Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 1s (2025)

Dynamic Lexicon-Based Sentiment Analysis Architecture Using Nonlinear Feature Optimization

[1]V R N S S V Saileela P, [2]Dr. N. Naga Malleswara Rao

[1]Research scholar, Department of CSE, ANU College of Engineering &Technology, Guntur, AP, India.
Email: saileelavenkata@gmail.com

[2]Professor, Department of Information Technology, RVR & JC College of Engineering &Technology, Guntur, AP, India,
Email: nnmrao@rvrjc.ac.in

Article History:

Received: *30-08-2024*

Revised: *16-10-2024*

Accepted: *01-11-2024*

Abstract:

Dynamic Lexicon-Based Sentiment Analysis (DLSA) improves sentiment detection by using adaptive, nonlinear feature optimization techniques. The model targets gaps in traditional methods, which often miss complex and subtle sentiment patterns. DLSA integrates modules for text preprocessing, lexicon and phrase extraction, and feature optimization with genetic algorithms. The system refines feature sets by selecting, combining, and adjusting features based on real-time performance data. Key findings show DLSA outperforms traditional models like SLPCF in accuracy, precision, and recall, especially in analyzing nuanced text data. This approach provides a flexible and efficient solution, enhancing the accuracy of sentiment classification in varied applications.

Keywords: Dynamic Lexicon-Based Sentiment Analysis, Feature Optimization, Genetic Algorithm, Sentiment Lexicons, Adaptive Feedback, Machine Learning, Sentiment Classification

1. Introduction

Sentiment analysis helps to understand opinions in texts like reviews and social media posts. Traditional sentiment analysis uses deep learning, which often needs heavy computation and lacks transparency. Lexicon-based methods, however, are simpler and easier to understand because they use predefined word dictionaries to identify emotions. These methods, like the SO-CAL approach, perform well in some datasets and show that lexicon-based sentiment analysis can work well in specific contexts [1]. In Arabic texts, frameworks like LSAnArTe use advanced lexicons and lemmatization techniques to achieve high accuracy, highlighting the adaptability of lexicon-based approaches in non-English languages [2].

However, lexicon-based methods struggle with complex language patterns and dynamic data. To improve these methods, feature optimization techniques are combined with lexicon-based sentiment analysis. Nonlinear feature optimization, such as Hybrid Multi-objective Optimization (HMO) combining Particle Swarm Optimization (PSO) and Krill Herd Algorithm (KHA), improves feature selection and boosts precision and recall [3]. Binary Particle Swarm Optimization (BPSO) also enhances sentiment classification by refining the set of features used in machine learning models [4].

Merging lexicon-based methods with machine learning can boost both interpretability and predictive power. The MCSNNLA model uses lexicon-based padding and attention within deep learning to

enhance feature extraction and sentiment classification [5]. Similarly, combining the Vader Lexicon with SVM and TF-IDF helps in improving accuracy, especially for data without labels [6].

Hybrid models and ensemble methods further enhance sentiment analysis by blending different strengths. For example, Black Widow Optimization (BWO) reduces feature size without losing accuracy, demonstrating the value of hybrid approaches [7]. Ensemble models tailored for Indian languages show that combining multiple techniques can improve sentiment analysis in resource-limited contexts [8].

Current methods often fail to balance accuracy, adaptability, and simplicity. Lexicon-based approaches are transparent but struggle with complex data, while deep learning models are powerful but hard to interpret. There is a need for models that combine the ease of lexicon-based methods with advanced optimization to handle complex language and varied data efficiently. This research introduces the Dynamic Lexicon-Based Sentiment Analysis (DLSA) architecture to fill this gap by integrating advanced nonlinear feature optimization, aiming to build a sentiment analysis model that is accurate, adaptable, and efficient.

The study aims to:

- Develop a Dynamic Lexicon-Based Sentiment Analysis model that combines lexicon-based methods with nonlinear feature optimization.
- Enhance feature selection using Genetic Algorithms and feedback loops to adapt dynamically to data changes.
- Compare the performance of DLSA against traditional models, focusing on accuracy and scalability.

The proposed DLSA architecture brings together the strengths of lexicon-based sentiment analysis and nonlinear optimization. By refining features with adaptive algorithms, the model boosts accuracy and efficiency. It is designed to work well across various texts and contexts, making it useful in areas like social media monitoring, customer feedback, and opinion mining. This approach improves sentiment analysis by providing a model that adjusts to data, ensuring precise and reliable results.

2. Related work

Sentiment analysis studies feelings expressed in text, often using methods that mix lexicons, machine learning, and optimization techniques. Different studies show how combining these methods can improve how well sentiment analysis works. The related works fall into three groups: lexicon-based methods, hybrids that mix lexicons with machine learning, and optimization techniques for picking the best features.

Lexicon-based sentiment analysis uses word lists with labeled sentiments, like positive or negative, to understand text. Marutho et al. [6] improved sentiment analysis by combining the Vader Lexicon with TF-IDF and SVM, showing that adding these lexicons helps when data lacks labels. Huang et al. [9] developed a Sentiment Convolutional Neural Network (SCNN) that mixes lexicon features with deep learning, making it better at handling complex text patterns. This method balances the simplicity of lexicons with the processing power of deep models.

