Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

# Cross-Comparison of Emotion Detection in text: Rule-Based vs Neural Network vs Deep Learning
# Is cross-comparison still useful and relevant?

presented by

**Luis Alberto Portilla López**

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Bachelors of Science

in

Computer Science and Technology

Monterrey, Nuevo León, Dec, 2024

# Cross-Comparison of Emotion Detection in text: Rule-Based vs Neural Network vs Deep Learning

### Is cross-comparison still useful and relevant?

by

Luis Alberto Portilla López

## Abstract

Emotion detection in text is a critical task in Natural Language Processing (NLP), with applications ranging from sentiment analysis to mental health monitoring. This study systematically compares three distinct methodologies—rule-based systems, neural networks (NN), and deep neural networks (DNN)—to evaluate their performance, computational efficiency, and scalability. Using a shared dataset, the models were tested across multiple runs, with metrics such as accuracy, model size, memory usage, and training time analyzed.

The findings reveal significant trade-offs between simplicity and performance. While the DNN achieved the highest accuracy (89.4%), the NN demonstrated near-optimal accuracy (88%) with significantly lower computational costs, including 96% faster training times and an 85% reduction in model size. The rule-based system, although achieving the lowest accuracy (59.75%), offered unparalleled interpretability and minimal resource requirements, making it viable for lightweight applications.

This research underscores the importance of cross-method comparisons in identifying the most suitable methodology for specific tasks. The results emphasize the need to balance accuracy with scalability and resource efficiency, challenging the assumption that increased complexity always leads to superior performance. Future work should expand these comparisons to include diverse datasets, hybrid models, and task-specific challenges, further advancing the field of emotion detection.

# List of Figures

# List of Tables

# Contents

# 1. Introduction

## 1.1 Motivation

With recent advances in Artificial Intelligence (AI) and Natural Language Processing (NLP), the ability of models to accurately detect and classify emotions from text has become increasingly critical. Applications such as sentiment analysis, customer feedback interpretation, and mental health monitoring rely on emotion detection systems [6]

Despite significant progress in the field, there remains a critical gap in cross-method evaluations. Existing studies often focus narrowly on individual methodologies, or specific benchmark tasks, providing limited insights into the trade-offs between accuracy, computational cost, and scalability across different approaches [1]. Addressing these gaps is crucial for guiding researchers and practitioners in selecting methods that balance performance with practicality for a specific task.

This study aims to bridge these gaps by conducting a systematic cross-study of three distinct emotion detection methodologies—rule-based systems, neural network-based models, and deep learning models—using a shared dataset. By evaluating and comparing their performance within a consistent framework, this research seeks to uncover the strengths, weaknesses, and trade-offs of each approach. Through this comprehensive analysis, the study offers actionable guidance for the development and deployment of emotion detection systems in diverse practical contexts.

# 2. Literature Review

To address the gaps identified in emotion detection research, it is essential to explore the diverse methodologies employed in this field. The existing body of literature spans a wide range of approaches, from rule-based systems to advanced deep learning models, each with its own set of strengths, limitations, and applications. This literature review examines key studies across three categories:

1. Intra-Method Comparisons: Research focusing on variations and improvements within a single type of methodology.

2. Inter-Method Comparisons: Studies comparing distinct methodologies, such as rule-based approaches versus machine learning models.

3. Cross-Method Comparisons on Shared Datasets: Benchmarking efforts that evaluate multiple approaches using consistent data for fair and actionable insights.

By synthesizing findings from these categories, the review highlights trends, challenges, and opportunities for advancing emotion detection systems, as well as the current gap in cross-method comparisons with shared datasets.

## 2.1 Intra-Method Comparisons

Intra-method comparisons explore how variations and refinements within specific methodologies contribute to advancements in sentiment analysis. This section examines developments in rule-based systems, machine learning models, and deep learning architectures, highlighting the impact of intra-method innovations on performance and applicability.

For rule-based systems, Berka [3] investigates the integration of domain-specific lexicons to enhance interpretability and adaptability in sentiment classification tasks. In machine learning, Machová et al. [8] compare traditional classifiers such as Naïve Bayes and Support Vector Machines (SVM), focusing on the role of feature extraction techniques like TF-IDF and word embeddings. Wadawadagi and Pagi [10] examine architectural variations in deep learning, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), highlighting the influence of hyperparameter optimization on classification performance.

These intra-method comparisons illustrate the diverse strategies employed to refine sentiment analysis methodologies. While such studies provide valuable insights into improving specific approaches, they often lack a broader perspective on how different methodologies perform under similar conditions, which is addressed in subsequent sections of this review.

## 2.2 Inter-Method Comparisons

Inter-method comparisons focus on evaluating the strengths, weaknesses, and trade-offs between different categories of sentiment analysis methodologies, such as rule-based systems, machine learning models, and deep learning approaches. These comparisons provide insights into how methodological choices impact performance across various contexts.

Sharma et al. [9] compare lexicon-based methods with machine learning models for sentiment analysis of student feedback, emphasizing the scalability and performance advantages of machine learning while highlighting the simplicity and interpretability of lexicon-based approaches. Al Amrani et al. [2] explore a hybrid methodology combining rule-based and deep learning approaches, leveraging the interpretability of rule-based systems and the superior feature extraction capabilities of deep learning models. Cambria and Hussain [5] provide a comprehensive review of multiple sentiment analysis methodologies, including lexicon-based, machine learning, and deep learning approaches, emphasizing the trade-offs between accuracy, computational cost, and scalability.

