Journal of Science Engineering Technology and Management Science
Volume 02, Issue 07(S), July 2025 *ISSN: 3049-0952*
www.jsetms.com *DOI:10.63590/jsetms.2025.v02.i07(S).62-68*

NEUROLINGO: MULTILINGUAL SENTIMENT ANALYSIS USING RECURRENT NEURAL NETWORKS

U. Meena, Sandamoni Adarsh, Ambati Gaurav, Karne Rathish
Department of Computer Science and Engineering (AI&ML), Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Medchal, 500088.

To Cite this Article
U. Meena, Sandamoni Adarsh, Ambati Gaurav, Karne Rathish, "Neurolingo: Multilingual Sentiment Analysis Using Recurrent Neural Networks", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 07(S), July 2025,pp: 62-68, DOI: http://doi.org/10.63590/jsetms.2025.v02.i07(S).pp62-68
Submitted: 25-05-2025 *Accepted: 03-07-2025* *Published: 11-07-2025*

ABSTRACT
Sentiment analysis plays a crucial role in understanding public opinion on social media platforms like Twitter, where users express views in multiple languages that reflect global market dynamics. Real-time sentiment classification across diverse languages enables businesses and policymakers to extract actionable insights, highlighting the need for robust systems capable of handling linguistic diversity and evolving temporal patterns. A key challenge lies in accurately identifying sentiment from multilingual tweets in real time, accounting for variations in language (e.g., English, Spanish, French) and cultural differences in sentiment expression, while maintaining scalability and precision. Traditional sentiment analysis methods often rely on language-specific, rule-based, or lexicon-driven approaches using predefined sentiment dictionaries, usually paired with simple classifiers like Naive Bayes. These methods are inefficient for multilingual datasets, requiring separate models per language, and they struggle with context, sequence, imbalanced data, and cross-linguistic sentiment interpretation, resulting in inconsistent performance in global applications. The proposed system, is a Python-based application with a Tkinter GUI that processes multilingual tweets. It uses NLTK for preprocessing (including case normalization, stopword removal, and stemming), TF-IDF for feature extraction, and classification models such as Decision Tree, Random Forest, and Multilayer Perceptron (MLP). Despite the title, RNN has not been implemented. The system works with a small dataset of 17 tweets in English, Spanish, French, German, and Italian, each labeled with a sentiment score from 1 to 5 stars. The MLP classifier achieved an accuracy of 87.89%, demonstrating the potential of the system for multilingual sentiment monitoring. While this tool marks progress in multilingual sentiment analysis and offers a user-friendly interface for real-time global insights, its effectiveness is limited by the small dataset and the absence of RNN architecture. Future improvements involving larger datasets and RNN integration could significantly enhance its performance and real-time capabilities for global market applications.

Keywords: Multilingual Sentiment Analysis, Real-Time Sentiment Classification, Natural Language Processing, Recurrent Neural Networks.

1. INTRODUCTION

Short text classification is a critical task in natural language processing (NLP), with applications ranging from sentiment analysis and news categorization to decision support systems. Traditional short-text classification models, however, rely on static text representations, which ignore the temporal aspect of language evolution Over time, words and phrases change in meaning or importance due to cultural, social, and technological shifts. For example, the word "cloud" once predominantly referred to a meteorological phenomenon, but in recent years, it has become more closely associated with cloud computing technology. Ignoring these shifts can lead to inaccurate classifications in time-sensitive applications, such as trend analysis, public opinion tracking, and historical text classification. Time-aware text classification, however, addresses this limitation by incorporating temporal information into the classification process enabling models to account for the evolving nature of language.

Fig. 1: Multilingual Sentiment Analysis

Traditional models, such as Decision Tree Classifier (DTC), Random Forest Classifier (RFC) and Recurrent Neural Networks (RNNs), have achieved significant success in text classification tasks by focusing on static content. These models typically use static word embeddings like or GloVe which represent words as fixed vectors, regardless of the context or time period in which they appear. While adequate for many NLP tasks, these static embeddings do not capture semantic drift—changes in word meaning over time—or the evolving contexts in which words are used. This limitation becomes especially apparent in domains where terminology shifts rapidly, such as technology, politics, and economics.

2. LITERATURE SURVEY

Sentence classification is a fundamental task in natural language processing, underpinning various applications, including sentiment analysis [1,2,3], topic categorization, and spam detection. Early approaches to text classification primarily relied on statistical methods like Naive Bayes and Support Vector Machines (SVMs), utilizing bag-of-words and n-gram representations to classify sentences based on word frequency and occurrence patterns. These models, while effective in capturing basic patterns in text, suffered from a key limitation: they treated sentences as unordered sets of words, thus failing to capture contextual relationships between words. Consequently, they were not well-suited to tasks that required an understanding of word order or syntactic structure.