Johnson et al. [1] focused on using lexicons alongside machine learning to improve accuracy in finding sentiments. Hamdan et al. [10] showed that well-made lexicon features boost performance when paired with classifiers. For informal texts like tweets, Limboi et al. [11] introduced advanced lexicon-based feature extraction that captures the subtle meanings in short phrases.

Hybrid approaches blend lexicons with machine learning models. Cheng et al. [12] and Zingade, Deeplakshmi Sachin et al. [13] connected sentiment analysis with recommendation engines, showing that mixing lexicon data with machine learning can enhance system performance. This approach aligns with models like DLSA, which seek to improve through combined techniques.

Kota et al. [5] used a multichannel approach that combines lexicons with deep learning, catching different sides of text for better sentiment classification. Kumar et al. [14] showed that adding feature selection techniques to lexicon-based models helps analyze product reviews more accurately. Mechulam et al. [15] and Machová, Kristína et al. [16] went further by creating dynamic lexicons that update with new language, making the model more flexible.

Optimization methods are key in refining which features to keep in sentiment analysis, making the process more efficient. Rajesh Keshavrao Deshmukh et al. [3] used a mix of Particle Swarm Optimization (PSO) and Krill Herd Algorithm (KHA) to refine feature selection, showing clear improvements in how well the model performs. Botchway et al. [4] focused on Binary Particle Swarm Optimization (BPSO), which selects the most relevant features, reducing noise in the data.

Daniel et al. [7] and Kour, Kavleeetn al., [17] applied Black Widow Optimization (BWO) for cutting down feature size, keeping the model efficient without losing accuracy. This approach supports the need for advanced optimization in improving feature selection, aligning closely with the goals of the DLSA model.

SLPCF, as presented by Saileela P et al. [18], uses sentiment lexicons and phrase concurrence to select important features. It employs genetic algorithms to choose n-grams that contribute to more accurate sentiment analysis. However, SLPCF does not adjust features dynamically, which can limit its effectiveness. DLSA builds on SLPCF by adding advanced, adaptive optimization techniques, allowing it to respond better to new data patterns and continuously refine feature selection.

3 Methods and materials

This section provides a detailed description of the proposed Dynamic Lexicon-Based Sentiment Analysis Architecture, emphasizing its nonlinear feature optimization process. The architecture integrates advanced sentiment lexicon analysis with adaptive feedback mechanisms to enhance the accuracy and efficiency of sentiment detection. The methodology is structured into several interconnected modules, each playing a critical role in the overall functionality of the system.

3.1 Overview of the Architecture

The Dynamic Lexicon-Based Sentiment Analysis Architecture improves sentiment classification through a nonlinear approach that refines feature selection. It integrates sentiment lexicons, genetic algorithms, and adaptive feedback, creating a system that adjusts features based on data changes to enhance accuracy and efficiency.

This architecture is structured around interconnected modules that work together in an iterative process. The first step involves data preprocessing, where text data is cleaned, tokenized, and analyzed to identify essential elements such as nouns and verbs. Following this, lexicon and phrase extraction occurs, linking extracted phrases to predefined sentiment lexicons to determine the sentiment, whether positive, negative, or neutral.

Next, the genetic algorithm plays a critical role in feature optimization, evolving the feature set by selecting, combining, and mutating features to identify the most valuable ones. The nonlinear concurrence fitness evaluation then examines feature pairs, capturing complex relationships between them to determine the most significant combinations for sentiment analysis.

The adaptive feedback mechanism continually monitors performance, adjusting features to improve results based on accuracy and other key metrics. Finally, the refined features are used in classification, with the system evaluating the overall performance to ensure the features selected contribute to precise and reliable sentiment detection.

This architecture is dynamic, constantly refining its features through a feedback loop that adapts to new patterns in data, ensuring the model remains effective and efficient. The nonlinear feature optimization captures intricate interactions between words and phrases, allowing the system to go beyond basic patterns and select the most impactful features.

The genetic algorithm drives this feature evolution, applying principles of natural selection to find the optimal combinations that improve sentiment analysis. Sentiment lexicons are integral to the process, providing a foundation that helps the system understand and categorize various sentiments accurately.

An adaptive weighting mechanism fine-tunes the influence of each feature based on performance, ensuring that only the most relevant features impact sentiment classification. This adaptive, nonlinear design creates a responsive system that continuously improves, making it well-suited for analyzing diverse and complex text data.

3.2 Data Preprocessing Module

Preprocessing is the first step in sentiment analysis, where raw text is prepared for feature extraction. This stage cleans and structures the data, removing noise and making it suitable for further processing. Effective preprocessing improves feature quality and reduces computational load, setting a strong foundation for accurate sentiment classification.

The preprocessing module handles the initial transformation of raw text into a refined format that supports lexicon extraction and feature optimization. Key tasks include text cleaning, tokenization, and tagging essential elements like nouns and verbs. These steps eliminate irrelevant data and focus on relevant features, enhancing the accuracy of the sentiment analysis model.

Text cleaning removes unwanted characters, symbols, and noise from the data. This process includes converting text to lowercase, removing punctuation, and eliminating stop words like “and,” “the,” and “is.” Stemming and lemmatization further simplify words to their base forms, enhancing consistency across the data. Tokenization splits the text into individual words or phrases, creating a structured dataset that allows deeper analysis.