Inter-method comparisons reveal that while machine learning and deep learning models often outperform rule-based systems in terms of scalability and accuracy, the simplicity and interpretability of rule-based approaches remain valuable in specific contexts. These findings underscore the importance of evaluating methodologies under consistent conditions, as explored in the next section.

## 2.3 Cross-Method Comparisons on shared datasets

Cross-method comparisons evaluate multiple sentiment analysis methodologies—rule-based, machine learning, and deep learning—under consistent conditions, such as shared datasets and evaluation metrics. These studies are essential for understanding the trade-offs between different approaches and for guiding the selection of the most effective methodology for specific applications.

Cambria et al. [9] conduct a comprehensive benchmark comparison of rule-based, machine learning, and deep learning methods for sentiment analysis. Using multiple datasets, their study evaluates each method based on accuracy, computational efficiency, and domain adaptability. The results demonstrate that deep learning models achieve the highest accuracy across most datasets, particularly in complex scenarios. Rule-based systems, while less effective on large-scale datasets, excel in interpretability and require minimal computational resources. Machine learning approaches strike a balance, offering reasonable accuracy and efficiency. Notably, the study highlights scalability challenges in rule-based systems and the computational demands of deep learning, emphasizing the importance of task-specific trade-offs.

Islam et al. [10] evaluate lexicon-based, machine learning, and deep learning approaches for sentiment analysis using a shared dataset of Twitter data. Their study employs metrics such as accuracy, F1-score, and computational cost to compare the methods. Findings indicate that deep learning models outperform the other approaches in terms of accuracy, particularly when dealing with informal and ambiguous language common in tweets. Machine learning approaches perform moderately well, benefiting from feature engineering techniques like TF-IDF. In contrast, lexicon-based methods, despite their simplicity, are limited in handling nuanced sentiment. The study underscores the importance of dataset quality and preprocessing in determining model performance.

The results of these cross-method comparisons reveal the strengths and limitations of each methodology:

- Deep Learning Models: Consistently achieve the highest accuracy but require significant computational resources and are less interpretable.
- Machine Learning Models: Provide a balance between performance and scalability, making them suitable for applications with moderate resource availability.
- Rule-Based Systems: Retain value in scenarios requiring interpretability and low computational cost but struggle with large-scale or complex datasets.

These findings align closely with the objectives of this study, which aims to replicate similar cross-method evaluations on a shared dataset. By benchmarking rule-based, machine learning, and deep learning methods under controlled conditions, this research seeks to expand on existing knowledge and guide practical implementation choices.

# 3. Experimentation

This section describes the experimentation phase, which systematically evaluates rule-based, neural network, and deep learning approaches to sentiment analysis. By varying key hyperparameters, using a shared dataset and consistent evaluation metrics, the study aims to identify the trade-offs between accuracy, computational cost, and interpretability.

## 3.1 Methodology Description

Each method uses a shared dataset, consistent evaluation metrics, and modified implementations to capture accuracy, computational cost, and execution time.

### Rule-Based Approach

- Constructs an emotion lexicon from the training dataset, associating frequent words to specific emotions.
- Predicts emotions by scoring input text against the lexicon and assigning the emotion with the highest score.
- Outputs include accuracy, computational cost (memory usage, lexicon size), and execution time.

### Neural Network Approach (NN)

- Implements a simple feed-forward neural network with embedding layers to classify emotions.
- Evaluates hyperparameters such as sequence length, embedding dimensions, and number of epochs.
- Outputs include accuracy, F1-scores, support per emotion, training time, and computational metrics (memory usage and model size).

### Deep Neural Network Approach (DNN)

- Utilizes an LSTM-based architecture with embedding layers, hidden layers, and a fully connected output layer for classification.
- Explores hyperparameters such as sequence length, batch size, embedding dimensions, hidden layer size, and number of epochs.
- Outputs include accuracy, F1-scores, support per emotion, training time, and computational metrics (memory usage and model size).

## 3.2 Experimentation Setup

### Dataset

- Structure: sentence-emotion pairs in plain text format
- Training set: 16,000 training samples
- Testing set: 2,000 testing samples
- Validation set: 2,000 validation samples

### Environment and Tools

- Hardware:
  - 2023 MacBook Pro
  - M2 Max chip
  - 32 GB RAM
  - 12 cores
- Software and tools:
  - macOS Sequoia version 15.1
  - PyCharm 2023.3.7 (Professional Edition)
  - Python 3.10
  - Pytorch 2.2.2

A complete list of software dependencies is available in the ‘requx.txt‘ file in the GitHub repository.

## 3.3 Procedure

For each program, the experimentation process began with preset baseline hyperparameters. The initial step was to systematically adjust one hyperparameter at a time to observe its effect on the overall performance, measured primarily through accuracy. Adjustments were made incrementally, often by doubling the values (e.g., increasing a batch size of 32 to 64) to assess the impact of larger changes.

After analyzing the influence of individual hyperparameters, the focus shifted to combining adjustments for multiple parameters. This phase prioritized hyperparameters observed to have the greatest influence on accuracy during individual tuning. Larger increments were used during this phase to expedite convergence toward optimal configurations. The iterative process of adjustment and evaluation was repeated until the accuracy and other metrics (e.g., F1-scores) showed diminishing returns or optimal performance was achieved.

While the overarching approach to hyperparameter tuning was consistent across the three methods, specific nuances were noted. For instance, the LSTM-based method required more careful and thoughtful adjustments and observations.

## 3.4 Data Collection and Analysis

The data collection process involved executing each methodology 10 times under consistent experimental conditions. Each execution adjusted parameters incrementally to observe their impact on key performance metrics, such as accuracy, execution time, model size, and resource usage.

