

MULTI-TIER SENTIMENT ANALYSIS OF SOCIAL MEDIA TEXT USING SUPERVISED MACHINE LEARNING**DR.V. UMARANI**Professor, Department of Information Technology,
Jawaharlal Nehru Technological University Hyderabad,umarani@jntuh.ac.in**PODETI MEGHANA**Post Graduate Student, M. Tech (SE) Department of Information Technology, Jawaharlal Nehru
Technological University, Hyderabad,podetimeghana@gmail.com**ABSTRACT**

Social media platforms generate a large amount of text where users express their opinions and emotions. Sentiment analysis helps in identifying these opinions and understanding user behavior. Traditional sentiment analysis methods usually classify text into basic categories such as positive or negative, which provides limited insight. This project proposes a multi-tier sentiment analysis system using supervised machine learning techniques. The system processes social media text in multiple stages and classifies sentiments into five levels: very positive, positive, neutral, negative, and very negative. Text preprocessing is applied to remove noise and improve data quality. Feature extraction methods convert text into numerical form suitable for machine learning models. Supervised classification algorithms are then used to identify sentiment categories. Experimental results show that the proposed multi-tier approach improves accuracy and provides more meaningful sentiment classification. The system is simple, efficient, and suitable for analyzing large volumes of social media data.

Keywords:

Sentiment Analysis, SocialMedia, Machine Learning, Multi-tier Classification, Text Preprocessing, Feature Extraction, Supervised Learning, Opinion Mining.

INTRODUCTION

Over the past decade, the exponential rise of social media platforms has generated an unprecedented volume of user-generated textual data, transforming the digital landscape into a rich source of public opinion and behavioral insights. This rapid expansion has positioned Sentiment Analysis (SA) as a pivotal research domain, enabling the automated extraction and interpretation of opinions across diverse applications, including market analysis, policy evaluation, and decision support systems. The ability to accurately capture user sentiment has become increasingly critical for organizations and governments seeking to derive actionable intelligence from large-scale data.

Conventional supervised machine learning models, such as Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, have been extensively employed for sentiment classification tasks. While these models achieve strong performance in binary classification, their effectiveness significantly deteriorates in multi-class sentiment analysis, which aims to capture finer-grained emotional distinctions. Despite its complexity, multi-class classification is essential for providing deeper and more actionable insights compared to binary approaches. Additionally, aspect-based sentiment analysis further enhances interpretability by identifying sentiment toward specific attributes; however, it also introduces additional computational and modelling challenges.

A major limitation in existing approaches lies in their inability to effectively handle the inherent characteristics of social media text. Such data is often noisy, unstructured, and linguistically complex, containing misspellings, informal expressions, domain-specific slang, spam content, and intricate constructs such as negation and sarcasm. These factors contribute to non-linear feature distributions, making the data difficult to model using traditional algorithms. Consequently, baseline machine learning models fail to capture high-level contextual relationships, leading to suboptimal performance in multi-class settings. Although advanced deep learning architectures, including Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and transformer-based models such as BERT, have demonstrated superior capability in learning complex patterns,

they require substantial computational resources and large annotated datasets, limiting their practicality in resource-constrained environments.

In addition to model selection, effective pre-processing of textual data plays a crucial role in enhancing sentiment classification performance. Techniques such as tokenization, stop-word removal, normalization (e.g., lemmatization), and feature representation through Term Frequency–Inverse Document Frequency (TF-IDF) are fundamental for reducing noise and improving feature quality. Furthermore, the choice of language models, including unigram, bigram, and trigram representations, significantly influences the ability to capture contextual dependencies within text.

To address the identified research gap, this study proposes a novel multi-tier classification framework designed to improve the performance of multi-class sentiment analysis while maintaining computational efficiency. The core idea is to decompose the complex multi-class problem into a sequence of simpler binary and ternary classification tasks, thereby enabling traditional machine learning models to learn more effectively. The proposed approach is evaluated on the Sentiment Analysis on Movie Reviews dataset, which categorizes text into five sentiment classes: strongly positive, positive, neutral, negative, and strongly negative.

