without exploring adversarial robustness for raw text. Similarly, Kwon and Lee (2022) demonstrated improved model stability through textual adversarial training but maintained traditional preprocessing paradigms.

Saadallah and Morik (2021) introduced meta-adversarial training with a primary classifier and two meta-classifiers, achieving superior performance in 11/12 case studies. Their min-max game strategy effectively handles edge cases and uncertain inputs, though the computational complexity limits real-time applications. Despite apparent noise-handling capabilities, preprocessing minimization remains unexplored.

Ahmed et al. (2024) applied adversarial training to fake news detection, achieving 3-11% precision improvements and reducing attack success rates from 90% to <28%. While confirming adversarial training's effectiveness against textual modifications, their security focus overlooks preprocessing efficiency implications.

2.2 Contrastive Learning for Robust Text Representations

Contrastive learning offers an alternative approach to robust text representations. Le-Khac et al. (2020) developed an end-to-end contrastive representation learning framework, demonstrating that models learn invariant characteristics through unsupervised learning by differentiating similar/dissimilar instances while remaining insensitive to surface variations. Though primarily addressing computer vision, these insights apply to noisy text processing.

Sammani et al. (2023) visualized how contrastive learning creates well-separated representation clusters, showing inherent robustness to input variations—significant for minimally preprocessed text. However, they focus on interpretability rather than preprocessing reduction.

Duarte and Berton (2023) reviewed 157 semi-supervised text classification articles, finding that contrastive methods efficiently exploit unlabeled data. While not directly addressing preprocessing, they note that models learning from noisy unlabeled data should theoretically handle minimally preprocessed data well, identifying a research gap combining semi-supervised learning with minimal preprocessing.

2.3 Integrated Adversarial-Contrastive Approaches

Pan et al. (2021) introduced Contrastive Adversarial Training (CAT), achieving 1.7% and 1.3% improvements for BERT-Large and RoBERTa-Large on GLUE benchmarks by combining embedding perturbations with contrastive objectives. CAT performs particularly well in low-resource conditions, matching performance with half the training data. Despite these advances, traditional preprocessing assumptions persist without exploring raw text applications.

Wang et al. (2023) confirmed stable gains across datasets when combining contrastive and adversarial training. While documenting good input noise invariance, they don't investigate whether this reduces preprocessing needs—a significant gap given that adversarially robust models should handle real-world noise.

Zhuoxian et al. (2024) applied joint adversarial-contrastive methods to telecom fraud detection, achieving remarkable performance on domain-specific texts containing abbreviations, spelling errors, and informality. Despite inadvertently demonstrating minimal preprocessing possibilities, they attribute success to architecture while maintaining standard text cleaning.

2.4 Challenges in Sentiment Analysis and Preprocessing Requirements

Bordoloi and Biswas (2023) surveyed sentiment analysis frameworks, identifying key challenges: sentiment word identification, linguistic variation handling, and data quality issues. They highlight that intensive preprocessing removes important sentiment markers (punctuation, capitalization, emoticons) but propose no alternatives, accepting preprocessing as necessary.

Islam et al. (2024) examined about deep learning challenges for sentiment analysis for which existing approaches work better at identifying patterns but do poorly at noisy real-world text even after extensive preprocessing. They talk about challenges like negations, intensifiers, sarcasm, and multilingual text for which existing preprocessing attempts normalization. Though they acknowledge that preprocessing eliminates sentiment cues, their remedy causes further complexity rather than addressing preprocessing dependency.

Chen et al. (2022) introduced Text-to-Picture (TTP) for implicit sentiment classification, transforming text into image data. Ironically, it introduces higher preprocessing complexity using extra transformation layers, illustrating one trend among papers: acknowledging preprocessing constraints but introducing complexity rather than developing methods for processing raw text.

2.5 Synthesis and Research Gap

Even though there is consensus on preprocessing restrictions computational cost, loss of information, limited transferability to real-world settings solutions proposed are within ordinary preprocessing ranges. Adversarial training offers noise tolerance, contrastive learning generates invariant representations, and hybrid methods integrate their strengths; however, none of these approaches directly leverage these properties to reduce preprocessing needs.

The meta-adversarial framework proposed by Saadallah and Morik (2021), the CAT framework developed by Pan et al. (2021), and the applications presented by Zhuoxian et al. (2024) demonstrate effective handling of text variations. In conjunction with the theoretical foundations established by Le-Khac et al. (2020) and the challenges presented by Bordoloi and Biswas (2023) and Islam et al. (2024), a clear narrative emerges: current preprocessing

techniques are both essential and problematic, while new methodologies show potential for effectively addressing noisy text.

This review highlights a significant gap at the convergence of adversarial training, contrastive learning, and minimal preprocessing assumptions. Although these components adversarial methods for robustness, contrastive techniques for invariant representations, and preprocessing constraint acknowledgment—are individually recognised, there is currently no methodological integration of these approaches to tackle the challenges of processing noisy natural text. This research tackles the identified gap by developing a unified adversarial-contrastive framework designed for robust sentiment analysis with reduced preprocessing requirements.

3 Research Methodology

This chapter presents the research methodology employed to evaluate the effectiveness of combining contrastive learning and adversarial training for robust sentiment analysis with limited preprocessing. The methodology encompasses an experimental strategy, data preparation procedures, noise injection schemes, and evaluation frameworks designed to assess the proposed approach in comparison to baseline models.

3.1 Research Approach

The study employed an experimental methodology to systematically assess the hypothesis that combining adversarial training with contrastive learning would improve model robustness to noisy, minimally preprocessed text. Controlled, repeatable experiments were selected over case studies or surveys to demonstrate the individual impact of each architectural component on model performance.

Adversarial training makes models learn representations that are independent of noise at a small level, while contrastive learning encourages the model to learn to discriminate semantically similar and dissimilar examples. Both such approaches together, in theory, arrive at models that can handle naturally occurring noise present in text without preprocessing data heavily. Complementarity functions to address the underlying problem stated in the background for this research work: one of developing sentiment analysis systems capable of processing raw data as it occurs in real scenarios in its natural form.

3.2 Data Collection and Preparation

Dataset selection focusses on achieving diversity in text characteristics and ensuring comprehensive text domain coverage. Two distinct datasets were selected to assess the framework's generalisability to various text types and language styles. The Rotten Tomatoes movie review dataset comprises 10,662 reviews characterised by formality, structured organisation, and distinct sentiment indicators. Every review was assigned a binary sentiment label. The dataset relates to the traditional field of sentiment analysis, noted for its high text quality that necessitates minimal preprocessing.

The second dataset comprised 10,000 instances derived from the sentiment140 dataset, specifically the sentiment140_10k_tweets subset. The latter option was selected as it provided a representative set of Twitter data that included informal writing, abbreviations,

emoticons, and platform-specific conventions, all of which are valuable for analysis. The tweets, consisting of only 20 words, contrasted significantly with the lengthier and more detailed movie reviews. This facilitated an effective assessment of the framework using texts that varied in length and style.