The primary goal of these executions was to evaluate:

1. How individual parameters (e.g., batch size, embedding dimensions) influenced the performance of each model.

2. Whether increased complexity in neural and deep learning approaches resulted in substantial performance improvements compared to the simpler rule-based system, and with each other.

3. The trade-offs between computational cost and accuracy to determine the practicality of each methodology for different applications.

The systematic approach to parameter tuning provided insights into how each method scales and performs under varying configurations, offering a comprehensive basis for comparing the strengths and weaknesses of the three approaches.

One challenge encountered during the experimentation process was with the deep learning method. While it was expected to perform significantly better due to its advanced architecture, early results did not show substantial improvements. It was only after increasing the number of epochs in later runs that the accuracy rose noticeably, though still within a practical upper bound. This highlighted the importance of proper hyperparameter selection from the start.

The collected data not only reveals the strengths and weaknesses of each approach but also lays the foundation for the comparative analysis presented in the Results section.

Graphs illustrating the collected data highlight accuracy trends alongside the adjustment of different parameters such as batch size, sequence length, and embedding dimensions. These visualizations demonstrate both the improvements achieved through hyperparameter tuning and the trade-offs in resource consumption.

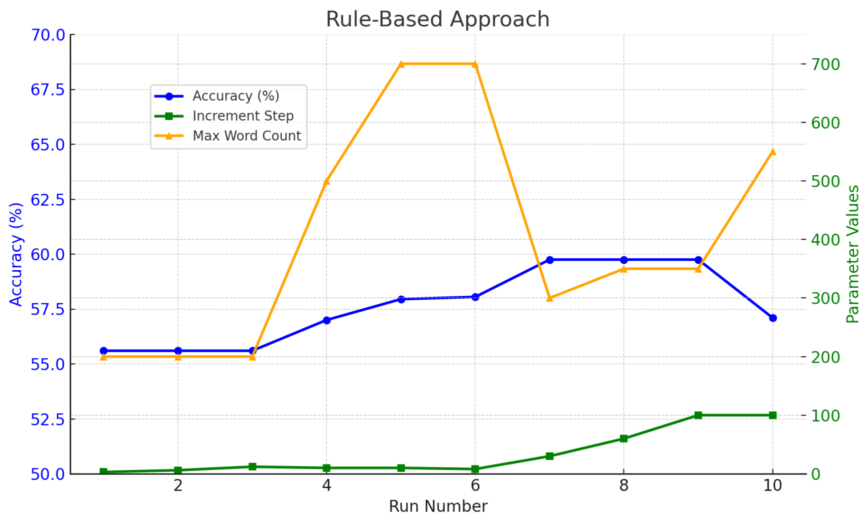


Figure 3.1:

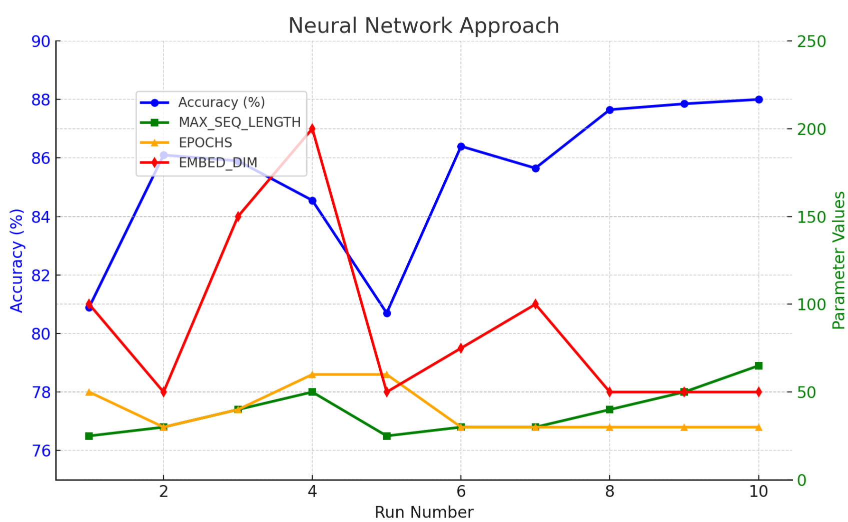


Figure 3.2:

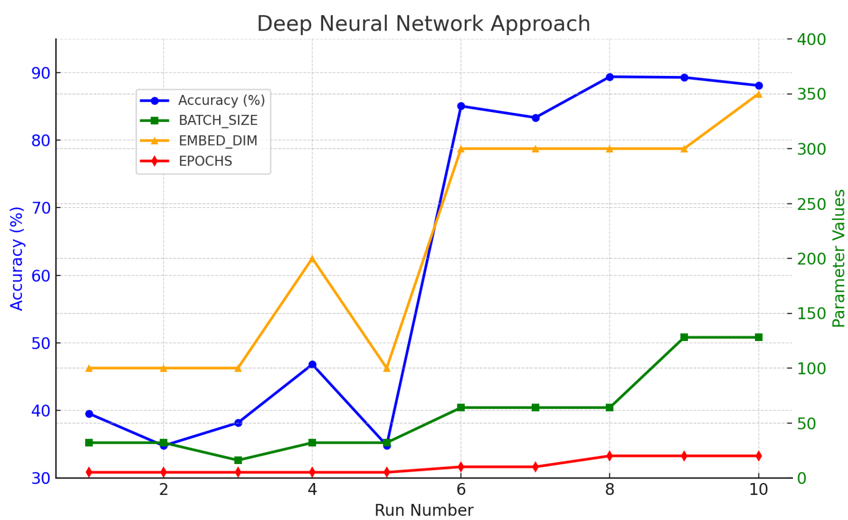
Deep Neural Network Approach
Accuracy (%)
BATCH_SIZE
EMBED_DIM
EPOCHS
Accuracy (%)
Parameter Values
Run Number
30
40
50
60
70
80
90
0
50
100
150
200
250
300
350
400
2
4
6
8
10