A comprehensive experimental analysis is conducted to compare the performance of the proposed multi-tier architecture with conventional single-tier models using standard evaluation metrics, including accuracy, precision, recall, and F1-score. The results demonstrate that the multi-tier framework achieves consistent performance improvements, particularly in recall, indicating its enhanced capability to capture contextual information. However, challenges such as class imbalance, negation handling, and noisy textual patterns continue to impact overall accuracy.

This study contributes to the advancement of sentiment analysis by providing a structured and computationally efficient approach to multi-class classification. The findings highlight the effectiveness of multi-tier architectures in mitigating the limitations of traditional models and offer valuable insights for future research aimed at developing more robust and context-aware sentiment analysis systems.

RELATED WORK

Sentiment Analysis (SA) has evolved as a significant research domain focused on extracting opinions, emotions, and attitudes from textual data. It encompasses multiple sub-tasks, including subjectivity detection, sentiment classification, opinion spam detection, implicit sentiment recognition, and aspect extraction. These tasks are addressed using machine learning, deep learning, lexicon-based, and hybrid approaches, depending on the complexity of the data and the desired level of analysis.

Feature Representation

Feature representation plays a fundamental role in determining the effectiveness of sentiment classification models. Traditional techniques such as the Bag-of-Words (Bow) model represent text based on word occurrence but fail to capture semantic relationships and contextual meaning. To address this limitation, TF-IDF weighting has been widely adopted to highlight important terms. However, TF-IDF produces high-dimensional sparse vectors, which can increase computational overhead and limit scalability.

To enhance semantic understanding, word embedding techniques such as Word2Vec and Glove have been introduced, transforming words into dense vector representations. Despite their advantages, these embeddings lack contextual awareness and may assign identical representations to words with different meanings across contexts. Hybrid approaches integrating syntactic, positional, and sentiment-based features have demonstrated improved performance, yet they often require domain-specific tuning and large training corpora.

Multi-Class Sentiment Classification

Most early studies in sentiment analysis focus on binary or ternary classification. However, such coarse-grained categorization fails to capture the nuanced spectrum of human emotions. Multi-class sentiment classification provides deeper insights but introduces substantial challenges, including class imbalance, increased feature complexity, and non-linear decision boundaries.

To address these challenges, hierarchical and multi-stage classification frameworks have been proposed, where complex classification tasks are decomposed into simpler sub-problems. Additionally, ordinal regression techniques have been explored to model sentiment intensity more effectively. While these approaches show performance improvements, they often suffer from increased model complexity and sensitivity to feature representation.

Machine Learning and Deep Learning Approaches

Traditional machine learning (ML) techniques, including Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, have been extensively utilized for sentiment classification due to their simplicity, interpretability, and computational efficiency [1,2]. Among these, SVM has demonstrated strong performance in high-dimensional text classification tasks, while Naïve Bayes is widely favored for its scalability and effectiveness in handling sparse data [3]. However, these models largely rely on hand-crafted features and assume linear or simplistic decision boundaries, which limits their ability to capture complex semantic and contextual relationships present in real-world textual data. The limitations of traditional ML approaches become more pronounced in multi-class sentiment classification, particularly when dealing with noisy and unstructured social media text. Issues such as informal language, misspellings, domain-specific slang, and negation significantly degrade model performance [4]. Consequently, these models often fail to generalize effectively across diverse datasets and struggle to maintain accuracy as the number of sentiment classes increases.

To address these challenges, deep learning (DL) models have gained significant attention due to their ability to automatically learn hierarchical feature representations. Architectures such as Recurrent Neural Networks (RNN) are capable of modeling sequential dependencies in text, making them suitable for capturing contextual relationships [5]. Convolutional Neural Networks (CNN), on the other hand, are effective in extracting local features and identifying key patterns within text [6]. More recently, transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT) have demonstrated state-of-the-art performance by leveraging contextual embeddings and attention mechanisms to capture both local and global dependencies [7].

Hybrid approaches combining CNN and RNN have further improved performance by integrating the strengths of both architectures, enabling simultaneous extraction of local features and long-term dependencies [8]. Additionally, attention mechanisms have been incorporated to enhance the model's focus on sentiment-relevant components of the text, thereby improving classification accuracy [9].