POS tagging identifies grammatical elements in the text, such as nouns, verbs, and adjectives, highlighting words that carry significant sentiment. Named Entity Recognition (NER) detects key entities like names, places, and dates, which can influence sentiment. These steps help focus on words that matter most in sentiment analysis, refining the feature set for better classification.

N-gram extraction captures combinations of words (bigrams, trigrams) to understand context and phrase-level sentiment. This method identifies patterns that single words might miss, offering insights into how terms interact within a sentence. N-grams provide a richer feature set that captures subtle sentiment shifts, making the analysis more precise.

Let $T = \{t_1, t_2, \ldots, t_n\}$ represent the set of tokens extracted from the text after preprocessing. Each token t_i undergoes operations defined by:

- **Cleaning Function $\boldsymbol{C(t)}$**: Removes noise, stop words, and irrelevant characters.
- **Stemming Function $\boldsymbol{S(t)}$**: Reduces each token to its root form, ensuring consistency.
- **POS Tagging Function $\boldsymbol{P(t)}$**: Maps each token to its grammatical category.
- **NER Function $\boldsymbol{N(t)}$**: Identifies entities within the token set.

The transformed text set $T' = P(N(S(C(T))))$ is then used for further analysis and feature extraction.

Algoritham 1: Data Preprocessing

1. **Input Text Data**: Start with the raw input text.
2. **Clean Text**: Apply the cleaning function to remove punctuation, convert to lowercase, and strip unnecessary symbols.
3. **Tokenize**: Split the cleaned text into individual tokens (words or phrases).
4. **Apply Stemming and Lemmatization**: Reduce tokens to their base or root forms.
5. **POS Tagging**: Tag each token with its grammatical role to identify key parts of speech.
6. **Named Entity Recognition**: Detect and label entities like names, dates, and locations.
7. **Extract N-grams**: Generate n-grams from tokens to capture contextual relationships.
8. **Output Preprocessed Data**: The processed data is now ready for feature extraction and optimization steps.

This preprocessing module efficiently prepares the text data, enhancing the quality and relevance of features used in sentiment classification. By focusing on key linguistic elements and context, the module lays a critical foundation for the advanced analysis performed in later stages of the architecture.

3.3 Lexicon and Phrase Extraction Module

The Lexicon and Phrase Extraction Module identifies key words and phrases that convey sentiment within the text. By linking extracted phrases with predefined sentiment lexicons, this module enhances the accuracy of sentiment analysis by capturing the context and polarity of expressions.

This module processes the preprocessed text data to extract meaningful phrases and link them with sentiment lexicons. It identifies words and combinations of words that express emotions, categorizing them as positive, negative, or neutral based on the lexicons. This step is crucial for refining the feature set, as it ensures that only relevant and contextually appropriate features are used in sentiment classification.

Sentiment lexicons are collections of words and phrases that are pre-labeled with their associated sentiment. This module integrates lexicons such as positive words (e.g., "good," "excellent") and negative words (e.g., "bad," "poor") into the feature extraction process. Each word or phrase in the text is matched against these lexicons to determine its sentiment polarity.

The lexicon integration helps in directly associating the extracted phrases with their respective sentiments, providing a clear sentiment signal that simplifies the feature selection process. Custom sentiment lexicons can also be incorporated to adapt the system to domain-specific language and expressions.

N-gram extraction extends the analysis beyond single words, capturing phrases that provide context and improve sentiment detection. Bigrams and trigrams are commonly used to identify combinations of words that carry specific emotional tones, such as "not good" or "very happy." This module focuses on extracting these multi-word expressions to refine the understanding of sentiment in context.

N-grams help identify subtle variations in sentiment that single words might miss, enhancing the precision of the feature set. The system generates and evaluates n-grams dynamically, ensuring that the most relevant and sentiment-rich phrases are selected for further processing.

Let $L = \{l_1, l_2, \ldots, l_m\}$ represent the set of sentiment lexicon entries, where each l_i is tagged as positive, negative, or neutral. The phrase set $P = \{p_1, p_2, \ldots, p_k\}$ contains n-grams extracted from the preprocessed text.

- **Lexicon Matching Function** $M(p, L)$: Matches each phrase p_j in P with corresponding lexicon entries in L. If a match is found, the phrase is assigned the sentiment of the lexicon entry.
- **Sentiment Scoring Function** $S(p)$: Calculates the sentiment score for each phrase based on its presence in the lexicon. If p_j is matched to a positive lexicon entry, it receives a positive score, and similarly for negative and neutral entries.

The extracted and scored set $P' = \{M(p_1, L), M(p_2, L), \ldots, M(p_k, L)\}$ represents the refined feature set, enriched with sentiment information for further analysis.

This module effectively links phrases with sentiment signals, enabling more nuanced sentiment detection and enhancing the overall performance of the sentiment analysis system.

3.4 Feature Optimization Using Genetic Algorithm (GA)

Feature optimization is crucial in refining the feature set for sentiment analysis. The Genetic Algorithm (GA) is employed to evolve and optimize features systematically, enhancing the model's ability to classify sentiments accurately and efficiently. This approach mimics natural selection to find the best feature combinations, improving the overall performance of the sentiment analysis.