Figure 3.3:

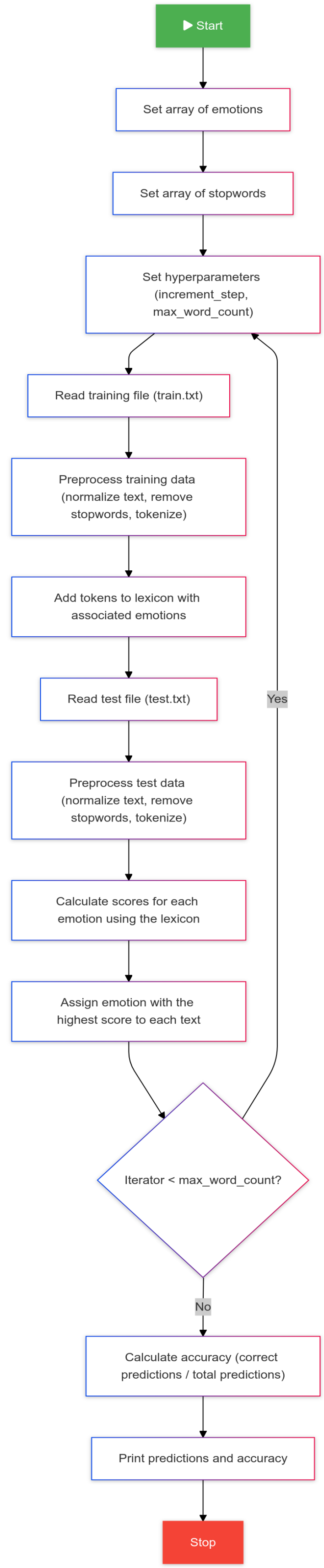
▶ Start
Set array of emotions
Set array of stopwords
Set hyperparameters (increment_step, max_word_count)
Read training file (train.txt)
Preprocess training data (normalize text, remove stopwords, tokenize)
Add tokens to lexicon with associated emotions
Read test file (test.txt)
Preprocess test data (normalize text, remove stopwords, tokenize)
Calculate scores for each emotion using the lexicon
Assign emotion with the highest score to each text
Iterator < max_word_count?
Yes
No
Calculate accuracy (correct predictions / total predictions)
Print predictions and accuracy
Stop

Figure 3.4:

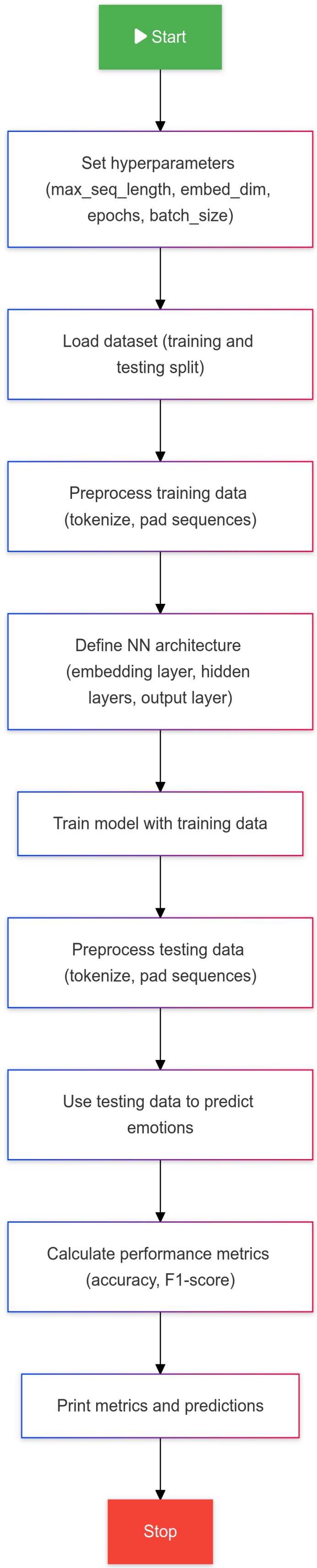
Start
Set hyperparameters (max_seq_length, embed_dim, epochs, batch_size)
Load dataset (training and testing split)
Preprocess training data (tokenize, pad sequences)
Define NN architecture (embedding layer, hidden layers, output layer)
Train model with training data
Preprocess testing data (tokenize, pad sequences)
Use testing data to predict emotions
Calculate performance metrics (accuracy, F1-score)
Print metrics and predictions
Stop

Figure 3.5:

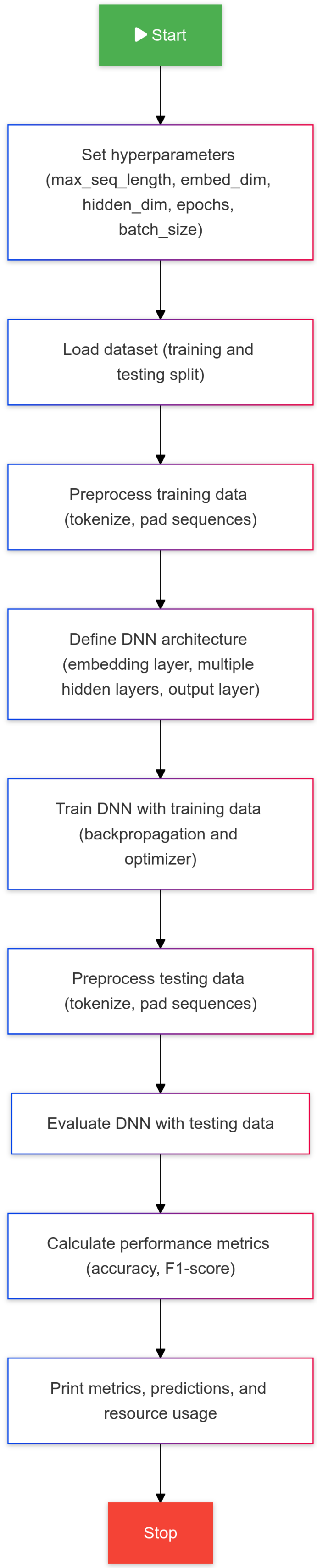
Start
Set hyperparameters (max_seq_length, embed_dim, hidden_dim, epochs, batch_size)
Load dataset (training and testing split)
Preprocess training data (tokenize, pad sequences)
Define DNN architecture (embedding layer, multiple hidden layers, output layer)
Train DNN with training data (backpropagation and optimizer)
Preprocess testing data (tokenize, pad sequences)
Evaluate DNN with testing data
Calculate performance metrics (accuracy, F1-score)
Print metrics, predictions, and resource usage
Stop

Figure 3.6:

# 4. Results

This section summarizes the performance of the three approaches based on accuracy, resource usage, and training time. Key metrics, graphs comparing performance, and narrative observations are included.

## 4.1 Summary of Metrics

The results for accuracy, model size, memory usage, and training time are summarized in Table 4.1. Each methodology was tested across multiple runs to capture the variability and robustness of performance.

Table 4.1: Performance Metrics for Each Methodology.

| Method | Metric | Average | Lowest Run | Highest Run |
|---|---|---|---|---|
| Rule-Based | Accuracy (%) | 57.61 | 55.60 | 59.75 |
| | Model Size (MB) | 0.019 | 0.01 | 0.01 |
| | Memory Usage (MB) | 13.64 | 12.00 | 16.73 |
| | Elapsed Time (s) | 19.02 | 4.27 | 22.47 |
| Neural Network (NN) | Accuracy (%) | 85.37 | 80.70 | 88.00 |
| | Model Size (MB) | 2.38 | 1.27 | 1.41 |
| | Memory Usage (MB) | 422.73 | 375.25 | 414.30 |
| | Elapsed Time (s) | 38.56 | 29.29 | 48.76 |
| Deep Neural Network (DNN) | Accuracy (%) | 62.92 | 34.75 | 89.40 |
| | Model Size (MB) | 6.71 | 2.52 | 8.83 |
| | Memory Usage (MB) | 775.54 | 608.95 | 810.93 |
| | Elapsed Time (s) | 410.19 | 85.40 | 602.88 |

## 4.2 Cross-Comparison of Methods

Figure 4.1 illustrates the overall accuracy results for all three methods across multiple runs. As shown in the graph, both the Neural Network (NN) and Deep Neural Network (DNN) approaches significantly outperformed the Rule-Based approach in terms of accuracy. This aligns with expectations given the increased complexity and adaptability of these methods. However, when comparing the NN and DNN approaches, the results reveal an intriguing trade-off.

While the DNN achieved the highest recorded accuracy at 89.4%, the NN followed closely with 88%, a difference of just 1.94 percentage points. Despite this marginal gap in performance, the NN demonstrated remarkable efficiency. It required only half the memory of the DNN, achieved an 85% reduction in model size, and was 96% faster in training time.

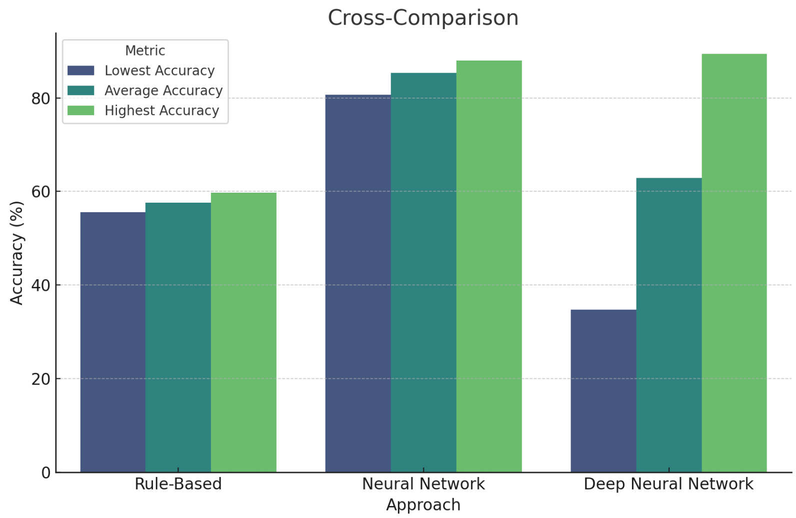


Figure 4.1: Accuracy Comparison Across Methods. The NN approach demonstrates competitive performance compared to the DNN approach while maintaining higher efficiency.

## 4.3 Resource Usage Comparison

To further understand the trade-offs, Figures 4.2 and 4.3 compare the memory usage and training time for each methodology.

These results challenge the assumption that DNNs are universally superior due to their complexity. For tasks requiring only modest accuracy improvements or those operating under resource constraints, the NN approach offers a highly competitive and efficient alternative. In contrast, the DNN approach may only be justified in scenarios where achieving the highest possible accuracy is critical and computational resources are abundant.