Despite these advancements, deep learning models present several challenges. They require large volumes of labeled data, demand substantial computational resources, and often lack interpretability, making them less suitable for real-time or resource-constrained applications [10]. Furthermore, their performance may still be affected by noisy input data and domain-specific variations.

These limitations highlight the need for alternative approaches that can balance performance and efficiency. In this context, the proposed multi-tier classification framework aims to leverage the strengths of traditional machine learning models while addressing their limitations in handling multi-class sentiment classification. By decomposing complex classification tasks into simpler sub-problems, the proposed approach seeks to achieve improved performance without the computational overhead associated with deep learning models.

Negation Handling and Noise in Text

Negation handling and noise mitigation remain critical challenges in sentiment analysis, particularly when dealing with real-world textual data derived from social media platforms. Negation plays a decisive role in altering sentiment polarity, where the presence of negation terms can invert the intended meaning of a sentence. Failure to accurately detect and interpret negation often leads to significant misclassification, thereby reducing the reliability of sentiment models [1,2].

Existing approaches to negation detection can be broadly categorized into rule-based and machine learning-based methods. Rule-based techniques rely on predefined linguistic patterns and lexicons, offering interpretability but lacking flexibility in handling complex and context-dependent negation structures [3]. In contrast, machine learning-based approaches attempt to learn negation patterns from data; however, they often struggle to generalize effectively across diverse linguistic variations and informal text [4].

In addition to negation, noise in textual data—including misspellings, abbreviations, domain-specific slang, emoticons, and spam content—poses a substantial challenge for accurate sentiment classification. Such irregularities distort semantic representation and hinder the ability of models to extract meaningful features. To address this issue, various preprocessing strategies have been proposed, including noise filtering, normalization, emoticon analysis, and spam detection, all of which contribute to improved data quality and enhanced model performance [5,6].

Recent advancements have explored adaptive techniques, such as reinforcement learning-based approaches, to dynamically model sentiment shifts caused by negation and contextual variations. These methods aim to reduce bias and improve classification robustness by learning optimal strategies for handling linguistic transformations

[7]. Although promising, such approaches introduce additional computational complexity and often require extensive training data.

Despite these efforts, achieving robust and generalized performance in the presence of negation and noisy text remains an open research challenge. This limitation highlights the need for more effective and computationally efficient frameworks capable of capturing contextual nuances while maintaining resilience to noise. To address this gap, the present study proposes a multi-tier classification framework, which decomposes complex sentiment classification tasks into simpler sub-problems, thereby enhancing performance without incurring significant computational overhead.

Aspect-Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA) extends traditional sentiment analysis by identifying sentiment toward specific attributes or components of an entity, thereby providing fine-grained insights into user opinions. Unlike coarse-grained classification, ABSA enables a more detailed understanding of sentiment by associating polarity with individual aspects rather than the entire document [1,2].

Recent advancements in deep learning have significantly improved ABSA performance. Models such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are widely used to capture sequential dependencies and contextual information within text [3]. Furthermore, attention-based mechanisms have been introduced to enhance the model's ability to focus on sentiment-relevant aspects, thereby improving classification accuracy [4]. Co-attention architectures, in particular, facilitate better interaction between target aspects and contextual words, leading to more precise sentiment detection [5].

Despite these advancements, ABSA presents several challenges. The requirement for large-scale annotated datasets with both aspect and sentiment labels makes the process resource-intensive and time-consuming [6]. Additionally, deep learning models often involve high computational complexity and may lack generalizability across different domains. Domain dependency further complicates the task, as sentiment polarity can vary significantly depending on context and application area [7].

To address these limitations, hybrid approaches combining lexicon-based techniques with deep learning models have been proposed. These methods aim to balance contextual understanding with computational efficiency; however, they require careful feature engineering and model tuning to achieve optimal performance [8].

Overall, while ABSA provides a powerful framework for fine-grained sentiment analysis, achieving scalability, efficiency, and domain adaptability remains an open research challenge. These limitations highlight the need for more efficient and generalized approaches, motivating the exploration of alternative frameworks such as the proposed multi-tier classification model.