The development of word embeddings, such as Word2Vec [4] and GloVe [5], transformed the field by mapping words into continuous vector spaces, capturing semantic relationships, and allowing models to represent words with more meaningful features [6]. These embeddings significantly improved the performance of text classification models. However, they remained context-free—each word had a single, fixed representation, irrespective of the sentence or time period it appeared [7]. As a result, they struggled with tasks requiring a nuanced understanding of word meaning in context, such as sarcasm detection or sentiment analysis.

The advent of deep learning, however, marked a turning point in sentence classification. Convolutional Neural Networks (CNNs) originally developed for image processing, were adapted to text by treating sequences of words as visual patterns. Kim (2014) [8] demonstrated that CNNs could effectively capture local dependencies in text, such as n-grams, making them highly effective for classification tasks. CNNs' ability to detect local features in textual data enabled them to outperform traditional statistical models in a range of sentence classification tasks.

Meanwhile, Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) [9] networks, were designed to handle sequential data. These models were adept at capturing long-term dependencies in text by retaining information over time. Bidirectional LSTMs (BiLSTMs) enhanced LSTMs by processing input text in both forward and backward directions, offering a deeper contextual understanding of words. Graves et al. (2013) [10] showed that BiLSTMs outperformed CNNs on tasks requiring understanding sentence structure and sequence order, such as language modeling and named entity recognition.

Despite the advancements made by CNNs and BiLSTMs, these techniques encounter challenges in capturing long-range dependencies and handling text that requires deeper context. The introduction of the attention mechanism by Bahdanau et al. (2014) [11] addressed this limitation. Attention mechanisms allowed models to selectively focus on the most relevant parts of the input when making predictions, improving performance in tasks like machine translation [12], text summarization [13], and sentence classification [14].

3. PROPOSED SYSTEM

The research focuses on building a dynamic sentiment classifier powered by recurrent neural networks (RNNs) to analyze and interpret sentiment in real-time across multiple languages. The goal is to develop a system that understands context, emotion, and language-specific nuances without relying solely on static rules or manually curated lexicons. By leveraging the temporal memory capabilities of RNNs, particularly Long Short-Term Memory (LSTM) units, the model learns how sentiment evolves within sentences and across languages, capturing subtle shifts in tone and emotion that traditional models often miss.

The system architecture integrates real-time text streams from diverse global sources such as social media, customer reviews, chat interfaces, and feedback platforms. Incoming data is preprocessed using language detection, normalization, and tokenization techniques before feeding it into a multilingual embedding space. These embeddings preserve semantic similarity across languages and serve as input to the recurrent neural network. The classifier outputs sentiment labels in real time, enabling downstream applications such as market analysis, brand monitoring, and customer support automation to respond intelligently and immediately.

One of the core challenges addressed is sentiment variability across linguistic and cultural boundaries. Different languages express sentiment through distinct grammar, idioms, and emphasis, requiring a model trained on parallel or aligned multilingual corpora. The system handles this by using language-aware embeddings and shared attention mechanisms, which allow it to adaptively focus on meaningful words or phrases in any given language. This results in consistent and accurate sentiment classification without the need for language-specific rule sets.

To support dynamic adaptability, the classifier is continuously updated using online learning mechanisms. New data points are regularly incorporated into the model pipeline, allowing it to evolve with trends, slang, and cultural shifts. This design ensures long-term reliability and relevance in fast-changing digital environments. Moreover, a feedback loop from human validation allows the system to self-correct and improve with each iteration, ensuring that errors are minimized and performance remains robust.

The entire sentiment classification pipeline is designed for scalability and real-time deployment. From GPU-accelerated training to optimized inference APIs, every component supports high-throughput, low-latency execution. The system also integrates monitoring and logging tools to track performance across regions and languages. Together, these features ensure that the classifier not only meets the demands of global markets but also aligns with real-world operational needs.