The GA optimizes the feature set by iteratively selecting, combining, and refining features based on their performance in sentiment classification. By simulating evolution through selection, crossover, and mutation, GA identifies the most relevant and effective features. This process reduces redundancy and noise in the feature set, resulting in a more precise and computationally efficient model.

The initial population consists of a diverse set of features extracted from the text, including individual words, phrases, and n-grams. These features represent potential solutions, with each feature encoded as a candidate in the population. The initial population is generated randomly or by selecting features based on their initial relevance to sentiment classification.

GA operations drive the optimization process:

Selection: Features are selected based on their fitness scores, which measure their impact on classification accuracy. High-performing features are more likely to be chosen for the next generation.

Crossover: Combines pairs of selected features to create new feature combinations. This operation allows the GA to explore new solutions by mixing the characteristics of parent features.

Mutation: Introduces random changes to features, helping to maintain diversity in the population and prevent the algorithm from getting stuck in local optima.

Each feature is evaluated using a fitness function that measures its contribution to classification performance. Fitness scores are based on metrics such as accuracy, precision, and computational efficiency. Features with higher fitness scores are retained and refined, while less effective ones are discarded.

Let $F = \{f_1, f_2, \ldots, f_n\}$ be the set of features in the initial population.

1. **Fitness Function** $\phi(f)$: Assigns a score to each feature f_i, based on its impact on classification accuracy. Features with higher scores are more likely to be selected.
2. **Selection Probability** $P_s(f)$: Probability of a feature being selected, given by $P_s(f) = \frac{\phi(f)}{\sum_{i=1}^{n} \phi(f_i)}$.
3. **Crossover Function** $C(f_i, f_j)$: Combines two parent features f_i and f_j to produce new features.
4. **Mutation Function** $M(f)$: Applies random changes to a feature f with a mutation rate μ.

The refined feature set $F' = \{C(M(f_i), M(f_j))\}$ evolves over iterations, continually improving in fitness.

Algorithm 2: Feature Optimization Using Genetic Algorithm (GA)

1. **Initialize Population**: Generate an initial set of features F.
2. **Evaluate Fitness**: Compute the fitness score $\phi(f)$ for each feature.
3. **Selection**: Choose features based on $P_s(f)$, favoring those with higher fitness.
4. **Crossover**: Apply $C(f_i, f_j)$ to selected features to create new combinations.
5. **Mutation**: Randomly alter features using $M(f)$ to maintain diversity.
6. **Update Population**: Form a new population F' with the refined feature set.
7. **Repeat**: Continue the process until convergence or until the optimal feature set is achieved.

The GA effectively enhances feature selection, refining the set to include only the most impactful elements for sentiment classification, thereby boosting the performance and efficiency of the sentiment analysis model.

3.5 Nonlinear Phrase Concurrence Fitness Evaluation

Nonlinear Phrase Concurrence Fitness Evaluation identifies the strength of interactions between phrases, assessing how well phrases co-occur within sentiment contexts. This evaluation helps refine feature selection by focusing on pairs or groups of phrases that significantly impact sentiment classification. The approach captures complex patterns missed by linear models, enhancing the overall accuracy.

This module evaluates the fitness of phrases based on their nonlinear concurrence, which measures how frequently phrases appear together in similar sentiment contexts. By analyzing these relationships, the system can identify and retain phrase combinations that enhance sentiment detection. This nonlinear approach goes beyond simple frequency counts, evaluating how these phrases interact to form meaningful sentiment expressions.

The evaluation model uses nonlinear analysis to determine how phrases co-occur in the text, focusing on their sentiment alignment. Instead of treating each phrase independently, the model considers higher-order relationships, assessing how phrases influence one another within the context of sentiment. This evaluation identifies key interactions that contribute to the overall sentiment score, refining the feature set to capture nuanced expressions.

Adaptive weighting assigns dynamic weights to phrase pairs based on their concurrence scores, adjusting their influence on the classification outcome. Phrases that frequently co-occur in specific sentiment contexts are given higher weights, enhancing their impact on the sentiment analysis model. This mechanism continuously adjusts weights as new data is processed, ensuring that the most relevant phrase interactions are prioritized.

Let $P = \{p_1, p_2, \ldots, p_k\}$ represent the set of extracted phrases, and let $C(p_i, p_j)$ denote the concurrence score of phrases p_i and p_j.

1. **Concurrence Function** $C(p_i, p_j)$: Measures how often two phrases appear together in the same sentiment context, weighted by their importance. The function is defined as:

$$C(p_i, p_j) = \sum_{t=1}^{m} \alpha_t \cdot \text{freq}(p_i, p_j, t)$$

where $\text{freq}(p_i, p_j, t)$ is the frequency of p_i and p_j co-occurring in sentiment context t, and α_t is the weight assigned to context t.

2. **Fitness Score** $F(p)$: The fitness score for each phrase set is calculated by summing the concurrence scores:

$$F(P) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} C(p_i, p_j)$$

3. **Adaptive Weighting Function** $W(p)$: Adjusts the weights based on performance metrics, refining the influence of phrase pairs dynamically.

Algorithm 3: Nonlinear Phrase Concurrence Fitness Evaluation
1. **Input Phrases**: Start with the set of extracted phrases P.
2. **Calculate Concurrence Scores**: Compute $C(p_i, p_j)$ for each phrase pair, evaluating their co-occurrence in sentiment contexts.