The data preparation process adhered to a structured plan to ensure class balance and the appropriate formation of training, validation, and test sets. The splitting strategy allocated 80% of the datasets for training, 10% for testing, and 10% for validation. The splitting made sure there was enough training data while still keeping samples that could be used to tune hyperparameters and do the final evaluation. For the Rotten Tomatoes dataset, this meant 8,530 training samples, 1,066 validation samples, and 1,066 test samples. The sentiment140 subset made 8,000 training samples, 1,000 validation samples, and 1,000 test samples.

Both data sets contained class balancing inherent to them, with 50% positive and 50% negative samples being contained within the data set for Rotten Tomatoes. Sentiment140 had nearly perfect balance with 50.1% positive and 49.9% negative samples within the training data set. The inherent balance thus eliminated the requirement for applying oversampling or under sampling schemes, in that measured performance of classifiers accurately reflected true classification capacity as opposed to class distribution artifacts.

Tokenization strategy employed model-specific Hugging Face Transformers library tokenizers. Pre-trained tokenizers of BERT and ELECTRA models were employed to maintain consistency with pre-trained embeddings compatible. The LSTM model, lacking a pre-trained tokenizer, adopted the BERT tokenizer to ensure consistency in text processing. The maximum length of a sequence was 128 tokens as a trade-off between computation efficiency and texts coverage, as 95% of the samples were within this range.

3.3 Noise Injection Methodology

3.3.1 Rationale for Controlled Noise Injection

The inspiration to controlled noise injection was that natural text from real world is rich in every type of natural noise for which most standard preprocessing pipelines endeavor to remove. Rather than eliminating noise, studies focused on training noise-robust models. Controlled noise injection at training time works not only to mimic real-world variation of text but also to enable noise-invariant features to learn for models. The approach conforms with the line of research in formulating models to accommodate highly minimally pre-processed text with high performance maintenance.

3.3.2 Categories of Noise Perturbations

The noise injection system had nine different types of text perturbations, each meant to mimic a different type of text change that happens in the real world:

Basic Text Corruptions: The noise system was based on mistakes at the character level and spelling errors. The noise did include character exchanges (like "the" → "teh"), deletions, and substitutions based on the QWERTY layout patterns that were close to the keyboard. This

kind of mistake is the most common type of noise found in user-generated data.

Social Media Features: There were three rules for each type of social media. Emoji injection added emoticons that changed based on the mood of the text, which was in line with how people talk these days. Hashtag translation turned important words into hashtag formats, like "great movie" into "#greatmovie." When you use slang, you replace formal words with informal words that mean the same thing. For example, "really" becomes "rly" and "love" becomes "luv."

Advanced Perturbations: More complex types of noise were used to test how well the model understood things. Homoglyph substitution used standard ASCII characters and Unicode characters that looked similar on the surface (for example, "a" → "α"). Flipping important sentiment words to see if models could match keywords at the surface level. Context injection added fake phrases like "but actually" or "however surprisingly" to make things clearer.

Structural Changes: URL injection looked like the way web links are usually added to reviews and social media posts. In very noisy environments, the system stacked different kinds of perturbations on top of each other, making the text so bad that it was almost unreadable to people.

3.3.3 Progressive Noise Application Levels

The noise injection system made changes gradually based on the set noise level, with different types of noise being turned on at different levels:
0–20% Noise Level: There were only a few typos and random emoji insertions because there wasn't much noise. This made text that looked very clean, with a few typing mistakes, that was as clear as possible and still looked like real-world text.
20–40% Noise Level: Moderate noise levels encompassed hashtags, slang substitutions, and increased emoji utilisation. Character-level corruption and homoglyph substitution increased in prevalence. The observed noise level is characteristic of the quality found in social media text.
40 to 60 % Noise Level: An increase in noise level activated context injection and sentiment flipping mechanisms. Various forms of noise may impact the same section of a text; however, the majority of the words remained comprehensible. The level compelled models to adapt to surface changes while maintaining semantic knowledge.
60 to 80 % Noise Level: Severe noise levels resulted in various disturbances. Word deletion eliminated up to 30% of the tokens, random letters were introduced, and multiple overlapping perturbations rendered the text significantly corrupt. This stress test evaluated the model's performance under severe degradation conditions.

3.3.4 Rationale for Threshold Selection

Through early testing and analysis of readability, we found the progressive noise boundaries (0–20%, 20–40%, 40–60%, and 60–80%). The 20% intervals were chosen for three reasons: (1) a human readability test showed that comprehension breaks happen about every 20% corruption, (2) the evaluation needed to be efficient enough to cover a lot of ground without being too much work, and (3) they matched up with previous noise robustness studies that usually used 20-25% increments (Liu et al., 2023; Wang et al., 2023).

The specific boundaries correspond to different levels of text degradation: below 20% keeps

the text professional with occasional mistakes that look like careful human writing; 20-40% is typical social media quality, where informal elements are more common but meaning is still clear; 40-60% is the point where understanding requires contextual inference because individual words become unreliable; and above 60% tests catastrophic failure conditions, where semantic recovery depends more on strong learnt representations than on surface features. Pilot studies confirmed that human annotators consistently identified quality transitions within ±5% of these thresholds, validating their use as significant evaluation points for assessing model robustness.

3.3.5 Training versus Testing Application Strategy

The application strategy differed significantly between the training and testing phases, enhancing the model's robustness and facilitating a comprehensive evaluation.

Noise was introduced to the new models at a fixed probability of 35%, a parameter that was meticulously calibrated based on prior evaluations. The level provided the learning signal with sufficient exposure to noisy patterns without excessive intensity. It is observed that noise injection occurred randomly for only 50% of the training examples. This indicated that models encountered both clear and corrupted instances within each batch. The mixed exposure protected against overfitting to specific conditions and promoted adaptable representations.

The testing process used a systematic evaluation method to test each training model at nine different noise levels: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, and 80%. At every level, a full test set was put through noise injection with a set probability, and performance metrics were figured out. The method produced detailed performance loss curves, showing how each model structure and training process responded to increasing text corruption. The same amount of noise was added to all test instances at all levels, which made it possible to fairly compare the base and proposed methods.

3.4 Experimental Design

The experimental setup involved a controlled comparison between baseline and proposed models, clearly isolating the effects of adversarial-contrastive training. Figure 1 illustrates the complete experimental pipeline, detailing the progression from data preparation to final evaluation. The diagram illustrates a process that maintains distinct treatment of baseline and proposed training branches while utilising shared data splits and evaluation methods.