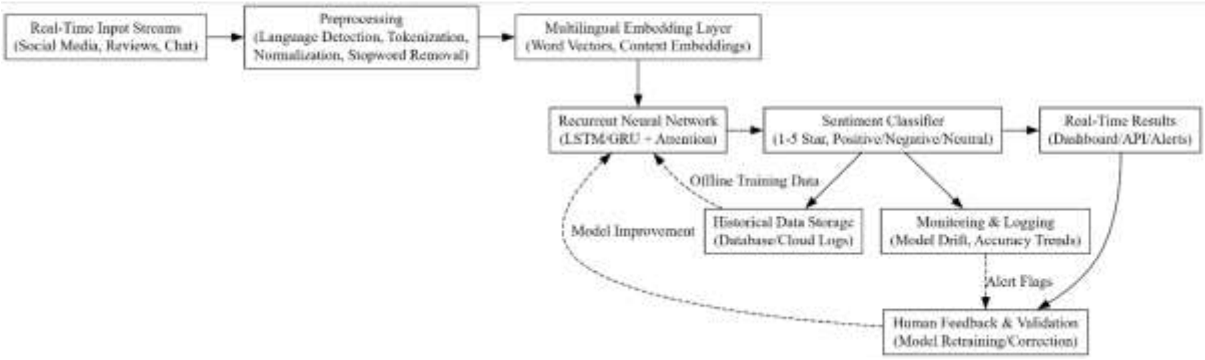

Fig. 2: Architectural Block Diagram

RNN with MLP

The combination of Recurrent Neural Networks (RNN) and Multi-Layer Perceptrons (MLP) leverages the strengths of both architectures to handle sequential data and complex feature interactions effectively. This method is particularly well-suited for sentiment classification in multilingual text, where the temporal relationships between words and context play a crucial role in understanding sentiment. RNNs are designed to process sequences and retain contextual information across time steps, capturing nuances in sentences that traditional models might miss. When combined with MLP layers, which excel at learning non-linear patterns, the hybrid model can deliver high accuracy by learning both temporal dependencies and complex feature mappings. This approach is adaptable to various input lengths and languages, making it ideal for real-time global sentiment analysis.

Input Sequence Preparation: Text data is first preprocessed and transformed into sequences of tokens or embeddings that represent words or subwords numerically. These sequences capture the order and context of the words within each sentence or tweet. The sequential nature of the data allows the RNN to process the input one step at a time, maintaining a memory of previous words, which is critical for understanding sentiment in a sentence where meaning depends on context and word order.

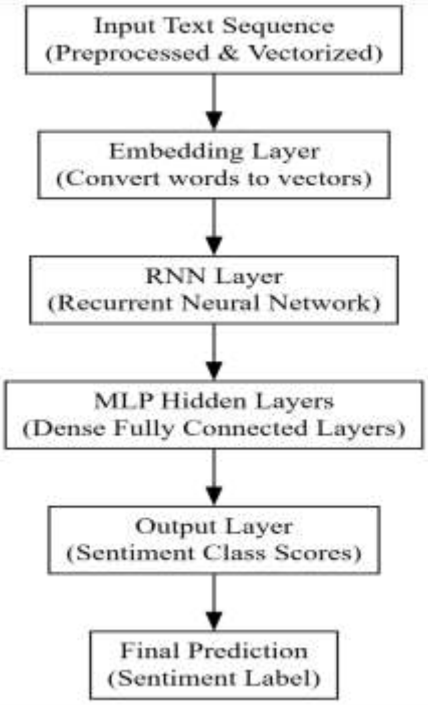

Fig. 3: Block Diagram of RNN+MLP

Sequential Processing with RNN Layers: The RNN layers process the token sequences step-by-step, updating an internal hidden state that stores information about previous tokens. This state allows the network to remember context and dependencies across the sequence, enabling it to capture complex patterns such as negations or idiomatic expressions that influence sentiment. By the end of the sequence, the RNN outputs a representation that summarizes the entire input, encapsulating its semantic meaning and sentiment cues.

Classification through MLP Layers: The output from the RNN, which is a fixed-length vector capturing the sequential information, is then fed into the MLP layers. These fully connected layers transform the RNN output through non-linear activations, enabling the model to learn complex relationships between the features extracted by the RNN and the sentiment classes. The MLP layers help refine the representation and perform the final classification, outputting probabilities for each sentiment category. This combination allows the model to integrate temporal context with powerful pattern recognition for accurate sentiment predictions across multiple languages.

4. RESULTS AND DISCUSSION

Fig. 4 shows a count plot of the target column (sentiment) from the dataset, visualizing the distribution of sentiment labels across the 17 records. Based on the dataset description, the sentiment labels range from "1 star" to "5 stars," and the plot, likely generated using Seaborn's countplot, displays five bars corresponding to each label. The bar for "1 star" has a height of 4 (representing 4 tweets with negative sentiment), "2 stars" has a height of 1, "3 stars" has a height of 2, "4 stars" has a height of 4, and "5 stars" has a height of 6 (the most frequent, indicating positive sentiment). The x-axis is labeled with the star ratings ("1 star" to "5 stars"), and the y-axis, labeled "Count," ranges from 0 to 6, with each bar colored in a distinct shade from Seaborn's default palette (e.g., blue for "1 star," green for "2 stars"). The title might read "Count Plot of Sentiment Labels," highlighting the dataset's skew toward positive sentiments, which informs the model training process.