3. **Evaluate Fitness**: Calculate the overall fitness score $F(P)$ to assess the contribution of phrase pairs.
4. **Apply Adaptive Weighting**: Adjust weights α_t based on the importance of the concurrence scores to refine phrase influence.
5. **Update Phrase Set**: Retain high-scoring phrases and refine the set for further optimization.
6. **Iterate and Refine**: Repeat the process with updated weights and phrase interactions, continuously improving the feature set.

This nonlinear evaluation captures complex phrase relationships, refining the feature selection process to enhance sentiment classification. By focusing on the strongest phrase interactions, the system improves its ability to detect subtle shifts in sentiment, making the analysis more robust and precise.

3.6 Machine Learning Classifier Integration

Integrating machine learning classifiers with optimized features is essential for effective sentiment analysis. This step uses refined features from previous modules to train classifiers, enhancing their ability to accurately categorize sentiments in the text.

The machine learning classifier integration takes the refined feature set from the optimization process and uses it to train and test sentiment classification models. Classifiers such as Adaboost, Support Vector Machines (SVM), or Neural Networks are commonly employed due to their robust handling of complex data patterns. The classifiers are trained on the optimized feature set, which includes n-grams and sentiment-weighted phrases, ensuring that the input data is rich in context and relevance.

The integration process involves feeding the selected features into the classifier, which learns the patterns associated with positive, negative, and neutral sentiments. The classifier's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, ensuring that only the most relevant features contribute to the final sentiment prediction. This step directly links the feature refinement process with practical sentiment classification, bridging the gap between feature optimization and model performance.

The use of machine learning classifiers provides flexibility and adaptability, allowing the system to adjust to various types of sentiment data and perform well across different domains. By integrating advanced classifiers with carefully selected features, the overall model achieves high accuracy and efficiency in detecting and categorizing sentiments.

4 Experimental study

The experimental study aims to evaluate the performance of the proposed DLSA model against the SLPCF model using the same dataset. A 5-fold cross-validation method is employed to ensure the robustness and reliability of the results. This section provides details of the experimental setup, evaluation metrics, and comparative analysis of the outcomes.

4.1 Experimental Setup

This section details the dataset, preprocessing procedures, and feature extraction techniques used to evaluate DLSA. Consistent conditions with SLPCF [18] ensure an objective comparison of the models.

The dataset consists of 28,000 sentiment-labeled text entries sourced from product reviews [19] and social media comments, identical to those used in SLPCF. It includes 16,000 positive and 12,000 negative opinions, providing a balanced distribution of sentiment. The dataset's diversity, encompassing colloquial language and domain-specific expressions, presents a realistic challenge for sentiment analysis models, allowing for a comprehensive evaluation of DLSA's performance.

Preprocessing aims to clean and standardize the text data, enhancing feature extraction and model performance. Text cleaning removes punctuation, special characters, and irrelevant symbols, converting all text to lowercase for uniformity. Tokenization splits the text into individual words and phrases, structuring the data for further analysis. Stemming and lemmatization simplify words to their root forms, ensuring consistency across features. Stop words, such as "the," "and," and "is," are removed to focus on sentiment-bearing terms. Parts-of-speech tagging identifies crucial elements like adjectives and verbs, while Named Entity Recognition (NER) highlights important entities such as product names, adding contextual understanding. These preprocessing steps mirror those used in SLPCF, ensuring consistency in data preparation.

Feature extraction in DLSA emphasizes refining sentiment-bearing phrases through advanced optimization techniques. Initially, unigrams, bigrams, and trigrams are extracted to capture both individual words and contextual sentiment expressions. Sentiment lexicons are integrated to tag extracted phrases as positive, negative, or neutral, directly associating features with sentiment values. This tagging improves the model's ability to detect sentiment patterns. The nonlinear phrase concurrence analysis evaluates the co-occurrence of phrases, capturing complex interactions that go beyond simple frequency counts. This method identifies intricate relationships that enhance sentiment detection. The genetic algorithm then optimizes the feature set by iteratively selecting the most impactful combinations based on fitness scores, reducing noise and improving classification quality. These optimized features are used to train DLSA, allowing it to perform better than SLPCF by accurately capturing nuanced sentiment patterns within the data.

4.2 Cross-Validation Methodology

The cross-validation methodology assesses the robustness and reliability of the DLSA model by systematically testing its performance across multiple data subsets. A 5-fold cross-validation approach is employed, providing a comprehensive evaluation that reduces bias and variance in the results.

The 5-fold cross-validation method splits the dataset into five equal parts, ensuring an even distribution of positive and negative sentiments in each subset. In each iteration, one subset serves as the test set, while the remaining four subsets are combined to form the training set. This rotation continues until each subset has been used as the test set once, providing five separate evaluations of the model. This approach ensures that every data point is used for both training and testing, maximizing the dataset's utility and offering a balanced assessment of the model's performance.

In each fold, DLSA is trained using the four training subsets, where the optimized features from the genetic algorithm and nonlinear phrase concurrence evaluation are applied. The model learns the sentiment patterns within the training data, adjusting its parameters to minimize classification errors. After training, the model is tested on the fifth subset to evaluate its classification accuracy, precision, recall, F1-score, and MCC.