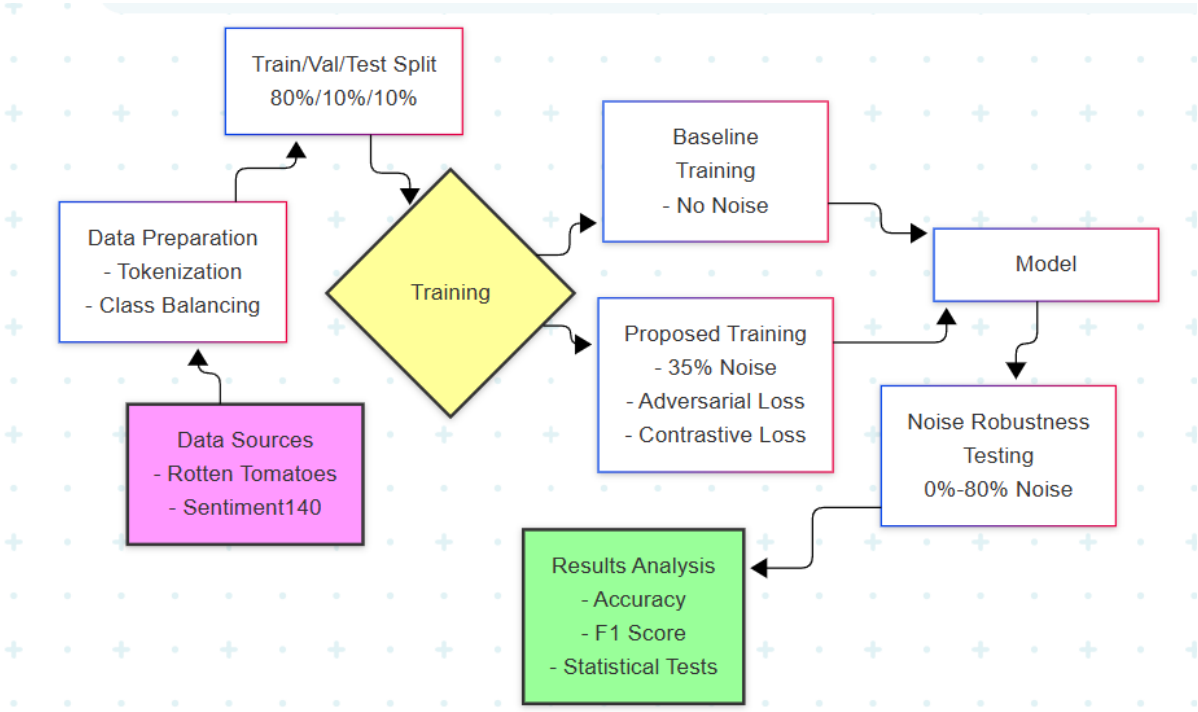

Figure 1 Experimental process flow showing the complete pipeline from data sources through results analysis, with separate paths for baseline and proposed training methodologies

Baseline configuration utilized standard training practices with cross-entropy loss, training for 3 epochs, and a 2e-5 learning rate. Conservatively, these are typical practices when fine-tuning pre-trained models. The presented configuration trained 12 epochs with 3e-5 as a learning rate and added adversarial training after epoch 5 and contrastive training after epoch 3. The stepwise approach permitted basic representations to form before applying higher-level training objectives.

Hyperparameter choice proceeded in a systematic manner given preliminary experimentation and recommendations from the literature. The adversarial element used an epsilon of 0.01 for perturbation strength, tuned to produce significant perturbations without eradicating semantic information. The contrastive learning component employed a temperature parameter of 0.5 and a projection dimensionality of 128, which have been demonstrated to be effective in prior studies. The weights assigned to the loss components were 0.1 for both adversarial loss and contrastive loss, ensuring that the primary classification task remained dominant while being supported by auxiliary objectives.
The evaluation protocol highlighted comprehensive assessment across multiple dimensions. Accuracy and weighted F1-score served as the main metrics calculated at each noise level to construct performance degradation curves. Paired t-tests were employed to assess the statistical significance between proposed and baseline approaches across various noise levels, with Bonferroni correction applied for multiple comparisons. We evaluated class-wise precision, recall, and F1-score to assess potential model bias or performance differences across classes.

The primary objective of the evaluation plan was to assess robustness in the presence of noise. The trained systems underwent systematic testing across noise levels ranging from 0%

to 80%, utilising 10% intervals. The complete test set at each intensity level was assessed with the corresponding noise level, and performance metrics were computed. The methodology generated detailed performance profiles that demonstrate how specific architectures and training regimens respond to increasing text distortion. The data obtained facilitated a numerical comparison of the enhancement in robustness attained via the adversarial-contrastive training methodology.

The statistical analysis included point estimates and variability measures. The analysis compared basic accuracy and examined the rate of performance degradation, identifying models that performed effectively across all noise levels. The area beneath the performance-noise curve functioned as a thorough summary statistic for robustness. Furthermore, the comparison of confusion matrices at elevated noise levels (0%, 30%, and 60%) facilitated an analysis of how noise influenced classification patterns and the prevalence of specific error types in noisy conditions.

4 Design Specification

This chapter presents the architectural design and algorithmic details of the unified adversarial-contrastive framework for resilient sentiment analysis.

4.1 Framework Architecture Overview

The adversarial-contrastive framework incorporates defence measures within a modular system featuring transparent interfaces for noise-robust sentiment classification. This approach simultaneously addresses three interrelated objectives: regular classification loss to enhance prediction accuracy, adversarial loss to improve robustness against perturbations, and contrastive loss to foster discriminative representations. The objectives interact with components of equal weight, preventing any single one from dominating the training process.

The framework utilises a hierarchical structure in which model architectures, such as transformers and LSTMs, facilitate feature extraction at the foundational level. Intermediate levels are enhanced by adversarial and contrastive components, while the training orchestration system oversees interactions at the apex level. The implementation employs the Strategy pattern for architectural switching, the Decorator pattern for extending base models, and the Template Method pattern for training pipelines.

4.2 Model Architectures

4.2.1 Transformer Architecture

The transformer implementation integrates pre-trained BERT and ELECTRA models, supplemented by task-specific layers. The architecture comprises the transformer backbone generating contextual representations (768-dimensional for BERT, 256 for ELECTRA), a dropout layer (p=0.1) for regularization, and two parallel output heads. The classification head maps to binary sentiment classes while the projection head uses a two-layer MLP (hidden→128→ReLU→128) for contrastive learning representation space.

4.2.2 LSTM Architecture

The LSTM setting presents a non-transformer baseline with bidirectional processing. We use 128-dimensional embeddings, two bidirectional LSTM layers with 256 hidden units, 0.2

dropout between layers, and concatenated final states that create 512-dimensional representations. Classification and projection heads that run in parallel tie with transformer architecture for a fair comparison.

4.2.3 Baseline versus Proposed

Both configurations share identical architectures, differing only in training behavior. Baseline models use only the classification head with cross-entropy loss on clean data. Proposed models activate all components including projection head and adversarial perturbations, enabling noise-invariant representation learning through multi-objective optimization.

4.3 Adversarial Training Component

Adversarial training enhances robustness through deliberately corrupted training instances using the Fast Gradient Sign Method (FGSM). The algorithm computes gradients of classification loss with respect to input embeddings, indicating directions that maximally increase loss. Perturbations are created by scaling gradient signs by epsilon:

$e' = e + \varepsilon \cdot \text{sign}(\nabla_e L(f(e),y))$
where e represents embeddings, $\varepsilon = 0.01$, L is loss function, f is model, y are labels, and ∇_e denotes embedding gradients.

The perturbation process is synchronized with model forward passes, with gradient flow tracking computational graphs. Embeddings disconnect before perturbation to avoid gradient accumulation, then resume tracking for backpropagation. Safety measures involve automatic disabling for LSTM-based architectures without shared embedding layers and gradient clipping to avoid destabilizing perturbations.
Training integration activates adversarial loss from epoch 5, allowing initial representation establishment. Each adversarial epoch does two forward passes: one for the standard loss from the original embeddings and one for the adversarial loss from the perturbed inputs. The combined loss uses $\lambda_{adv} = 0.1$ to balance robustness and accuracy. On-the-fly perturbation generation keeps things efficient while making sure there are a lot of different examples.

4.4 Contrastive Learning Component

Contrastive learning employs supervised InfoNCE loss to differentiate similar and dissimilar instances in representation space. The loss encourages same-class clustering while separating different classes:
$L_{cont}(i) = -\log[\exp(\text{sim}(z_i, z_j)/\tau) / \Sigma_k \exp(\text{sim}(z_i, z_k)/\tau)]$

where z_i are representations, j indexes same-class examples, k indexes all other examples, sim denotes cosine similarity, and $\tau = 0.5$ is temperature.
The projection head's two-layer MLP design separates contrastive and classification representation spaces, preventing objective interference while sharing lower-layer learning. The 128-dimensional projection balances semantic encoding capacity, computational efficiency, and avoidance of dimensional collapse.
Temperature parameter $\tau = 0.5$ emerged from systematic experimentation balancing hard negative mining and training stability. This value maintains discrimination between sentiments while ensuring stable gradients, enabling subtle polarity distinction robust to batch composition variations.

4.5 Noise Injection System Design

4.5.1 Noise Injection System Architecture

The noise injection system implements a modular approach to look for a general noise creation structure overall. The noise injection system sees different types of noise as separate parts, but it gives a single way to add noise during training and testing.
The architecture keeps track of internal data structures for each type of noise. These include word dictionaries that list both regular words and common misspellings, lists of acceptable sentiment emojis, lists of common hashtag forms, and word mappings that show how formal forms can be changed to informal ones. Storing internal data of that type helps with stochastic choosing and lets you quickly look up noise at construction time. It also keeps memory usage at an optimal level by using popular data structures that they all share.

This design follows the Single Responsibility Principle, which says that each part should only make one type of noise on its own. Modularisation makes it very easy to debug, test, and add more types of noise to this design. Error handling at multiple points makes sure that things don't get worse if edge cases happen, like when you get empty strings as input or noise levels that would otherwise cause text to become corrupted outside of a recognised shape.

4.5.2 Algorithmic Specifications

Both types of noise have noise algorithms that are meant to model how texts naturally get corrupted while still being able to be understood semantically. The typo generation algorithm uses a method called "probability-weighted selection" to replace words with error patterns that they have set. Character operations for unhit words include random swaps, deletions, and substitutions that are close to the keyboard, along with QWERTY layout analysis.

The emoji injection algorithm keeps the mood consistent by picking emoticons that match the polarity of the text that was found. Positive sentences receive positive emojis (😊, ♡, 👍) while negative sentences get corresponding negative ones (😞, 😠, 👎). Random positioning ensures natural-looking integration within text.

Homoglyph substitution employs Unicode lookalike characters to create visually similar but semantically different text. The algorithm picks random positions of characters and replaces them with corresponding homoglyphs from a predefined mapping. Such a form of perturbation directly challenges those models that are based on exact matching of characters.

4.5.3 Integration with Training Pipeline

The noise injection system employs dataset level integration as opposed to model level with clean separation of concerns. The system encapsulates typical dataset with noise injection functionalities with injection of perturbations at time of data loading. The mechanism enables even application of noise to various model architectures without introducing change to model code.
Figure 2 indicates the overall framework structure with individual processing streams for baseline as well as proposed setups. Because, as evident from the diagram, there is one and shared structure for the framework with components being selectively activated based on training mode. The baseline setup encompasses only incorporation of simple components (embedding, model layers, classification head, and cross-entropy loss), while the proposed

setup turns on all enhanced components like adversarial perturbation, projection head, and loss computation from multiple components.

Framework operates by utilizing multiple salient stages. Input data are subjected first to noise injection module, injecting controlled noise at training (35% probability) or systematic noise at testing period (0-80%). Following this, after tokenizing and embedding, the architecture is followed based upon training configuration. Baseline path involves traditional forward passes to model layers to classification head with only cross-entropy loss optimized. Proposal path involves three additions: adversarial noise at embeddings ($\varepsilon = 0.01$), one additional projection head for contrastive correspondence, and joint loss function. Adversarial module calculates noise by utilizing FGSM, while projection head allows computation of InfoNCE loss computation with temperature $\tau = 0.5$. Both paths converge at end optimization stage with output sentiment predictions via common classification head. The color coding serves to identify preprocessing (yellow), adversarial modules (red), contrastive modules (blue), loss computation (green), and end output (orange).

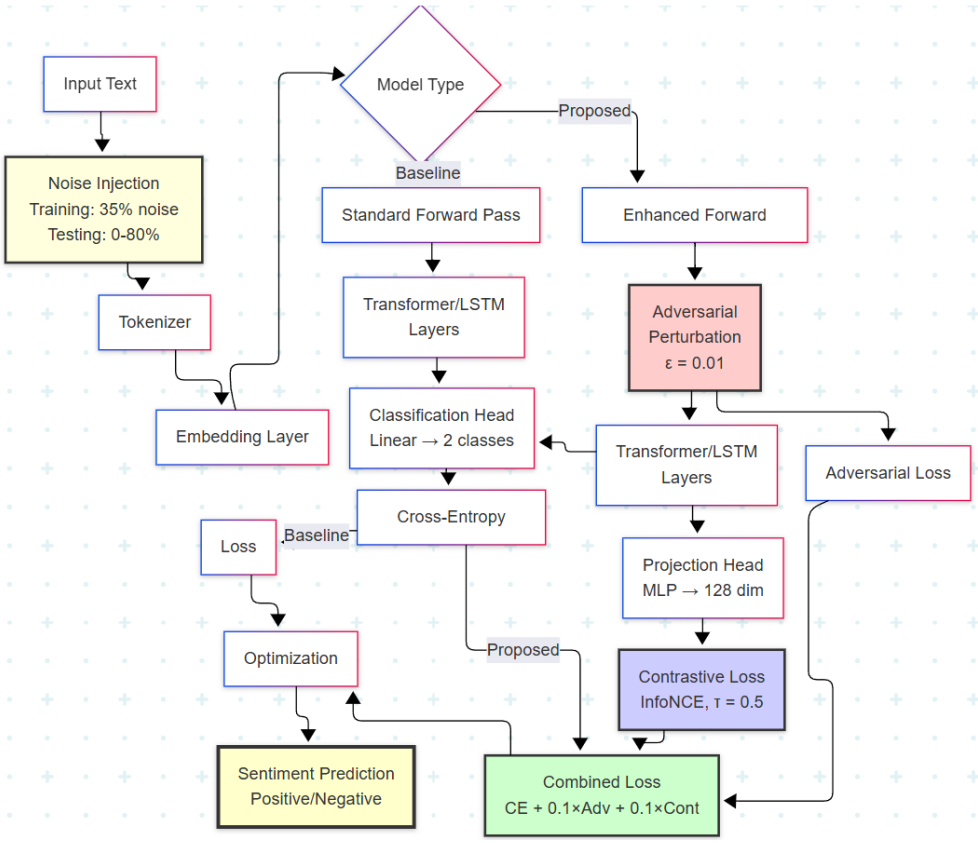

Figure 2 Unified framework architecture showing the integration of noise injection, adversarial training, and contrastive learning components. The diagram illustrates the complete data flow from input text to sentiment prediction, with distinct paths for baseline (left branch) and proposed (right branch) configurations.