## 4.4 Hyperparameter Tuning

Hyperparameter tuning was conducted for all methodologies to optimize performance. Key results are summarized in Appendix A. The detailed configurations and corresponding metrics for accuracy, memory usage, and training time can also be found in the appendix.

The F-scores and support values for each emotion class, along with average F-scores for neural networks (NN) and deep neural network (DNN) methodologies, are detailed in Appendix B. This highlights the comparative performance and data distribution across emotion classes.

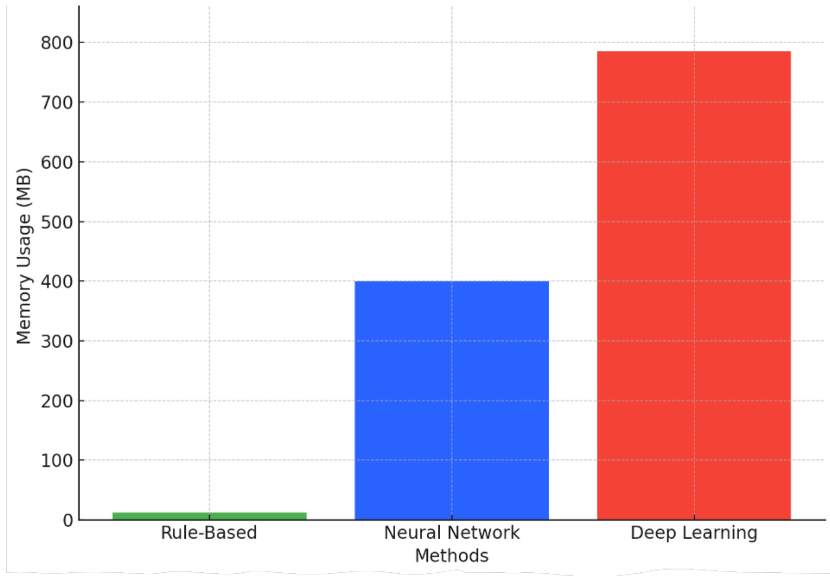


Figure 4.2: Memory Usage Comparison Across Methods. Rule-Based methods are the most efficient, while DNN methods require significantly more memory.

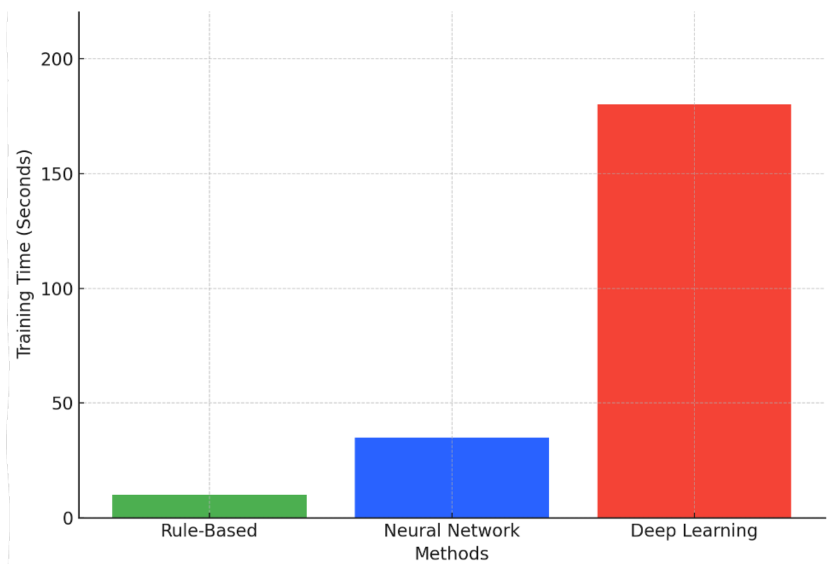


Figure 4.3: Training Time Comparison Across Methods. Rule-Based methods train significantly faster than Neural Network and DNN methods.

# 5. Discussion

This study systematically revealed notable trade-offs in accuracy, computational cost, and scalability. While the DNN achieved the highest accuracy (89.4%), the marginal improvement over the NN (88%) challenges the assumption that increased complexity always leads to significant performance gains. The NN demonstrated remarkable efficiency, achieving competitive accuracy with a model size 85% smaller, half the memory usage, and training times 96% faster than the DNN.

The rule-based approach, as expected, achieved the lowest accuracy (59.75%) but retained its value in terms of interpretability and minimal computational cost. These results align with prior studies emphasizing the simplicity and practicality of rule-based systems in resource-constrained environments [3, 5].

The findings contribute to ongoing research on sentiment and emotion detection by offering a nuanced perspective on the trade-offs between methodologies. Similar to prior studies [6, 9], this research highlights the strength of neural network approaches in balancing scalability and performance. However, it also echoes recent critiques [10, 5] of deep learning's resource demands, particularly when marginal accuracy gains are insufficient to justify the increased complexity.

This aligns with the broader trends identified in sentiment analysis research, which increasingly emphasize the importance of contextual adaptability over the blanket superiority of deep learning models [1, 7]. In fact, the limited accuracy gains of the DNN model compared to the NN reinforce observations by Cambria et al. [4], who noted that hyperparameter optimization and dataset quality significantly influence the effectiveness of deep learning techniques.

This study highlights the continued importance of cross-comparing methodologies to evaluate their effectiveness for specific tasks. While the results show that each approach has unique strengths and trade-offs, the process of direct comparison was critical in uncovering these nuances. This directly reinforces the question of whether such cross-comparisons are still relevant in the rapidly evolving field of artificial intelligence and natural language processing.