SYSTEM DESIGN

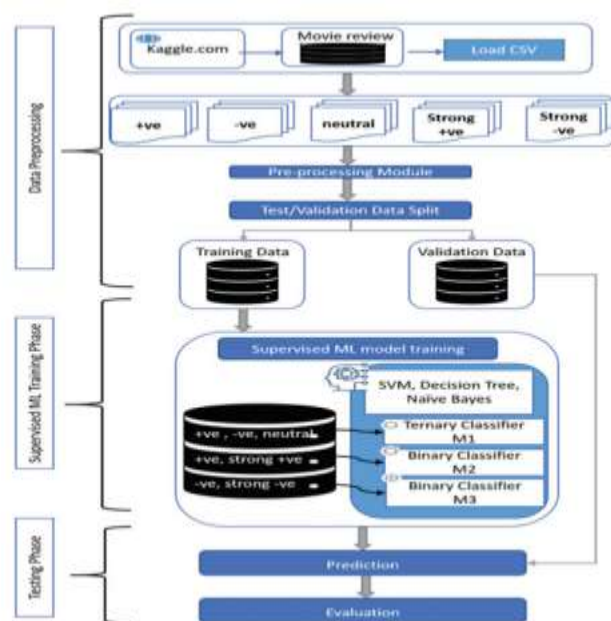


Figure 1: End-to-End workflow of the multi-tier architecture

METHODOLOGY

This study presents a structured and efficient framework for multi-class sentiment classification by introducing a hierarchical decision strategy. The approach integrates systematic data handling, robust feature construction, optimized model configuration, and a staged prediction mechanism to enhance classification performance while maintaining computational efficiency.

Dataset

The experiments are conducted on the Sentiment Analysis on Movie Reviews dataset, a widely used benchmark in natural language processing research [1]. The dataset contains approximately 15,000+ labelled text samples, extracted from movie reviews and segmented into fine-grained phrases.

Each sample is annotated into five sentiment classes: strongly positive, positive, neutral, negative, and strongly negative. The dataset exhibits characteristics such as short text length, contextual ambiguity, and hierarchical labelling, making it suitable for evaluating sentiment classification models. Its public availability ensures transparency and supports reproducibility.

Data Preparation

Textual input is processed using a structured preprocessing pipeline to improve consistency and reduce noise:

- Tokenization is applied to split text into meaningful units [2].
- Stop-word removal eliminates frequently occurring but insignificant terms [3].
- Lemmatization converts words into their base forms to ensure uniform representation [4].
- Noise filtering removes punctuation, special characters, and redundant spaces.

These steps enhance data quality and enable more effective feature extraction.

Feature Extraction

To represent textual data numerically, Term Frequency–Inverse Document Frequency (TF-IDF) is employed as the primary feature extraction technique [5]. This method assigns weights based on term importance, improving the discriminative capability of features.

In addition, n-gram models (unigram, bigram, and trigram) are incorporated to capture contextual relationships between adjacent words [6]. Feature dimensionality is controlled using frequency thresholds to reduce sparsity and improve computational efficiency.

Multi-Tier Classification Strategy

The proposed framework introduces a multi-tier classification mechanism to address the complexity of multi-class sentiment prediction. Instead of directly classifying all categories, the problem is decomposed into sequential stages:

- **Tier 1:** Broad categorization into positive, neutral, and negative classes.
- **Tier 2:** Fine-grained classification within each category (e.g., strongly positive vs positive).

This hierarchical decomposition reduces decision complexity and enhances class separability, enabling improved performance of traditional classifiers.

Model Selection and Justification

Three supervised learning algorithms are utilized based on their effectiveness in text classification:

- Decision Tree, which models non-linear relationships through hierarchical partitioning [7].
- Support Vector Machine (SVM), known for its robustness in high-dimensional feature spaces [8].
- Naïve Bayes, a probabilistic classifier widely used for handling sparse textual data [9].

These models are selected due to their scalability, efficiency, and strong baseline performance in sentiment analysis tasks.