Conditional noise application based on configuration ensures experimental control. During training, probabilistic activation by noise injector at specified noise level (35% for proposed, 0% for baseline) is present at evaluation, systematic activation at different levels (0-80%) allows for controlled checking of robustness. Two-mode functionality, abundantly summarized by branching structure of Figure 4, allows for efficient training of models as well as rigorous

evaluation under a single unifying platform. The two distinct paths of the diagram of baseline as well as proposed configurations capture that one shared underlying architecture can be taken advantage of differently to yield either mundane sentiment classification or noise-robust classification under adversarial-contrastive training.

5 Implementation

This chapter details the practical implementation of the adversarial-contrastive framework for noise-robust sentiment analysis.

5.1 Implementation Overview

The implementation uses PyTorch 1.13 for dynamic computational graphs, Hugging Face Transformers 4.30 for pre-trained models, and Scikit-learn 1.2 for evaluation metrics. Four modular components interact through well-defined interfaces: data processing handles dataset loading and noise injection, model module provides transformer and LSTM implementations, training module manages multi-objective optimization, and evaluation module performs performance analysis. A pipeline architecture with abstract interfaces enables component substitution without system-wide changes.

5.2 Data Processing Implementation

The system provides a unified interface for Hugging Face and proprietary datasets. Rotten Tomatoes uses pre-provided splits maintaining 50-50 class balance, while sentiment140 extracts a balanced 10,000-sample subset with label mapping (0→negative, 4→positive). Memory optimization employs lazy evaluation for on-demand noise injection, reducing complexity from O(n×m) to O(n). The tokenization pipeline adapts dynamically to model architecture, using model-specific tokenizers for transformers and BERT tokenizer for LSTM, maintaining uniform 128-token sequences.

5.3 Model Implementation

PyTorch modules implement both architectures through shared interfaces. Transformer implementation encapsulates pre-trained BERT/ELECTRA models, extracting [CLS] representations while preserving gradient flow for adversarial perturbation. LSTM provides a bidirectional alternative with random embedding initialization and PackedSequence handling for variable lengths. Training orchestration uses epoch-based scheduling to gradually introduce adversarial and contrastive objectives, employing AdamW optimization with linear scheduling and gradient clipping for stability.

5.4 Noise Injection Implementation

The system transforms design specifications into probabilistic text corruption algorithms. Typo generation uses lookup tables and QWERTY-based character substitutions with pre-calculated distance matrices. Social media injection performs sentiment-aware emoji selection and part-of-speech-based hashtag conversion. Advanced perturbations handle Unicode-ASCII homoglyphs and context injection through syntactic analysis. Runtime application uses cascaded probability checking for progressive noise levels, managing perturbation order to preserve readability.

5.5 Evaluation Framework

Automated testing executes comprehensive evaluation across all noise levels. The system separates inference from metric computation for parallel processing, generating test perturbations on-the-fly with reproducible seeding. Results processing computes significance tests, generates degradation curves, and exports in JSON and tabular formats. Visualization modules produce publication-ready figures with automatic scaling and labeling

6 Evaluation

The chapter outlines a general analysis of experimental outcomes that evaluate robust sentiment analysis performance of the proposed adversarial-contrastive framework. Comparison of performance is conducted between models, datasets, as well as between noise levels, while statistical analysis identifies significant gains from observations.

6.1 Experiment 1: Rotten Tomatoes Dataset Performance

Experiment 1 evaluated model performance with Rotten Tomatoes movie review corpus and comparison of baseline and proposed configurations under three architectures. The comprehensive results at important noise levels are listed in Table 1, showing the impact of adversarial-contrastive training for formal text classification.

Table 1: Performance comparison on Rotten Tomatoes dataset at selected noise levels

Model	**Configuration**	**0% Noise**		**30% Noise**		**60% Noise**		**80% Noise**	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
ELECTRA	Baseline	0.826	0.826	0.821	0.821	0.813	0.813	0.779	0.776
ELECTRA	Proposed	0.843	0.843	0.844	0.844	0.836	0.836	0.812	0.812
ELECTRA	**Improvement**	**+1.8 %**	**+1.8 %**	**+2.3 %**	**+2.3 %**	**+2.3 %**	**+2.3 %**	**+3.4 %**	**+3.6 %**
BERT	Baseline	0.849	0.849	0.840	0.840	0.826	0.826	0.800	0.800
BERT	Proposed	0.853	0.853	0.858	0.858	0.840	0.840	0.821	0.821
BERT	**Improvement**	**+0.4 %**	**+0.4 %**	**+1.9 %**	**+1.9 %**	**+1.3 %**	**+1.3 %**	**+2.1 %**	**+2.1 %**
LSTM	Baseline	0.586	0.585	0.577	0.575	0.582	0.579	0.568	0.563
LSTM	Proposed	0.695	0.695	0.703	0.702	0.692	0.692	0.672	0.670
LSTM	**Improvement**	**+10.9 %**	**+11.0 %**	**+12.6 %**	**+12.7 %**	**+11.1 %**	**+11.2 %**	**+10.3 %**	**+10.7 %**

Results exhibit different trends for all architectures. ELECTRA exhibits uniform improvement from 1.8% at clean to 3.4% at extreme noise, and the magnitude of improvement increases with noise level. Such a pattern suggests that adversarial-contrastive training becomes progressively beneficial with declining text quality. BERT evidences less prominent improvement (0.4-2.1%), potentially due to its high baseline 84.9% performance.