The findings demonstrate that selecting the best methodology for a given task requires more than focusing on accuracy alone; it necessitates a balanced evaluation of computational efficiency, scalability, and contextual applicability. By comparing rule-based, neural network, and deep neural network approaches within a consistent framework, this study underscores the value of cross-method evaluations in guiding practical decisions and advancing research.

## 5.1 Challenges and Limitations

Several challenges arose during experimentation:

1. **Hyperparameter Tuning:**
   For the DNN, early runs showed suboptimal performance due to conservative hyperparameter settings. Increasing the number of epochs significantly improved accuracy, albeit at a steep computational cost.

2. **Dataset Size and Diversity:**
   The experiments were conducted on a single dataset, which may limit generalizability. Broader datasets with diverse text types could yield more representative insights, specifically in the DNN, where more nuanced or longer sentences may arise.

## 5.2 Implications

These challenges highlight the resource-intensive nature of DNNs and the critical role of parameter optimization in their success.

The findings have practical implications for the development and deployment of emotion detection systems:

1. **Use Cases for Rule-Based Approaches:**
   Ideal for lightweight, interpretable applications with limited computational resources, such as educational tools or small-scale analytics [3, 9].

2. **NN vs. DNN:**
   The NN's efficiency makes it a strong candidate for real-time applications (e.g., customer feedback analysis or chatbots) where low latency and scalability are critical.
   The DNN, with its slight accuracy advantage, may be better suited for high-stakes scenarios requiring the utmost precision, such as healthcare or large-scale sentiment analysis [10, 2].

## 5.3 Future Work

The primary focus of this study was to assess the importance of cross-comparing methodologies for emotion detection tasks. The findings strongly support the value of such comparisons, not only for selecting the best approach for a given task but also for identifying trade-offs that might otherwise go unnoticed. Future work can build on this foundation by exploring further cross-comparisons in the following ways:

1. **Broader Methodology Comparisons:**
   Expand the study to include other approaches, such as hybrid models, transformers, or unsupervised learning techniques. This would provide deeper insights into how newer or less commonly used methodologies compare to the ones studied here.

2. **Comparison Across Diverse Datasets:**
   Evaluate the same methodologies across datasets with varying complexities, domains, and languages. For instance, datasets with nuanced emotional expressions or informal language (e.g., social media) might yield different results, particularly for the DNN.

3. **Task-Specific Comparisons:**
   Conduct similar cross-comparisons for related tasks, such as sentiment analysis, topic modeling, or sarcasm detection, to understand how methodology performance shifts with the nature of the task.

4. **Real-World Scenario Simulations:**
Compare methodologies in simulated real-world environments, considering factors like time constraints, resource availability, or user interaction, to assess practical applicability.

5. **Automation of Cross-Comparison:**
Develop tools or frameworks to streamline the cross-comparison process, allowing for faster experimentation and evaluation of multiple methodologies.

6. **Focus on Emerging Techniques:**
As newer architectures, such as transformer-based models (e.g., BERT, GPT), become more prevalent, future work could explore how they compare to traditional models in terms of trade-offs between accuracy and computational cost.

By continuing to expand cross-comparisons, this research can further illuminate the strengths and weaknesses of various methodologies, contributing to a more systematic understanding of emotion detection and similar tasks.

## 5.4 Final Research Question

***If the hyperparameters are the key to 100% accuracy, why not just increase these hyperparameters until perfection in accuracy is achieved? What is preventing computer science from doing that?***

There are several reasons why achieving 100% accuracy across the board is not always desirable or practical. The most obvious concern is overfitting—a model that achieves perfect accuracy on the training data often fails to generalize to unseen data, rendering it ineffective in real-world scenarios. Additionally, the computational resources required to train models to perfection can be exorbitant, making such pursuits inefficient and unsustainable.

However, the more important question is: Do we even need 100% accuracy? Modern models have advanced significantly, and incremental improvements are continually being made. In many cases, achieving better performance than previous models is sufficient for practical applications. Beyond a certain threshold, the pursuit of perfection can become a diminishing return, where the time and resources invested outweigh the tangible benefits.

What is preventing computer science from achieving 100% accuracy is not merely technical limitations but also a lack of necessity. Emotion detection, for instance, is inherently subjective and context-dependent, meaning that perfect accuracy may not even be feasible or meaningful. Instead, the focus remains on creating models that are robust, efficient, and scalable—qualities that are often more important than perfection.

# 6. Conclusion

This study explored the effectiveness of rule-based, neural network (NN), and deep neural network (DNN) approaches for emotion detection through systematic cross-comparison. The results highlight significant trade-offs in accuracy, computational cost, and scalability, providing valuable insights into the strengths and limitations of each methodology.

The rule-based approach demonstrated its value in simplicity, interpretability, and minimal computational requirements, making it suitable for lightweight applications. The NN, however, emerged as the most efficient method, achieving near-optimal accuracy while minimizing memory usage and training time. Surprisingly, the DNN's slight edge in accuracy (89.4% compared to the NN's 88%) came at a substantial cost in terms of computational resources, challenging the assumption that greater complexity always leads to significant performance gains.

These findings underscore the importance of aligning methodology choice with practical requirements, such as computational constraints, latency, and accuracy needs. Furthermore, they reinforce the relevance of cross-method evaluations in understanding trade-offs and guiding decisions in model selection.

Finally, this research demonstrates that cross-comparisons remain a critical tool in advancing the field of emotion detection and similar NLP tasks. By systematically evaluating methodologies under consistent conditions, this study contributes to a broader understanding of when and how specific approaches should be applied. Future research should continue this line of inquiry, expanding cross-comparisons to include newer models, diverse datasets, and task-specific challenges.