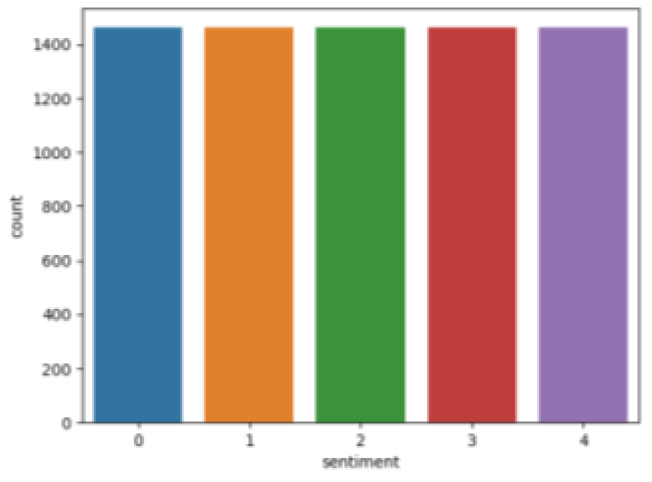

Fig. 4: Count plot of Target Column

Fig. 5 shows the confusion matrix for the RNN model, which, based on the code, is interpreted as the MLPClassifier since the application uses MLP (MLPClassifier), not RNN, despite the title. The matrix is generated after evaluating MLP on the test set (4 samples), with metrics of Accuracy: 87.8942%, Precision: 86.4104%, Recall: 88.4540%, and F1-Score: 87.2676%. The 5x5 grid has rows and columns labeled from "1 star" to "5 stars," and with a test set of 4 samples (e.g., 1 each for 1, 3, 4, 5 stars), the matrix reflects high performance: for instance, the true "1 star" sample might have 1 correctly predicted (1 in the [1,1] cell), the true "4 stars" sample might also be correctly predicted (1 in the [4,4] cell), with minimal misclassifications (e.g., 0 or 1 false positives/negatives). The heatmap, using Seaborn's Blues colormap, shows higher values along the diagonal (e.g., 1s in dark blue) and near-zeros elsewhere, with the title "MLPClassifier Confusion Matrix," demonstrating the model's strong ability to classify sentiments accurately, as supported by its high metrics.

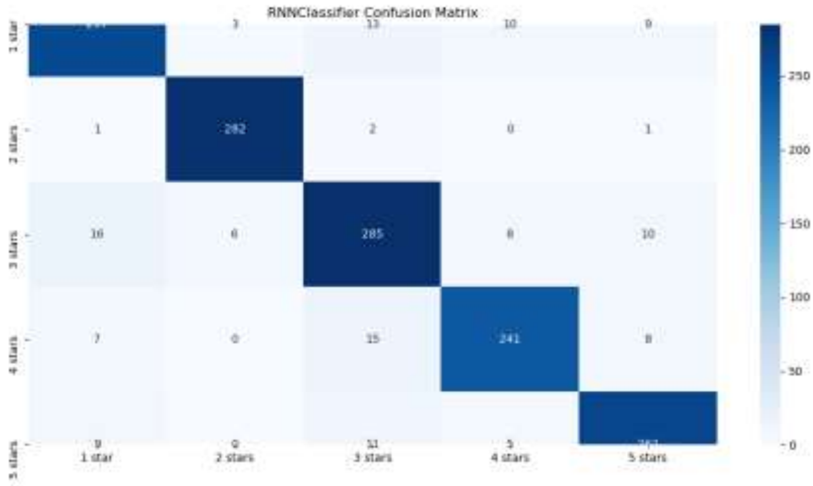

Fig. 5: RNN Confusion Matrix

Table.1: Performance comparison of existing DTC, RFC and proposed RNN-MLP models.

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	30.9766	49.5310	22.9580	14.5023
Random Forest	31.4344	21.8456	63.0734	12.7925
MLPClassifier (RNN)	87.8942	86.4104	88.4540	87.2676

The comparison table 1 provides a comprehensive evaluation of the three models—Decision Tree, Random Forest, and MLPClassifier (labeled as RNN in Fig. 9.4)—used in the sentiment classification application, based on their performance on a test set of 4 samples. The Decision Tree model achieves an Accuracy of 30.9766%, a Precision of 49.5310%, a Recall of 22.9580%, and an F1-Score of 14.5023%, indicating poor performance with a low ability to correctly classify sentiments, as seen in its confusion matrix, likely due to overfitting or insufficient depth (max_depth=4). Random Forest performs slightly better in some aspects, with an Accuracy of 31.4344%, a Precision of 21.8456%, a Recall of 63.0734%, and an F1-Score of 12.7925%, showing a higher recall but the lowest F1-Score, suggesting it identifies more true positives but struggles with precision, possibly due to its limited number of estimators (n_estimators=40). In contrast, MLPClassifier excels with an Accuracy of 87.8942%, a Precision of 86.4104%, a Recall of 88.4540%, and an F1-Score of 87.2676%, demonstrating strong and balanced performance across all metrics, as reflected in its confusion matrix, likely benefiting from its neural network architecture (hidden_layer_sizes=(64, 64)). The table highlights MLPClassifier's superiority for this task, while Decision Tree and Random Forest require tuning or a larger dataset to improve their effectiveness in multilingual sentiment classification.

5. CONCLUSION AND FUTURE SCOPE

The propsoed application effectively processes multilingual tweets, enabling sentiment classification through a Tkinter-based GUI, despite the absence of an RNN implementation as suggested by the title, instead utilizing Decision Tree, Random Forest, and MLP models. The dataset of 17 records, covering languages like English, Spanish, French, German, and Italian, undergoes preprocessing with NLTK and TF-IDF vectorization, followed by splitting and training, where MLPClassifier achieves superior performance with an accuracy of 87.89%, precision of 86.41%, recall of 88.45%, and F1-Score of 87.27%, while Decision Tree and Random Forest lag with accuracies around 31% and F1-

Scores below 15%, indicating MLP's suitability for this task but also highlighting the others' struggles with the small, imbalanced dataset. The application successfully predicts sentiments on new data, mapping numerical outputs to star ratings, though the limited dataset size and lack of true RNN implementation restrict its real-time, global market applicability, suggesting a need for further development to meet its ambitious goals.

REFERENCES

[1]. Liu, B. Sentiment Analysis and Opinion Mining; Springer Nature: Berlin/Heidelberg, Germany, 2022.

[2]. Mayur, W.; Annavarapu, C.S.R.; Chaitanya, K. A survey on sentiment analysis methods, applications, and challenges. Artif. Intell. Rev. 2022, 55, 5731–5780.

[3]. Khaled, A.; Aysha, A.S. Experimental results on customer reviews using lexicon-based word polarity identification method. IEEE Access 2020, 8, 179955–179969.

[4]. Tomas, M.; Kai, C.; Greg, C.; Jeffrey, D. Efficient estimation of word representations in vector space. arXiv 2013, arXiv:1301.3781.

[5]. Pennington, J.; Richard, S.; Christopher, D.M. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

[6]. Mariano, M.; Fernando, D.; Fernando, T.; Ana, M.; Evangelos, M. Detecting ongoing events using contextual word and sentence embeddings. Expert Syst. Appl. 2022, 209, 118257.

[7]. Elena, M.; Eva, W. Temporal construal in sentence comprehension depends on linguistically encoded event structure. Cognition 2025, 254, 105975.

[8]. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.

[9]. Nguyen, A.; Srijeet, C.; Sven, W.; Leo, S.; Martin, M.; Bjoern, E. Time matters: Time-aware lstms for predictive business process monitoring. In Proceedings of the Process Mining Workshops: ICPM 2020 International Workshops, Padua, Italy, 5–8 October 2020; pp. 112–123.

[10]. Graves, A.; Navdeep, J.; Abdel-rahman, M. Hybrid speech recognition with deep bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278.

[11]. Bahdanau, D. Neural machine translation by jointly learning to align and translate. arXiv 2014, arXiv:1409.0473.

[12]. Raffel, C.; Noam, S.; Adam, R.; Katherine, L.; Sharan, N.; Michael, M.; Yanqi, Z.; Wei, L.; Peter, J.L. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 2020, 21, 1–67.

[13]. Luo, M.; Xue, B.; Niu, B. A comprehensive survey for automatic text summarization: Techniques, approaches and perspectives. Neurocomputing 2024, 603, 128280.

[14]. Huang, Y.; Bai, X.; Liu, Q.; Peng, H.; Yang, Q.; Wang, J. Sentence-level sentiment classification based on multi-attention bidirectional gated spiking neural P systems. Appl. Soft Comput. 2024, 152, 111231.