Multilingual Toxic Comment Classification Using Bidirectional LSTM

Md. Nazmul Abdal, Md. Azizul Haque, Most. Humayera Kabir Oshie, and Sumaya Rahman

Abstract The growth of social networking sites and online platforms has brought about an unprecedented surge in user-generated content. However, along with the immense benefits of increased communication and information sharing, there has been an alarming growth in toxic and offensive comments. Detecting and moderating such comments is crucial to maintain a healthy and safe online environment. In this research, we propose a multilingual toxic comment classification system that leverages the power of Bidirectional Long Short-Term Memory (BiLSTM) neural networks. We use a comprehensive dataset which contains a diverse range of toxic comments in multiple languages. We employ a BiLSTM architecture because it is effective at detecting both contextual and sequential dependencies in text data. We train our model by combining word embeddings with character level embeddings in order to capture the semantic and morphological information found in the comments. Multiple cutting-edge methods are used to compare the model's performance, including RNN and LSTM. The experimental findings show that the suggested model performs competitively in classifying multilingual toxic comments, surpassing other approaches with an accuracy of 94.21%.

Keywords Multilingual toxic comment classification · Natural language processing · BiLSTM · Neural network

Md. N. Abdal (✉) · Md. A. Haque
Khulna University, Khulna 9208, Bangladesh
e-mail: mnabdal25@gmail.com

Most. H. K. Oshie
Jahangirnagar University, Savar, Dhaka 1342, Bangladesh

S. Rahman
Pundra University of Science and Technology, Rangpur Road, Gokul, Bogura, Bangladesh

R. N. Shaw et al. (eds.), *Innovations in Electrical and Electronic Engineering*, Lecture Notes in Electrical Engineering 1115, https://doi.org/10.1007/978-981-99-8661-3_23

1 Introduction

The emergence of social media and online platforms has fundamentally changed how we share information and interact. However, with the exponential increase in user-generated content, there has also been a concerning rise in toxic and offensive comments that pollute online discussions. A 2014 Research Institute survey indicated that 73% of adult internet users had seen someone being harassed in some way online, and that 40% of internet users had personally experienced online harassment with 45% of those experiencing substantial harassment [1]. Toxic comments not only propagate hate speech, harassment, and discrimination but also create a hostile environment for users. Therefore, it is now essential to identify and remove toxic comments in order to promote a welcoming and safe online community.

The problem of classifying toxic comments is extremely challenging in multilingual situations where comments are made in several languages and cultural contexts. Traditional approaches to classifying harmful comments sometimes rely on language-specific models, which limits their ability to handle several languages. Various strategies have already been recommended to tackle the issue of toxic comment classification, including numerous machine learning and deep learning algorithms [2]. However, the majority of machine learning-based classifiers depend on manually constructed features obtained from training data. Deep learning is an emerging branch of machine learning that has recently undergone significant progress due to the unanticipated rise in computing capacity. Our quality of life has substantially improved as a result of the widespread adoption of deep learning-based apps in our daily lives. Among the most successful and popular deep learning architectures, Recurrent Neural Network (RNN) [3] and its variants Long Short-Term Memory (LSTM) [4], and Gated Recurrent Unit (GRU) [5] have lately been applied to toxic comment identification.

Natural language processing (NLP) applications including text categorization [6], sentiment analysis [7], and question answering [8] have all seen impressive success in recent years because of LSTM. BiLSTM is a unique LSTM version that operates somewhat differently in order to perform better on some particular tasks. The main goal of BiLSTMs is to improve LSTM modeling by taking into account both the past and future contexts of each input sequence. BiLSTM processes sequences in two directions concurrently as opposed to normal LSTMs, which only process sequences in a forward direction. A more thorough understanding of the sequential data is made possible by this bidirectional processing, which enables the model to incorporate dependencies in both past and future contexts [9].

This paper presents a method for categorizing toxic comments using a BiLSTM-based model, driven by the requirement for a reliable multilingual toxic comment classifier. The capability of this kind of network to identify contextual and sequential relationships in text data is well known. By using this paradigm, we hope to create a multilingual harmful comment classification system that is quick and easy to use. To achieve our objective, we utilize a comprehensive and diverse dataset specifically curated for toxic comment classification in multiple languages. Our research focuses on creating a solid model architecture that can accurately identify the underlying trends and subtle differences in harmful comments across many languages. We make use of BiLSTM network's advantages to represent the contextual data included in comments, giving the system the ability to comprehend language's sequential nature and the effects it has on toxicity. Furthermore, we include attention techniques to draw attention to the most crucial sections of a comment, which helps the model provide correct predictions.

The results of this study have contributed to the creation of a multilingual toxic comment classification system that can recognize and categorize poisonous comments in a variety of languages with consistency. We seek to improve the functionality and adaptability of toxic comment classification models, enabling efficient moderation in multilingual online contexts. We also conduct comprehensive experiments and contrast the suggested model with existing techniques to show its effectiveness. The outcomes demonstrate the potential of our method for prompt actions, accurate and effective multilingual toxic comment identification, and improved online communication.

The following sections make up the remainder of the paper. In Sect. 2, we offer a summary of the related studies in multilingual toxic comment classification. Section 3 describes the methodology, including the MobileNetV2 architecture. The results are presented in Sect. 4, whereas we compare the results with other research in Sect. 5. Finally, Sect. 6 provides the findings of the study and recommendations for the future.

2 Related Work

Researchers have made numerous attempts to classify toxic comments using various methodologies over the years. These classification techniques frequently use well-known machine learning models as their foundation. Recent advances in deep learning techniques have led to the development of an increasing number of methods for addressing the toxic comment identification problem. These approaches include RNN, LSTM, and GRU.

The authors of [10] set out to analyze any text in order to spot several types of toxicity, including vulgarity, threats, insults, and hatred fueled by identity. They used the designated Wikipedia Comment Dataset for their work. A 6-headed machine learning model was used by then which attained an absolute validation accuracy of 98.08%. Rahul et al. [11] looked at the scope of harassment on the internet and labeled the content to analyze the toxicity as accurately as possible. They deployed six machine learning algorithms to their data to address the text classification problem, and depending on their evaluation metrics for the categorization of harmful comments, they chose the best machine learning method. Additionally, they sought to accurately assess the toxicity with the goal of reducing any negative impacts. In [12], the authors created a powerful model to identify and categorize toxicity in user-generated content on social media using the transformer-based BERT model. A well-known labeled harmful comment dataset was used to fine-tune the pretrained model and three of its variants. They also tested the suggested approach using two datasets collected from Twitter over two distinct time periods in order to identify toxicity in user-generated content. According to their findings, the system could effectively classify harmful tweets. The Kaggle toxic comment dataset was used by the authors of study [13] to train a deep learning network as they classified the comments into the following categories: toxic, severe toxic, obscene, threat, insult, and identity hate. Several deep learning methods were used to train the dataset. They also examined the superior deep learning model for comment classification. Kulkarni et al. [14] introduced a multichannel convolutional network based on bidirectional GRU for spotting offensive comments in a multilabel setting. The proposed approach generated word vectors with pretrained word embeddings. Additionally, their hybrid model gathered regional features using a variety of filters and kernel sizes. According to their tests, the suggested model performed better in terms of multilabel criteria.

In [15], the authors suggested an effective approach to word representation that produced weighted word vectors while including sentiment information into the well-known TF-IDF algorithm. By using a feed-forward neural network classifier, they were able to determine the comment's sentimental propensity. Their approach was contrasted with those of RNN, CNN, LSTM, and Naive Bayes for sentiment analysis under the identical settings. Xie [16] provided a methodology for creating toxicity models. To increase the classification accuracy, he used subsample, pseudo-labeling with accessible subtitles, converting non-English languages to English, and post-processing. The model successfully performed under cross-lingual toxicity detector with an AUC of 0.9469 for the initial training dataset and 0.9485 for the testing data. Based on LSTM and BiLSTM, the authors of article [17] created a model for classifying harmful comments. For their work, they used the Kaggle benchmark harmful comment dataset. Both suggested strategies' accuracy ratings were assessed and contrasted. Finally, they demonstrated that the Bi-LSTM algorithm outperformed LSTM with an improved accuracy of 98.07%. Dubey et al. [4] used LSTM neural networks to build a model for classifying harmful comments. Their technology successfully classified and identified hate speech, allowing it to be excluded. The program could classify provided comments as dangerous or harmless with 94.49% precision, 92.79% recall, and a 94.94% accuracy score. For the purpose of detecting cyberbullying, the authors in [18] developed a LSTM-CNN architecture. They trained the customized word embeddings on which they constructed their model using word2vec. They used comments and tweets to test their framework. Additionally, they developed a website that made use of the algorithm to categorize tweets as bullying or not, depending on its toxicity level and other factors. The technique was also applied to the Telegram Bot, which monitors and stops online harassment. The ROC-AUC score for the model developed by the authors was 97%. Li et al. [19] suggested a modified model for detecting harmful behavior based on Bi-LSTM technique. Using the updated SMOTE algorithm, they boosted the representation of minorities in the detrimental comment material. They also combined the text vector with the model for detection. The results of the studies showed that the model outperformed other currently in use models in classification accuracy and improved total detection.

3 Methodology

This section provides a complete presentation of our RNN-based multilingual toxic comment classification process using BiLSTM architecture.

3.1 Data Collection

We make use of Kaggle's Jigsaw Multilingual Toxic Comment Classification dataset for our research [20]. It includes a substantial number of user comments in several languages along with appropriate toxicity labels. The dataset contains comments from several online sites that span a wide range of subjects and material. The level of toxicity included in each comment is denoted by a toxicity score, which ranges from 0 to 1. Languages including English, Spanish, French, German, Italian, Portuguese, Russian, and Turkish are included in the dataset. For the initial model, we use a training set comprising 223,549 samples. 8000 samples make up the validation set of data. We show the distribution of data in both the training and validation dataset in Fig. 1.

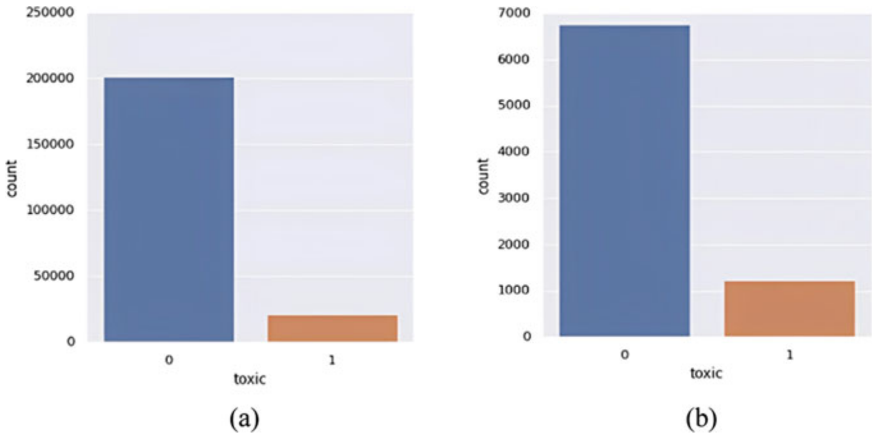


Fig. 1 Distribution of samples in the dataset: **a** training data and **b** validation data

3.2 Data Preprocessing

We use several preprocessing techniques to get the dataset ready for the model training. Firstly, we clean the comment text by removing HTML tags, special characters, and URLs. The purpose of this phase is to clean up the comments and standardize their format. The cleaned comments are then tokenized into distinct words using a tokenizer. This method makes it easier to transform text data into numerical form. The tokenized comments are then padded to a predetermined length to maintain consistency throughout the input sequences of data. Finally, we represent the tokens in a continuous vector space using word embeddings.

3.3 Conventional LSTM Model

LSTM is a form of RNN architecture that solves the limitations of conventional RNNs in preserving dependence over time in historical data. It uses memory cells and gating methods to address the issue of vanishing gradient [21]. The internal structure of an LSTM unit is shown in Fig. 2, which serves as the fundamental of an LSTM network. The LSTM unit has four feedforward neural networks. There are input and output layers in each of these neural networks. All of the output neurons have connections to all of the input neurons across all of these neural networks. The result is an LSTM unit with four fully interconnected layers. Information selection is handled by three of the four feedforward neural networks. They are the input gate, output gate, and forget gate. All three common memory management operations, including erasing data from memory, adding data to memory, and using data that is already in memory are carried out via these three gates. The fourth neural network develops fresh knowledge that could be stored in the memory [22]. The ability to selectively remember or forget information provided by the LSTM memory cell enables the network to recognize long-lasting dependency in sequential information. The different gating techniques regulate the information flow and aid in solving the vanishing gradient issue. LSTM networks often have a deep LSTM design made up of several stacked LSTM cells. This enables the network to understand the intricate temporal correlations and patterns of the incoming data.

3.4 Bidirectional LSTM Architecture

The bidirectional LSTM is a development of the LSTM architecture that uses past and present information to forecast sequential data. BiLSTMs capture interdependence from past and future contexts by interpreting the input sequences both forward and backward, providing a more thorough knowledge of the sequential data [23]. The BiLSTM may efficiently capture long-term dependencies and context in both

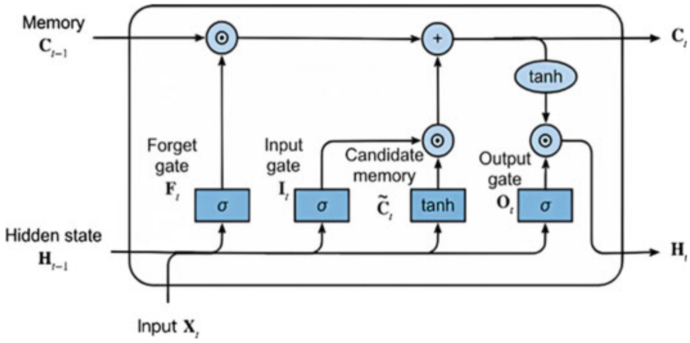


Fig. 2 Internal organization of LSTM [22]

directions by merging forward and backward information. This design is especially beneficial for tasks where it is crucial to comprehend the context from both the past and the future. Two unidirectional LSTMs that process the sequence in forward and backward directions comprise the bidirectional LSTM architecture. It is possible to think of this architecture as having two different LSTM networks. One receives the tokens in their current order, while the other receives them in the opposite direction [24]. The output of these LSTM networks is a probability vector, and the combined probability of both is the output. At each time step, the hidden states from both LSTMs are merged to create a final output, often via concatenation, sum, or averaging. The model can produce more accurate predictions at each time step due to the forward and backward context mix [25]. Figure 3 depicts the operation of the BiLSTM architecture.

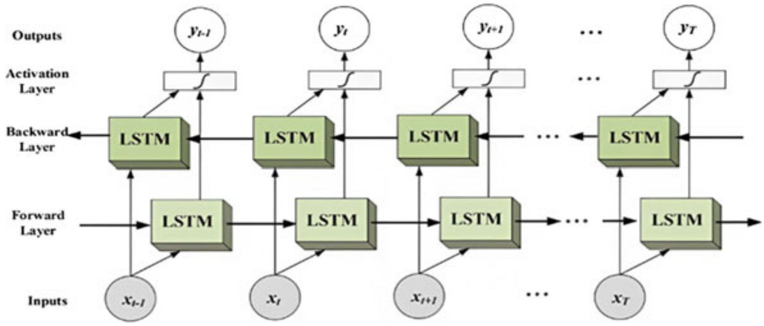


Fig. 3 Working procedure of BiLSTM [23]

3.5 Hyperparameter Tuning

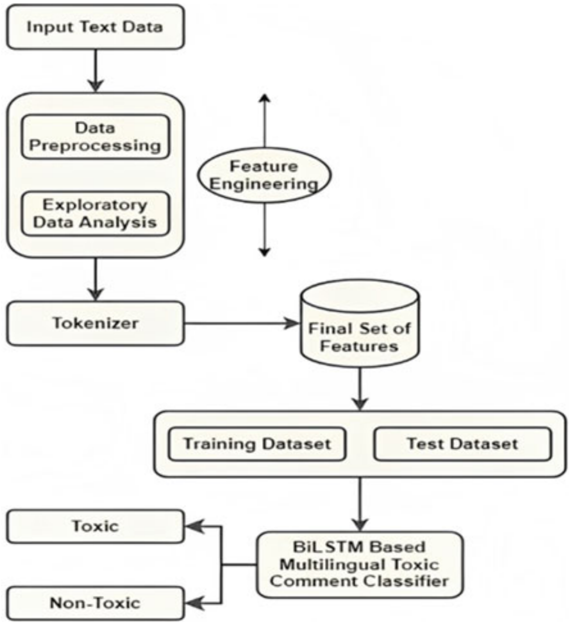
The model must be tuned using its hyperparameters in order to perform better. The number of layers, choice of optimizer, activation function, loss function, learning rate, batch size, and number of epochs are all hyperparameters in our proposed model. Tuning these hyperparameters should provide the final model with the best performance. We utilize the Adam optimizer to update the weights during training because it has a number of benefits, including shorter training times and computational efficiency with fewer memory use [26]. In our trials, we have used batch size 32. Furthermore, binary cross entropy is the loss function, and sigmoid is the activation function.

3.6 Multilingual Toxic Comment Classification Process

In our research, some preprocessing steps are applied to the initial data. Thus, we get a new dataset with clean text. After that, training data and testing data are created from the complete dataset. The model is trained using 80% of the training data, while the remaining 20% is randomly divided and utilized for validation. The proposed framework is then used to build the classification model from the data. The effectiveness of our model is then assessed using this classification on the test dataset with new comments. In order to evaluate the effectiveness of the suggested model, we have conducted a number of experiments in various configurations. The model is instructed to adjust a number of network settings to find the best combination of parameters. Some additional experiments are also conducted in our study by altering models like RNN and LSTM.

The general workflow of our proposed toxic comment classifier is illustrated in Fig. 4. The first step involves sending the raw texts to the feature engineering stage, where preprocessing and exploratory data analysis techniques are used to obtain preprocessed data. To construct the final features, preprocessed data is put into the tokenizer along with the embedding stage. Our last set of features will be padded in the next phase before being split into training and test datasets. After that, the classifier receives the training datasets and builds the model that categorizes the toxic and non-toxic comments. The effectiveness of our model is then assessed using this classification on the test dataset. The Python programming language is used to conduct each experiment on a Kaggle notebook with 13 GB of RAM, 16 GB of GPU memory, and 73 GB of disk space.

Fig. 4 Workflow of the proposed multilingual toxic comment classifier

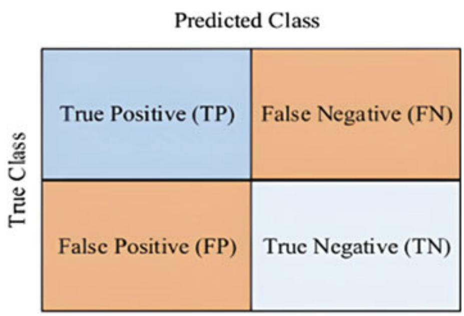


3.7 Performance Evaluation

During the evaluation process, a test dataset containing texts and associated labels is fed to the model. At this point, there is no mechanism for updating weights. In order to calculate model performance, the texts are fed into the model, and after the model categorizes them, we contrast the model classification with the appropriate label for the input data. The confusion matrix must be computed first to choose the optimal method for calculating the model performance. Figure 5 illustrates a basic confusion matrix.

True positive, true negative, false positive, and false negative values are the components of a confusion matrix. The values in the diagonal position show how accurately

Fig. 5 Confusion matrix



the model predicted the data [27]. For evaluating the performance of our model, we employ four metrics: accuracy, precision, recall, and F1-score. The evaluation metrics are calculated based on the confusion matrix by using the following equations:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - \text{Score} = 2 \times \frac{\text{Precison} \times \text{Recall}}{\text{Precison} + \text{Recall}} \tag{4}$$

4 Results

Besides our model, we have also performed training with two other models to show the comparison of results with our proposed system. This section discusses all the experimental results from our BiLSTM-based multilingual toxic comment classification model.

4.1 Classification Report

Our proposed model has performed admirably, with an accuracy of 94.21%. The precision, recall, and F1 score have all reached outstanding levels, which shows how well our suggested model performs. Our model possesses a lower total parameter count compared to the parameter count of other models currently in use. The classification report of the proposed model is summarized in Table 1.

Table 1 Classification report of the proposed model

Model	Accuracy	Precision	Recall	F1-Score
BiLSTM	0.94	0.94	0.95	0.94

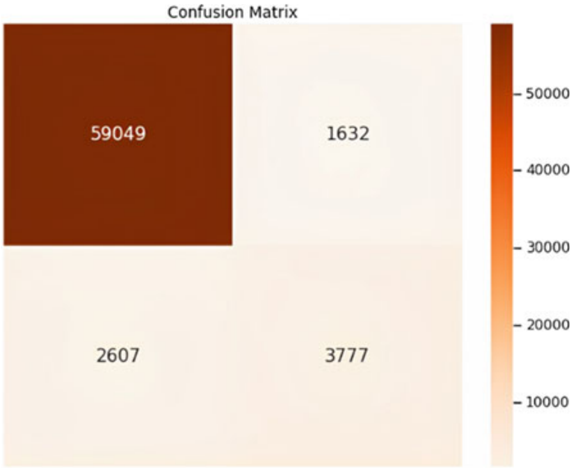


Fig. 6 Confusion matrix of the proposed model

4.2 Confusion Matrix

Confusion matrix is a tabular representation that contrasts a classification model's predicted values for a given dataset with the actual values, in an effort to assess the accuracy of the model's performance. We show the confusion matrix of our toxic comment classification task in Fig. 6. The confusion matrix demonstrates the model's outstanding performance with previously unseen data.

4.3 Accuracy Curve

Accuracy is used to determine a model's consistency in training and validation datasets. The validation set performs a different function from the training set by assessing the model's efficacy on unobserved data, in contrast to the training set's primary function of parameter adjusting [28]. In our results, we show the accuracy curve to support the model's performance on the dataset. The accuracy curve depicted in Fig. 7 illustrates the satisfactory convergence of the proposed model through the training and validation stages. Despite a few low spikes, the validation curve exhibits a steady validation accuracy over most portions of the curve.

4.4 Loss Curve

We also include the loss curve of our model in Fig. 8. The loss curve shows the overall distribution of losses for both the training and validation stages regarding the quantity of epochs. The y-axis of the plot displays the loss function's value, while

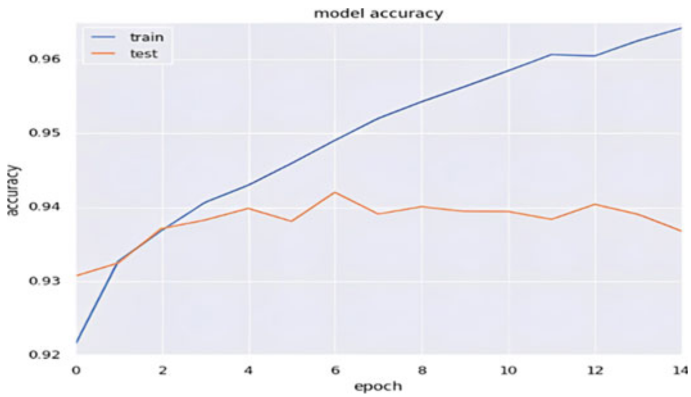


Fig. 7 Accuracy curve for the training process

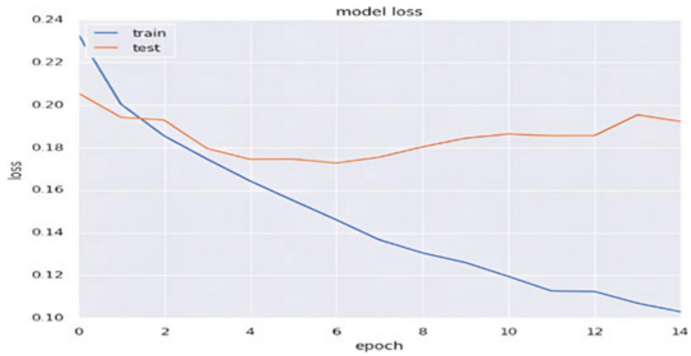


Fig. 8 Loss curve for the training process

its x-axis indicates the number of iterations [29]. The high loss values in the graph demonstrate how far the model's predictions diverge from the actual labels. When the loss value is low, the model gets better at making predictions and learning from the training set.

5 Comparative Analysis

We perform our experiments by modifying the BiLSTM model. After getting the results, we evaluate the model with other architectures. We analyze to determine which method is more suitable for this classification task using the same dataset. We compare our system with two other baseline models, including RNN and LSTM.

For the suggested model, the training process convergence is very high. As a result, the model can accurately predict the test data and is quick to learn. We also

Table 2 Convergence comparison with two models

Model	Epoch	Accuracy (%)
RNN	50	89.39
LSTM	35	92.03
BiLSTM	15	94.21

want to understand the different elements that affect the convergence of the model. To test this, we vary the number of epochs while attempting to maintain a constant target accuracy. Table 2 provides an overview of our findings. It shows that RNN and LSTM require more time to deliver the target accuracy than BiLSTM. Additionally, the losses of these models are greater.

6 Conclusion

This work proposes a novel method for classifying harmful comments in multilingual contexts based on the bidirectional LSTM architecture. The Jigsaw Multilingual Toxic Comment Classification dataset has been utilized to evaluate the performance of our proposed approach and conducted a comparative analysis against several baselines. Our results indicate that the BiLSTM model performs well at capturing long-term dependencies and context, enabling accurate classification of toxic comments across multiple languages. Through extensive experimentation and evaluation, we have observed that our proposed approach outperformed all the baseline models regarding accuracy, precision, recall, and F1-score. The BiLSTM model achieved an accuracy of 94.21%, showcasing its superior performance in multilingual toxic comment classification. Furthermore, language-specific evaluation metrics consistently demonstrated the robustness of the BiLSTM model across different languages present in the dataset. Our research contributes to natural language processing and text classification by providing a powerful multilingual toxic comment classification framework. The proposed BiLSTM architecture can be applied in various online platforms and social media environments to automatically identify and flag toxic comments, promoting safer and more inclusive online communities.

Future work in this area could focus on exploring additional techniques to enhance the performance of multilingual toxic comment classification. This may involve incorporating ensemble methods or exploring more advanced pretraining techniques using large-scale multilingual language models. Moreover, investigating ways to handle class imbalance and the detection of subtle forms of toxicity could further improve the accuracy and applicability of the classification model.

References

1. Online Harassment (2023). https://www.pewresearch.org/internet/2014/10/22/online-harassment/. Last Accessed 06 June 2023
2. Bonetti A, Martínez-Sober M, Torres JC, Vega JM, Pellerin S, Vila-Francés J (2023) Comparison between machine learning and deep learning approaches for the detection of toxic comments on social networks. Appl Sci 13(10):6038
3. Nazar S, Rajan R (2022) Multi-label comment classification using GloVe-RNN framework. In: 19th India council international conference (INDICON), pp 1–4
4. Dubey K, Nair R, Khan MU, Shaikh S (2020) Toxic comment detection using lstm. In: Third international conference on advances in electronics, computers and communications (ICAECC), pp 1–8
5. Wang Z, Zhang B (2021) Toxic comment classification based on bidirectional gated recurrent unit and convolutional neural network. Trans Asian and Low-Resour Lang Inform Process 21(3):1–12
6. Huan H, Guo Z, Cai T, He Z (2022) A text classification method based on a convolutional and bidirectional long short-term memory model. Connect Sci 34(1):2108–2124
7. Bhuvaneshwari P, Rao AN, Robinson YH, Thippeswamy MN (2022) Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model. Multimedia Tools and Appl 81(9):12405–12419
8. Balla HA, Llorens Salvador M, Delany SJ (2022) Arabic medical community question answering using ON-LSTM and CNN. In: 14th international conference on machine learning and computing (ICMLC), pp 298–307
9. Hameed Z, Garcia-Zapirain B (2020) Sentiment classification using a single-layered BiLSTM model. IEEE Access 8:73992–74001
10. Chakrabarty N (2020) A machine learning approach to comment toxicity classification. In: Computational intelligence in pattern recognition: proceedings of CIPR 2019, Springer, Singapore, pp 183–193
11. Kajla H, Hooda J, Saini G (2020) Classification of online toxic comments using machine learning algorithms. In: 4th international conference on intelligent computing and control systems (ICICCS), pp 1119–1123
12. Fan H, Du W, Dahou A, Ewees AA, Yousri D, Elaziz MA, Al-qaness MA (2021) Social media toxicity classification using deep learning: real-world application UK Brexit. Electronics 10(11):1332
13. Anand M, Eswari R (2019) Classification of abusive comments in social media using deep learning. In: 3rd international conference on computing methodologies and communication (ICCMC), pp 974–977
14. Kumar A, Abirami S, Trueman TE, Cambria E (2021) Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit. Neurocomputing 441:272–278
15. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7:51522–51532
16. Xie G (2022) An ensemble multilingual model for toxic comment classification. In: International conference on algorithms, microchips and network applications, vol 12176. pp 429–433
17. Gupta A, Nayyar A, Arora S, Jain R (2020) Detection and classification of toxic comments by using LSTM and bi-LSTM approach. In: International conference on advanced informatics for computing research, pp 100–112
18. Gada M, Damania K, Sankhe S (2021) Cyberbullying detection using LSTM-CNN architecture and its applications. In: International conference on computer communication and informatics (ICCCI), pp 1–6
19. Li S, Huang S, Zhou Y (2020) Toxic behaviour detection based on improved SMOTE algorithm and bi-LSTM network. Int J Intell Internet Things Comput 1(2):114–128
20. Jigsaw Multilingual Toxic Comment Classification (2023). https://kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification. Last Accessed 11 June 2023

21. Staudemeyer RC, Morris ER (2019) Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586
22. An Intuitive Explanation of LSTM (2023). https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c. Last Accessed 20 June 2023
23. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991
24. Bidirectional LSTM in NLP (2023). https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/. Last Accessed 25 June 2023
25. Bidirectional LSTM (2023) .https://saturncloud.io/glossary/bidirectional-lstm/. Last Accessed 26 June 2023
26. Konur O (2013) Adam optimizer, energy education science and technology part B: social and educational studies
27. Understanding Confusion Matrix (2023). https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. Last Accessed 28 June 2023
28. Accuracy, Precision, and Recall in Deep Learning, https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/, last accessed 2023/07/02
29. Loss and Loss Functions for Training Deep Learning Neural Networks (2023). https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/. Last Accessed 08 July 2023