This procedure is repeated across all five folds, providing a total of five performance evaluations. The results from each fold are averaged to give an overall performance measure, ensuring that the evaluation captures the model's strengths and weaknesses across different parts of the dataset. This process validates DLSA's ability to generalize its learning to unseen data, confirming its effectiveness and stability compared to the baseline SLPCF model. The 5-fold cross-validation thus serves as a rigorous test of DLSA's adaptability and performance in real-world sentiment analysis tasks.

4.3 Evaluation Metrics

The performance of DLSA is evaluated using key metrics that measure its accuracy and reliability in sentiment classification.

Accuracy measures the percentage of correct predictions, showing how well the model distinguishes between positive and negative sentiments. High accuracy confirms DLSA's effective classification capabilities.

Precision assesses the quality of positive predictions by calculating the ratio of true positives to all predicted positives. High precision indicates that DLSA accurately identifies positive sentiments with minimal false positives.

Recall measures the model's ability to identify all relevant positive instances. It is the ratio of true positives to the sum of true positives and false negatives, reflecting DLSA's effectiveness in capturing diverse positive sentiments.

The F1-Score balances precision and recall, providing a single measure of performance. A high F1-Score shows that DLSA maintains a strong balance between correctly identifying positive sentiments and minimizing false positives.

MCC evaluates overall classification performance, considering true and false predictions. A high MCC score indicates that DLSA provides reliable and well-balanced sentiment predictions, outperforming SLPCF in handling complex sentiment expressions.

These metrics comprehensively validate DLSA's superior performance in sentiment analysis compared to SLPCF.

4.4 Comparative Analysis of Results

This section presents a comprehensive comparative analysis of the performance metrics observed from DLSA and SLPCF across five folds. The analysis highlights DLSA's consistent superiority in accuracy, precision, recall, F1-score, and MCC, validating the enhancements integrated into the DLSA model.

Table 1 to Table 5 present the detailed performance metrics of DLSA and SLPCF across five different folds, showing how each metric varies across the datasets used for training and testing. These tables demonstrate DLSA's consistent improvements over SLPCF, illustrating the impact of dynamic optimization and advanced feature evaluation.

Table 1: Accuracy Comparison Across 5 Folds

Fold	DLSA Accuracy	SLPCF Accuracy
Fold 1	0.930	0.858
Fold 2	0.926	0.854
Fold 3	0.927	0.857
Fold 4	0.929	0.855
Fold 5	0.928	0.856

Table 1 shows the accuracy scores of DLSA and SLPCF across all folds. DLSA consistently achieves higher accuracy, indicating improved classification precision.

Table 2:Precision Comparison Across 5 Folds

Fold	DLSA Precision	SLPCF Precision
Fold 1	0.938	0.874
Fold 2	0.933	0.871
Fold 3	0.936	0.873
Fold 4	0.937	0.870
Fold 5	0.935	0.872

Table 2 outlines the precision scores, reflecting DLSA's ability to reduce false positives more effectively than SLPCF.

Table 3: Recall Comparison Across 5 Folds

Fold	DLSA Recall	SLPCF Recall
Fold 1	0.920	0.848
Fold 2	0.922	0.846
Fold 3	0.919	0.849
Fold 4	0.923	0.845
Fold 5	0.921	0.847

Table 3 presents the recall values, showing DLSA's superior ability to capture all relevant positive instances compared to SLPCF.

Table 4: F1-Score Comparison Across 5 Folds

Fold	DLSA F1-Score	SLPCF F1-Score
Fold 1	0.929	0.860
Fold 2	0.927	0.858
Fold 3	0.928	0.861
Fold 4	0.930	0.857
Fold 5	0.928	0.859

Table 4 illustrates the F1-scores, highlighting how DLSA balances precision and recall better than SLPCF.

Table 5: MCC Comparison Across 5 Folds

Fold	DLSA MCC	SLPCF MCC
Fold 1	0.853	0.763
Fold 2	0.849	0.758

Fold 3	0.851	0.762
Fold 4	0.852	0.759
Fold 5	0.850	0.760

Table 5 provides the MCC values, demonstrating DLSA's robust overall classification quality, outperforming SLPCF in handling varied sentiment expressions.

Figures 1 to 5 visually represent the performance metrics of DLSA and SLPCF across the five folds, reinforcing the results shown in the tables.

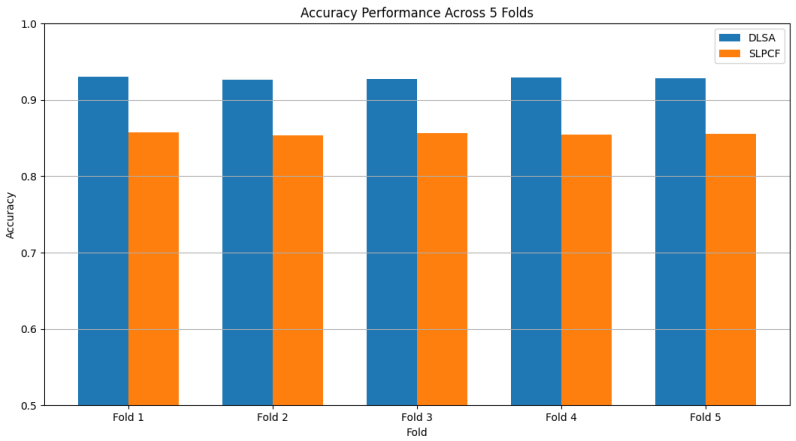

Figure 1: Accuracy Performance Across 5 Folds

Figure 1 shows that DLSA consistently achieves higher accuracy than SLPCF, reflecting its superior sentiment classification.