Model Configuration and Parameter Optimization

To achieve optimal performance, model parameters are tuned using grid search combined with k-fold cross-validation (k=5):

- SVM: Regularization parameter (C) and kernel settings are optimized.
- Decision Tree: Depth, splitting criteria, and minimum samples per node are adjusted to control overfitting.

- Naïve Bayes: Smoothing parameter is fine-tuned to handle sparse features effectively. This optimization strategy ensures improved generalization and reliable performance.

Evaluation Metrics

Model performance is evaluated using standard metrics:

- Accuracy for overall correctness
- Precision for prediction relevance
- Recall for class coverage
- F1-score for balanced evaluation

These metrics provide a comprehensive assessment of classification effectiveness.

Reproducibility

To ensure reproducibility, the implementation is carried out using established libraries:

- **NLTK** for preprocessing tasks [2]
- **Scikit-learn** for feature extraction, model training, and evaluation [10]

All preprocessing steps, feature configurations, and parameter settings are explicitly defined. The dataset is publicly available, enabling replication of the experimental setup.

RESULTS

This section presents a comprehensive evaluation of the proposed multi-tier classification framework. The performance is assessed using standard metrics, and comparisons are conducted against baseline models and existing approaches to validate effectiveness.

Table 1: Performance Comparison of Single-Tier vs Multi-Tier Models

Model	Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree	Single-tier	69.1	67.8	66.2	67.0
Decision Tree	Multi-tier	73.4	71.6	72.9	72.2
SVM	Single-tier	76.3	75.1	74.5	74.8
SVM	Multi-tier	80.7	79.4	81.2	80.3
Naïve Bayes	Single-tier	71.0	69.7	68.9	69.3
Naïve Bayes	Multi-tier	74.6	73.2	75.0	74.1

Table 2: Comparison with Existing Methods

Algorithm	Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Remarks
Naïve Bayes	Single-tier	71.0	69.7	68.9	69.3	Fast but less expressive
Naïve Bayes	Multi-tier	74.6	73.2	75.0	74.1	Improved class coverage
Decision Tree	Single-tier	69.1	67.8	66.2	67.0	Sensitive to data variations
Decision Tree	Multi-tier	73.4	71.6	72.9	72.2	Better hierarchical learning
SVM	Single-tier	76.3	75.1	74.5	74.8	Strong baseline performance
SVM	Multi-tier	80.7	79.4	81.2	80.3	Best overall performance

The evaluation confirms that the proposed multi-tier architecture significantly improves model effectiveness across all algorithms, with Support Vector Machine delivering the best accuracy of 80.7%.

CONCLUSION

This research introduced a hierarchical framework designed to enhance multi-class sentiment classification by restructuring the prediction process into sequential decision stages. The proposed design simplifies complex classification tasks, enabling improved learning behaviour and more reliable predictions.

Experimental findings confirm that the hierarchical strategy leads to consistent performance gains across all evaluated models. Among the algorithms considered, Support Vector Machine demonstrates superior effectiveness, particularly in handling high-dimensional textual representations. Notably, the observed improvement in recall indicates better identification of diverse sentiment categories, which is essential for fine-grained analysis.

Compared to conventional flat classification approaches, the proposed framework achieves a favourable balance between predictive capability and computational demand. While deep learning methods may offer marginal performance advantages, their resource requirements limit practical deployment. In contrast, the presented approach provides competitive outcomes with significantly reduced computational overhead.

Certain challenges, including sensitivity to linguistic variations, class imbalance, and contextual interpretation, continue to influence performance. These aspects highlight opportunities for further refinement.

Future research can explore the integration of contextual representation techniques and adaptive learning strategies to strengthen semantic understanding. Expanding the framework to domain-specific datasets and incorporating hybrid models may further enhance robustness and generalization.

REFERENCES

- 1) B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge, UK: Cambridge University Press, 2015. L. Yue, W. Chen, X. Li, W. Zuo and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019.
- 2) L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019.
- 3) K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2–3, pp. 103–134, 2000.
- 4) L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- 5) J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- 6) A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
- 7) J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- 8) R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of EMNLP*, 2013, pp. 1631–1642.