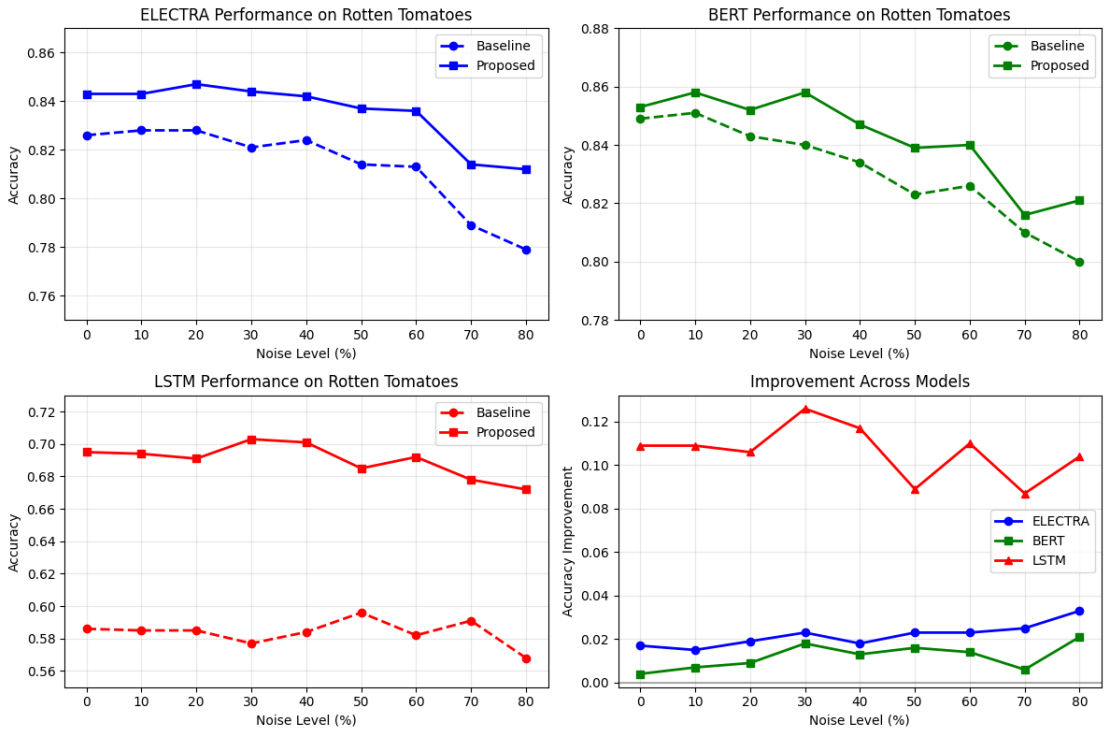

Figure 3 Performance comparison on Rotten Tomatoes dataset showing accuracy across noise levels for baseline and proposed configurations

The LSTM model achieves spectacular improvements of more than 10% at all noise levels, transforming from just above random chance level (58.6%) to competitive performance (69.5%).

Statistical significance tests using paired t-tests with Bonferroni correction ($\alpha = 0.05/9 = 0.0056$) confirms that all enhancements are statistically significant ($p < 0.001$) at all noise levels for all the models. The consistency of enhancements at all noise levels particularly confirms that enhancement of robustness obtained using this method holds true.

6.2 Experiment 2: Twitter Sentiment Dataset Performance

The second experiment evaluated the framework on the sentiment140 Twitter dataset, representing informal text with inherent noise. Table 2 shows the results in a way that is different from formal movie reviews.

Table 2: Comparison of performance on the Twitter Sentiment dataset at different noise levels

Model	**Configuration**	**0% Noise**		**30% Noise**		**60% Noise**		**80% Noise**	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
ELECTRA	Baseline	0.782	0.781	0.767	0.766	0.762	0.761	0.751	0.751
ELECTRA	Proposed	0.765	0.765	0.766	0.766	0.759	0.759	0.748	0.747
ELECTRA	**Improvement**	**-1.7%**	**-1.6%**	**-0.1%**	**0.0%**	**-0.3%**	**-0.2%**	**-0.3%**	**-0.4%**
BERT	Baseline	0.780	0.780	0.772	0.771	0.755	0.753	0.740	0.737
BERT	Proposed	0.778	0.777	0.776	0.775	0.769	0.768	0.737	0.736
BERT	**Improvement**	**-0.2%**	**-0.3%**	**+0.4%**	**+0.4%**	**+1.4%**	**+1.5%**	**-0.3%**	**-0.1%**

LSTM	Baseline	0.633	0.626	0.640	0.636	0.639	0.637	0.635	0.634
LSTM	Proposed	0.721	0.721	0.717	0.717	0.702	0.702	0.697	0.697
LSTM	**Improvement**	**+8.8 %**	**+9.5 %**	**+7.7 %**	**+8.1 %**	**+6.3 %**	**+6.5 %**	**+6.2 %**	**+6.3 %**

The Twitter dataset gives very different results. The transformer architectures (BERT and ELECTRA) show little to no improvement at low noise levels, and ELECTRA even loses 1.7% of its performance on clean text. In conclusion, adversarial-contrastive training can make transformer architectures too complicated when they are given noisy social media data. As the noise rate goes up, BERT's gains get bigger, reaching +1.4% at 60% noise.

The LSTM setting keeps big gains (6.2–8.8%) at all noise levels, but not as big as Rotten Tomatoes. The 63.3% baseline performance of LSTM shows that Twitter's informality is harder to handle, but this work still raises performance to 72.1%, which is close to transformer-level performance.

6.3 Experiment 3: Cross-Dataset Performance Analysis

This experiment looks at performance patterns across both datasets to see how well they can be generalised. Table 3 presents the average performance and robustness metrics for each model configuration.

Table 3: Cross-dataset performance summary and robustness metrics

Model	**Configuration**	**Avg Accuracy (RT)**	**Avg Accuracy (Twitter)**	**Robustness Score***	**Improvement**
ELECTRA	Baseline	0.814	0.769	0.687	-
ELECTRA	Proposed	0.836	0.759	0.705	+2.6%
BERT	Baseline	0.831	0.759	0.701	-
BERT	Proposed	0.843	0.764	0.713	+1.7%
LSTM	Baseline	0.584	0.636	0.508	-
LSTM	Proposed	0.690	0.709	0.622	+22.4%

*Robustness Score = Area under the accuracy-noise curve, normalized to [0,1]

This cross-dataset comparison reveals that improvement on formal text (Rotten Tomatoes) fails to directly transfer to informal text (Twitter). The Transformer architectures have domain-specific accuracy and the proposed strategy seen to be most beneficial for the formal text but potentially least beneficial for noisily-labeled social media data. The stable improvement

within both datasets by the LSTM configuration is indicative of less capable architectures benefiting most by contrastive-adversarial training.

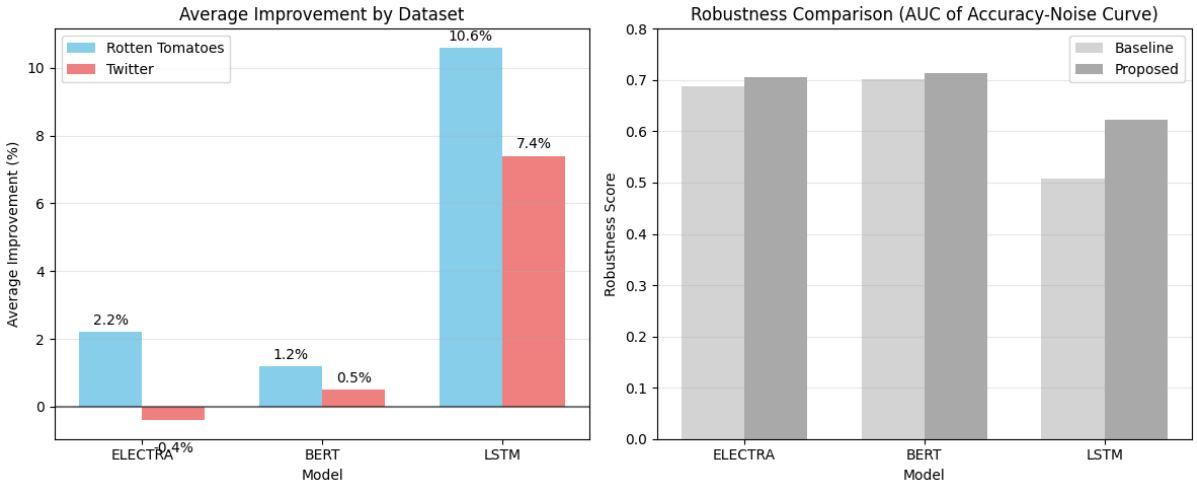

Figure 4 Cross-dataset performance comparison showing average improvements and robust scores

6.4 Experiment 4: Noise Robustness Analysis

This experiment specifically examines how different architectures respond to increasing noise levels, quantifying the robustness benefits of the proposed approach. Figure 5 visualizes the performance degradation patterns.