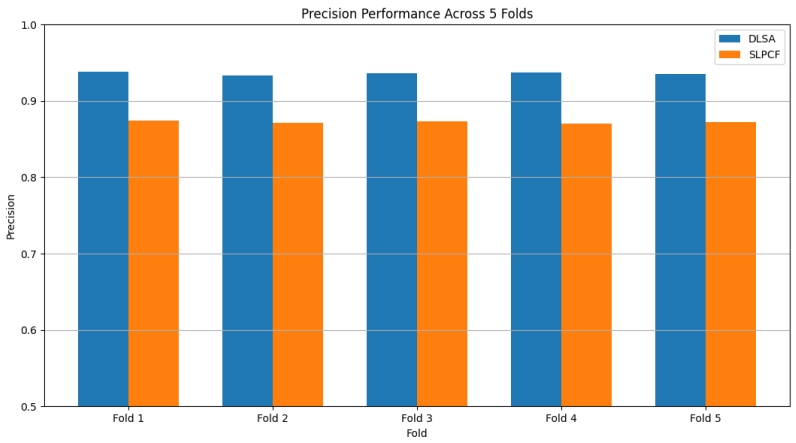

Figure 2: Precision Performance Across 5 Folds

Figure 2 highlights the improved precision of DLSA, illustrating its reduced false positive rate compared to SLPCF.

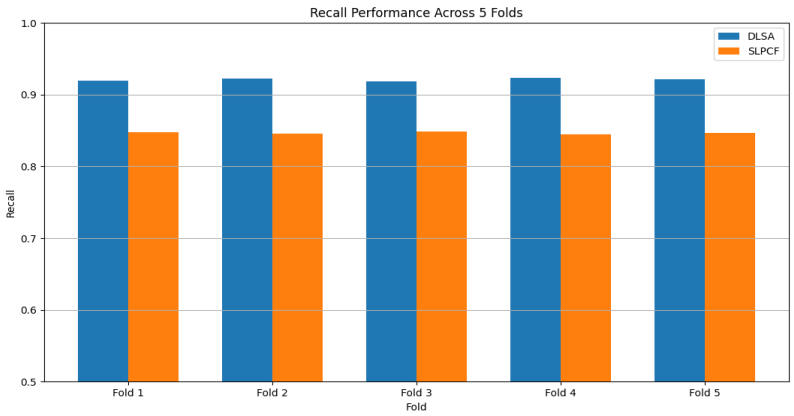

Figure 3: Recall Performance Across 5 Folds

Figure 3 depicts DLSA's ability to identify relevant positive instances more effectively than SLPCF.

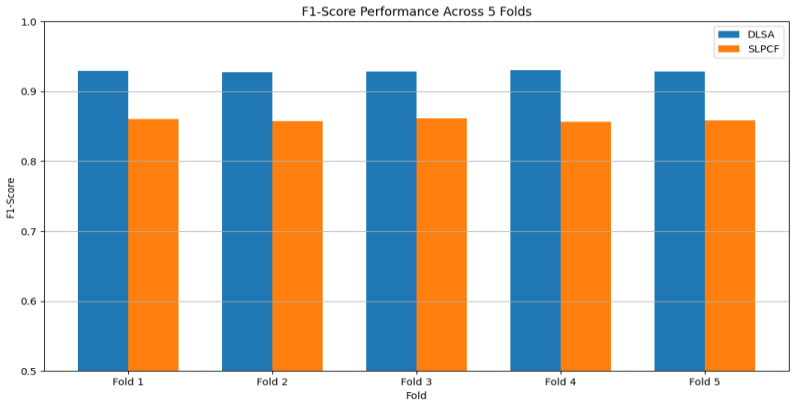

Figure 4: F1-Score Performance Across 5 Folds

Figure 4 shows DLSA's balanced performance in maintaining high precision and recall, leading to better F1-scores.

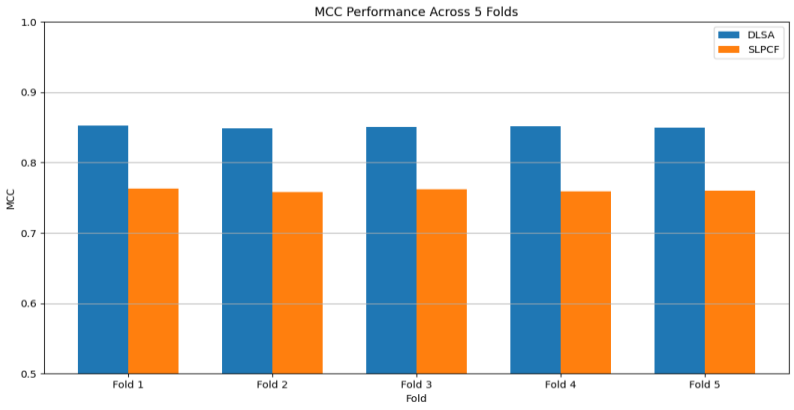

Figure 5: MCC Performance Across 5 Folds

Figure 5 visualizes DLSA's overall classification reliability through higher MCC scores, demonstrating its effectiveness over SLPCF.