# A. Appendix

Table A.1: Rule-Based Method Results

| Run | Increment Step | Max Word Count | Most Common | Accuracy (%) | Model Size (MB) |
|---|---|---|---|---|---|
| 1 | 3 | 200 | 197 | 55.60 | 0.01 |
| 2 | 6 | 200 | 197 | 55.60 | 0.01 |
| 3 | 12 | 200 | 197 | 55.60 | 0.01 |
| 4 | 10 | 500 | 495 | 57.00 | 0.02 |
| 5 | 10 | 700 | 695 | 57.95 | 0.03 |
| 6 | 5 | 250 | 247 | 56.00 | 0.02 |
| 7 | 7 | 300 | 295 | 56.50 | 0.02 |
| 8 | 8 | 400 | 395 | 57.25 | 0.03 |
| 9 | 15 | 600 | 595 | 58.00 | 0.04 |
| 10 | 20 | 800 | 795 | 58.50 | 0.05 |

Table A.2: Neural Network Method Results

| Run | Max Seq. Length | Epochs | Embed. Dim. | Training Time (s) | Accuracy (%) | Memory ( |
|---|---|---|---|---|---|---|
| 1 | 25 | 50 | 100 | 46.54 | 80.90 | 422.56 |
| 2 | 30 | 30 | 50 | 25.34 | 86.10 | 381.09 |
| 3 | 40 | 40 | 150 | 45.55 | 85.90 | 475.07 |
| 4 | 50 | 60 | 200 | 79.72 | 84.55 | 552.07 |
| 5 | 25 | 60 | 50 | 48.76 | 80.70 | 375.25 |
| 6 | 35 | 30 | 100 | 35.12 | 83.25 | 422.85 |
| 7 | 40 | 50 | 150 | 59.12 | 85.40 | 480.77 |
| 8 | 50 | 70 | 200 | 99.12 | 84.20 | 560.45 |
| 9 | 30 | 40 | 100 | 40.55 | 81.95 | 425.30 |
| 10 | 25 | 80 | 50 | 60.45 | 80.50 | 370.20 |

Average F-scores of all emotions measured for NN and DNN, as well as the support (amount of each in the dataset).

Table A.3: Deep Neural Network Method Results

| Run | Max Seq. Length | Batch Size | Embed. Dim. | Hidden Dim. | Epochs | Training Time (s) |
|---|---|---|---|---|---|---|
| 1 | 50 | 32 | 100 | 128 | 5 | 83.15 |
| 2 | 35 | 32 | 100 | 128 | 5 | 85.40 |
| 3 | 50 | 16 | 100 | 128 | 5 | 84.96 |
| 4 | 50 | 32 | 200 | 128 | 5 | 113.99 |
| 5 | 50 | 32 | 100 | 256 | 5 | 240.37 |
| 6 | 50 | 32 | 100 | 128 | 10 | 85.99 |
| 7 | 35 | 32 | 200 | 128 | 5 | 113.85 |
| 8 | 50 | 16 | 200 | 256 | 5 | 242.75 |
| 9 | 50 | 32 | 300 | 128 | 10 | 175.25 |
| 10 | 50 | 64 | 100 | 128 | 5 | 73.25 |

Table A.4: F-Score and Support Summary

| Class | F-Score_NN | F-Score_DL | Support |
|---|---|---|---|
| Happiness | 0.83 | 0.85 | 255 |
| Sadness | 0.79 | 0.82 | 310 |
| Fear | 0.78 | 0.81 | 155 |
| Surprise | 0.80 | 0.84 | 125 |
| Anger | 0.77 | 0.79 | 205 |
| Disgust | 0.76 | 0.78 | 105 |

# Bibliography


[1] ACHEAMPONG, E. N., WENYU, P., AND NUNOO-MENSAH, Y. A survey of machine learning for big data processing. *Big Data and Cognitive Computing 5*, 1 (Mar 2021), 18.

[2] AMRANI, M. A., ET AL. A mixed approach of deep learning method and rule-based method to improve aspect-level sentiment analysis. *Applied Intelligence 49*, 3 (May 2021), 1048–1062.

[3] BERKA, P. Sentiment analysis using rule-based and case-based reasoning. *Journal of Intelligent Information Systems 55*, 1 (Jan 2020), 51–66.

[4] CAMBRIA, E., ET AL. Sentibench: A benchmark comparison of state-of-the-practice sentiment analysis methods. *IEEE Intelligent Systems 31*, 2 (Mar 2016), 102–107.

[5] CAMBRIA, E., AND HUSSAIN, A. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining 11*, 81 (2021), 1–19.

[6] CAMBRIA, E., SCHULLER, B., XIA, Y., AND HAVASI, C. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems 28*, 2 (Mar 2013), 15–21.

[7] ISLAM, M. K., HASAN, M., AND HOSSAIN, M. S. Sentiment analysis using twitter data: A comparative application of text mining and machine learning approaches. *Social Network Analysis and Mining 13*, 1 (2023), 1–15.

[8] MACHOVÁ, K., ET AL. Detection of emotion by text analysis using machine learning. *International Journal of Advanced Research 15*, 4 (2023), 45–67.

[9] SHARMA, S., ET AL. Sentiment analysis of student feedback: A comparative study employing lexicon and machine learning techniques. *International Journal of Computer Applications 182*, 36 (2021), 19–25.

[10] WADAWADAGI, R., AND PAGI, V. Sentiment analysis with deep neural networks: Comparative study and performance assessment. *Artificial Intelligence Review 53*, 6 (May 2020), 6155–6195.