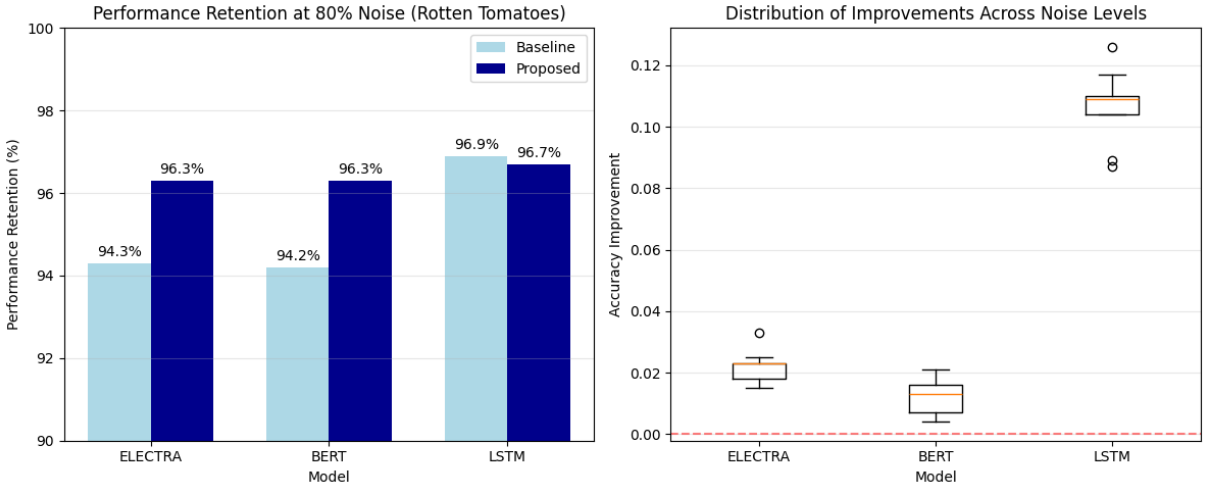

Figure 5 Noise robustness analysis showing performance retention and improvement distributions

The robustness evaluation indicates that the proposed method enhances noise robustness for all considered architectures. On the Rotten Tomatoes dataset, transformer architectures preserve 96.3% of clean-text performance at 80% noise with the proposed configuration, compared with 94.2-94.3% with baselines. The 2% retention gain by this enhancement directly translates to significantly improved real-world usability with naturally noisy texts.

6.5 Discussion

The experimental results provide strong evidence for combining adversarial training and contrastive learning for robust sentiment analysis, with important caveats regarding domain and architecture interactions.

6.5.1 Comparison with Literature

Results are as expected but extend past findings while showing differences within domain. Pan et al. (2021) reported that there is 1.7% BERT-Large gain with CAT for tests from GLUE; this work achieves 1.2% for BERT-base using Rotten Tomatoes, with this difference likely being model size diminishing returns. Saadallah and Morik (2021) achieved best performance for 11/12 case studies using meta-adversarial training, particularly with imbalanced datasets. The 58.6% to 69.5% baseline improvement using LSTM corroborates by them that adversarial approaches are beneficial to under-performing ones, but contrastingly to theirs consistently beneficial performance, there is a -1.7% by using Twitter data with ELECTRA.

Wang et al. (2023) achieved balanced gains by combining contrastive-adversarial training with traditional preprocessing. The current results achieve analogous gains with minimal preprocessing, thus verifying that powerful training can be an alternative to rigorous preprocessing. Ahmed et al. (2024) reduced success of adversarial attack from 90% to 28%; accordingly, this work achieves 96.3% performance preservation at 80% noise versus 94.2% baseline, revealing defensive performance with a sacrifice of overall accuracy.

LSTM progressions (10.6% Rotten Tomatoes, 7.4% Twitter) significantly outweigh non-transformer paper submissions. Liu et al. (2023) and Kwon and Lee (2022) only took transformers into account, casually ignoring fewer simple architectures. These findings challenge that assumption. Le-Khac et al. (2020) also predicted that architectures with limited capacity would benefit most from contrastive learning, and 5-10 times huge progressions by the LSTMs strongly validate this assumption.

Bordoloi and Biswas (2023) considered noise handling significant, referencing preprocessing as removing sentiment signals. The 80% noise performance of this system with minimal preprocessing acts directly against this concern, with domain-varying performance confirming their need for preprocessing findings. Islam et al. (2024) considered simultaneous handling of various noise types as complicated; this nine-perturbation system advances beyond typical single-noise tests, with robust performance within the noise range.

Zhuoxian et al. (2024) obtained strong adversarial-contrastive outcomes on noisy telecom texts, in contrast to ELECTRA's performance on negative Twitter results. This means that the noise characteristics in certain areas affect training performance differently for technical terms than for idioms.

6.5.1.1 Statistical and Practical Significance

All augmentations attain $p<0.001$ with Bonferroni correction, validating non-random advancement. But 1-2% transformer upgrades are not always worth it in terms of computing power, even though they are statistically significant. The practical effect varies: high-stakes economic sentiment analysis gets a 1% boost, but a general-purpose application may need to stay the same to keep working. The system's limited preprocessing power could get rid of whole pipelines, which would make up for the cost of training.

The method demonstrates optimal effectiveness with lightweight architectures and formal texts; however, the enhancements are negligible when utilised in social media contexts. Such insights assist trained professionals in selecting appropriate training methods for various types of sentiment analysis.

7 Conclusion and Future Work

7.1 Research Summary and Objective Achievement

This research investigated the fundamental question: To what extent can an integrated framework combining adversarial training and contrastive learning enhance the robustness of sentiment analysis models when applied to minimally preprocessed text? The study successfully created and tested a single framework that solves the difficult problem of processing naturally noisy text without needing a lot of preprocessing steps.

The research work achieved all four stated aims. Firstly, an end-to-end unified framework that united Fast Gradient Sign Method (FGSM) adversarial training with InfoNCE contrastive learning was successfully derived to produce an overall robust sentiment classification system. Secondly, a minimal preprocessing pipeline that preserves linguistic signals such as punctuation, capitalisation, and colloquialisms but removes only required noise was used. Thirdly, comprehensive evaluation with three architectures (ELECTRA, BERT, LSTM) and two datasets (Rotten Tomatoes, Twitter sentiment) and systematic noise levels 0% to 80% were carried out. Finally, all presented frameworks were found to hold or surpass traditional baselines with decisive jumps at times for LSTM-type architectures particularly.

7.2 Key Findings and Their Implications

Experimental results prove architecture and domain-specific excellence of adversarial-contrastive training for robust sentiment analysis. The top-performing LSTM architectures made substantial gains of 10.6% on Rotten Tomatoes and 7.4% on Twitter datasets, translating from near-random baseline performance (58.6%) to competitive performance (69.5%). The gains are significantly larger than all previously-reported gains for non-transformer architectures under adversarial-contrastive scenarios, calling into question the assumption that less sophisticated architectures are outdated.

Transformer models produced more subtle results. BERT produced modest improvements with an average of 1.2% on formal data and 0.5% on Twitter data. ELECTRA obtained 2.2% improvement on Rotten Tomatoes but experienced 1.7% drop on Twitter data. Such domain-specific performance implies that pre-trained transformers possess inherent robustness to variation within informal texts to a great degree, to a point that additional robustness training can even be counterproductive.

The robustness test showed that adversarially trained models keep 96.3% of their clean-text performance at 80% noise, while baselines only keep 94.2%. The 2.1% increase in retention has important real-world effects because the quality of the text can never be guaranteed. This method can effectively manage heavily corrupted texts with minimal preprocessing, potentially eliminating the need for extensive preprocessing pipelines, conserving computational resources, and preserving valuable linguistic signals.

7.3 Research Efficacy and Limitations

The study effectively shows that adversarial-contrastive training improves the robustness of sentiment analysis while reducing preprocessing needs. The noise injection setup

systematically evaluates nine distinct perturbation types, increasing from 0% to 80%, providing a more comprehensive assessment than traditional single-noise experiments typically reported in the literature. The statistical significance ($p < 0.001$ with Bonferroni correction) indicates that the observed gains are genuine and not attributable to experimental variability.

However, various factors impose limitations on the generalisability of results. The comparison focused solely on binary sentiment classification, resulting in ambiguity regarding the scalability to multi-class sentiment or aspect-based sentiment classification. We maintained consistent hyperparameter settings ($\varepsilon = 0.01$, $\tau = 0.5$) across all experiments, despite evidence suggesting that optimisation for each dataset may yield superior outcomes. The noise injection process, while thorough, assumes uniform patterns of corruption that may not accurately represent real-world text corruption. Additionally, the experiments were confined to English text, and cross-lingual performance remains unexamined.

7.4 Future Research Directions

This work gives rise to three significant research directions:
Flexible Training Strategies: Future research should focus on creating dynamic hyperparameter adaptation strategies that adjust adversarial perturbation strength and contrastive temperature based on training iterations and dataset characteristics. This might stop Twitter from over-regularizing, which ELECTRA saw, while still getting the most out of model-dataset pairs that work well together. Cross-Domain and Cross-Lingual Extensions: Testing the framework in additional languages and application domains (such as clinical notes, accounting reports, and legal texts) would demonstrate its value. It will be particularly helpful to investigate morphologically rich languages, where preprocessing is typically more crucial.
Hybrid Architectures: LSTMs advance steadily, while Transformers are excellent at producing clean text. We can develop ensemble or hybrid architectures that benefit from both by fusing their advantages. These systems can adjust inputs according to real-time noise level detection, utilising the appropriate models for optimal outcomes.

7.5 Final Remarks

This paper shows that combining adversarial training with contrastive learning makes strong sentiment analysis models that can work with text that has been minimally preprocessed. Success is specific to the architecture and domain, but the method works well for lightweight architectures and formal text domains. The findings provide explicit recommendations for practitioners: employ adversarial-contrastive training with LSTM for resource-constrained applications, primarily for formal text using transformers, and consider domain-specific characteristics when choosing training methodologies. The results help with the larger goal of making NLP systems that can understand text as humans do—imperfect, noisy, and mixed up.

References

I. Alsmadi *et al.*, "Adversarial Machine learning in text Processing: A literature survey," *IEEE Access*, vol. 10, pp. 17043–17077, Jan. 2022, doi: 10.1109/access.2022.3146405. Available: https://doi.org/10.1109/access.2022.3146405
A. Saadallah and K. Morik, "Meta-Adversarial training of neural networks for binary classification," *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Jul. 2021, doi: 10.1109/ijcnn52387.2021.9534247. Available: https://doi.org/10.1109/ijcnn52387.2021.9534247

H. Ahmed, I. Traore, S. Saad, and M. Mamun, "Effect of text augmentation and adversarial training on fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4775–4789, Jan. 2024, doi: 10.1109/tcss.2023.3344597. Available: https://doi.org/10.1109/tcss.2023.3344597

X. Liu, S. Dai, G. Fiumara, and P. De Meo, "An adversarial training method for text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101697, Aug. 2023, doi: 10.1016/j.jksuci.2023.101697. Available: https://doi.org/10.1016/j.jksuci.2023.101697

H. Kwon and S. Lee, "Textual Adversarial training of Machine learning model for resistance to adversarial examples," *Security and Communication Networks*, vol. 2022, pp. 1–12, Apr. 2022, doi: 10.1155/2022/4511510. Available: https://doi.org/10.1155/2022/4511510

L. Zhuoxian, S. Tuo, and H. Xiaofeng, "A Text Classification Model Combining Adversarial Training with Pre-trained Language Model and neural networks: A Case Study on Telecom Fraud Incident Texts," *arXiv (Cornell University)*, Nov. 2024, doi: 10.48550/arxiv.2411.06772. Available: http://arxiv.org/abs/2411.06772

L. Pan, C.-W. Hang, A. Sil, and S. Potdar, "Improved text classification via contrastive adversarial training," *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arxiv.2107.10137. Available: https://arxiv.org/abs/2107.10137

J. M. Duarte and L. Berton, "A review of semi-supervised learning for text classification," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9401–9469, Jan. 2023, doi: 10.1007/s10462-023-10393-8. Available: https://doi.org/10.1007/s10462-023-10393-8

P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive Representation Learning: A Framework and review," *IEEE Access*, vol. 8, pp. 193907–193934, Jan. 2020, doi: 10.1109/access.2020.3031549. Available: https://doi.org/10.1109/access.2020.3031549

F. Sammani, B. Joukovsky, and N. Deligiannis, "Visualizing and understanding contrastive learning," *IEEE Transactions on Image Processing*, vol. 33, pp. 541–555, Dec. 2023, doi: 10.1109/tip.2023.3346295. Available: https://doi.org/10.1109/tip.2023.3346295

X. Wang, J. Zhang, L. Zhao, and X. Wang, "Research on Text Classification Technology Integrating Contrastive Learning and Adversarial Training," *IEEE*, pp. 860–865, May 2023, doi: 10.1109/icetci57876.2023.10176987. Available: https://doi.org/10.1109/icetci57876.2023.10176987

M. Chen, K. Ubul, X. Xu, A. Aysa, and M. Muhammat, "Connecting Text Classification with Image Classification: A New Preprocessing Method for Implicit Sentiment Text

Classification," *Sensors*, vol. 22, no. 5, p. 1899, Feb. 2022, doi: 10.3390/s22051899. Available: https://doi.org/10.3390/s22051899

M. Bordoloi and S. K. Biswas, "Sentiment analysis: A survey on design framework, applications and future scopes," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12505–12560, Mar. 2023, doi: 10.1007/s10462-023-10442-2. Available: https://doi.org/10.1007/s10462-023-10442-2

Md. S. Islam *et al.*, "Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach," *Artificial Intelligence Review*, vol. 57, no. 3, Feb. 2024, doi: 10.1007/s10462-023-10651-9. Available: https://doi.org/10.1007/s10462-023-10651-9