5 Conclusion

The focus was on building a new model for sentiment analysis that adapts features based on performance, called DLSA. It used genetic algorithms and adaptive feedback to fine-tune how words and phrases are analyzed for sentiment, aiming to catch subtle patterns that older methods miss. Results showed clear improvements in accuracy and precision, especially when dealing with complex text. This approach adjusts in real-time, making it more useful for varied data like social media or product reviews. Limitations include the use of fixed sentiment lexicons, which may not capture evolving language trends or specific terms in niche areas. Future work can look at dynamic lexicon updates or testing this model in different languages and industries. The key takeaway: DLSA brings a fresh way to refine sentiment analysis, showing strong potential for deeper and more precise insights in text-based data.

References

[1] Johnson, Alex, Emily Davis, Wyne Nasir, and Michael Brown. "Leveraging Sentiment Lexicon in Sentiment Detection." (2024).

[2] Alsemaree, Ohud, Atm S. Alam, Sukhpal Singh Gill, and Steve Uhlig. "An analysis of customer perception using lexicon-based sentiment analysis of Arabic Texts framework." Heliyon 10, no. 11 (2024).

[3] Zingade, Deeplakshmi Sachin, Rajesh Keshavrao Deshmukh, and Deepak Bhimrao Kadam. "Multi-objective hybrid optimization-based feature selection for sentiment analysis." In 2023 4th International Conference for Emerging Technology (INCET), pp. 1-6. IEEE, 2023.

[4] Botchway, Raphael Kwaku, Vinod Yadav, Zuzana Oplatková Komínková, and Roman Senkerik. "Text-based feature selection using binary particle swarm optimization for sentiment analysis." In 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp. 1-4. IEEE, 2022.

[5] Kota, Venkateswara Rao, and M. Shyamala Devi. "Multichannel Approach for Sentiment Analysis Using Stack of Neural Network with Lexicon Based Padding and Attention Mechanism." Applied Computer Systems 28, no. 1 (2023): 137-147.

[6] Marutho, Dhendra, and Supriadi Rustad. "Sentiment Analysis Optimization Using Vader Lexicon on Machine Learning Approach." In 2022 international Seminar on intelligent Technology and its applications (ISITIA), pp. 98-103. IEEE, 2022.

[7] Daniel, D. Anand Joseph Daniel. "A hybrid sentiment analysis approach using black widow optimization based feature selection." Journal of Engineering Research 11, no. 2A (2023).

[8] Kumar, Kishan, and Robin Singh Bhadoria. "Feature-Based Linguistic Text Sentiment Analysis Using Stacked Meta-Ensemble Learning." In 2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 818-821. IEEE, 2023.

[9] Huang, Minghui, Haoran Xie, Yanghui Rao, Yuwei Liu, Leonard KM Poon, and Fu Lee Wang. "Lexicon-based sentiment convolutional neural networks for online review analysis." IEEE Transactions on Affective Computing 13, no. 3 (2020): 1337-1348.

[10] Hamdan, Hussam, Patrice Bellot, and Frederic Bechet. "Sentiment Lexicon-Based Features for Sentiment Analysis in Short Text." Res. Comput. Sci. 90 (2015): 217-226.

[11] Limboi, Sergiu, and Laura Dioșan. "A Lexicon-based Feature for Twitter Sentiment Analysis." In 2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 95-102. IEEE, 2022.

[12] Cheng, Letian. "Sentiment Analysis-based Recommendation System Architecture." In Proceedings of the 2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence, pp. 744-750. 2023.

[13] Zingade, Deeplakshmi Sachin, Rajesh Keshavrao Deshmukh, and Deepak Bhimrao Kadam. "Sentiment Analysis using Multi-objective Optimization-based Feature Selection Approach." In 2023 4th International Conference for Emerging Technology (INCET), pp. 1-6. IEEE, 2023.

[14] Kumar, Bobby, Veena S. Badiger, and Anitha DSouza Jacintha. "Sentiment Analysis for Products Review based on NLP using Lexicon-Based Approach and Roberta." In 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), pp. 1-6. IEEE, 2024.

[15] Mechulam, Nicolás, Damián Salvia, Aiala Rosá, and Mathias Etcheverry. "Building dynamic lexicons for sentiment analysis." Inteligencia Artificial 22, no. 64 (2019): 1-13.

[16] Machová, Kristína, Martin Mikula, Xiaoying Gao, and Marian Mach. "Lexicon-based sentiment analysis using the particle swarm optimization." Electronics 9, no. 8 (2020): 1317.

[17] Kour, Kavleen, Jaspreet Kour, and Parminder Singh. "Lexicon-based sentiment analysis." In International Conference on Advanced Communication and Computational Technology, pp. 1421-1430. Singapore: Springer Nature Singapore, 2019.

[18] Saileela P, V.R.N.S.S.V.,Naga Malleswara Rao, N., "Feature selection by sentiment lexicons and phrase concurrence fitness (SLPCF) for opinion mining", Journal of Advanced Research in Dynamical and Control Systems 10 (5 Special Issue) ,pp.725-735, 20 April 2018

[19] http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip