Neurocomputing 441 (2021) 272–278

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit

Ashok Kumar J [a], Abirami S [a], Tina Esther Trueman [a], Erik Cambria [b,*]

[a] Department of Information Science and Technology, Anna University, Chennai 600025, India
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 15 January 2021
Accepted 8 February 2021
Available online 2 March 2021
Communicated by Zidong Wang

Keywords:
Multilabel classification
Multichannel
Convolutional neural network
Bidirectional recurrent neural networks
Toxic comment classification
Multilabel metrics

ABSTRACT

Recently, toxicity identification has become the most serious problem in online communities and social networking sites. Therefore, an automatic toxic identification system needs to be developed for preventing and limiting users from these online environments. In this paper, we present a multichannel convolutional bidirectional gated recurrent unit (MCBiGRU) for detecting toxic comments in a multilabel environment. The proposed model generates word vectors using pre-trained word embeddings. Moreover, this hybrid model extracts local features with many filters and different kernel sizes to model input words with long term dependency. We then integrate multiple channels with a fully connected layer, normalization layer, and an output layer with a sigmoid activation function for predicting multilabel categories. The experimental results indicate that the proposed MCBiGRU model outperforms in terms of multilabel metrics.

1. Introduction

Online social networking sites provide a platform for people to anonymously share and express their opinions [1]. Sometimes, such opinions can be harassive, abusive, or trollicious to others and cause some individuals to stop sharing, getting depressed, or even have suicidal thoughts [2]. Therefore, an automatic system needs to be developed to avoid, remove, or flag such unhealthy contents from online platforms [3,4]. The development of such a toxicity identification system, however, is a very challenging task for online platform providers. Natural language processing (NLP) helps to identify toxicity in texts, which are expressed as posts or comments. These comments are naturally associated with multiple toxic labels such as insult, threat, and toxicity. This paper aims to focus on multilabel category detection. Moreover, various machine learning and deep learning approaches were proposed to solve the task of multilabel category detection. In traditional machine learning, the multilabel classification problem is solved using the problem transformation, adapted algorithms, and ensemble approaches. These approaches employed the bag-of-words (BoW) model representation.

However, the BoW model fails to capture semantic meaning among words [5,6]. Furthermore, deep learning techniques solve the multilabel classification problem with promising results [7,8,9]. These techniques capture the semantic meaning among words using word embeddings. In this paper, we propose a multichannel convolutional bidirectional gated recurrent unit (MCBiGRU) for multilabel category detection. The concept of multichannel represents the standard version of the same model (convolutional bidirectional gated recurrent unit) with different word embeddings. First, the proposed MCBiGRU model creates word vectors for the training data using pre-trained word vectors [10]. Second, these word vectors are passed to the embedding layer to capture the semantic meaning of words. Third, a convolutional neural network (CNN) is used to extract local features with many filters and different kernel sizes. Fourth, the extracted features are fed into the bidirectional gated recurrent unit (BiGRU) to model input words sequentially with long term dependency. These standard steps are repeated for each channel. We then integrate multiple channels using the concatenation layer. This result passed to the fully connected layer, normalization layer, and the output layer with a sigmoid activation function for predicting multilabel categories. The experimental results reveal that the proposed MCBiGRU model outperforms in terms of precision, recall, and F1-Score with macro, micro, and weighted score for examples and labels.

The remainder of this paper is structured as follows: Section 2 presents the related works for multilabel toxic category detection; Section 3 explains the proposed MCBiGRU model in detail; Section 4 illustrates the multilabel evaluation metrics; Section 5 sum-

* Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.neucom.2021.02.023
0925-2312/

marizes the results and their comparison with others; finally, Section 6 presents the conclusion and future works.

2. Related works

Text representation and classification significantly play an important role in the field of NLP [11]. Toxic comment classification is a NLP task close to sentiment analysis [12,13] which aims to further categorize negative comments into toxic (comments that are threatening, obscene, or insulting) versus non-toxic (comments that simply express a negative opinion). In recent years, toxic comment classification has been increasingly leveraging machine learning and deep learning models. Hossein et al. [14] proposed an attack on Google's perspective API for toxicity identification. The authors showed the detailed examples that a highly toxic phrase assigns lower toxic scores in the perturbing abusive phrases and high toxic scores to benign phrases. Ibrahim et al. [8] presented three data augmentation techniques, namely, unique words augmentation, random mask, and synonyms replacement to deal with the class imbalance problem. Moreover, the authors proposed an ensemble model (CNN, Bidirectional LSTM, and GRU) to detect toxicity in user-generated content. Anand and Eswari [15] studied various deep learning models (e.g., CNN, and LSTM) with and without pre-trained word embeddings for abusive comment classification. This study shows that CNN outperforms the pre-trained GloVe word embeddings. Pavlopoulos et al. [16] examined the performance of RNN with Greek news portal user comments and Wikipedia comments. This study indicates that the GRU method outperforms the logistic regression, multilayered perceptron, and CNN models. Georgakopoulos et al. [7] investigated the performance of CNN with text mining methodologies. Their study suggested that CNN enhances the task of toxicity identification. Mohammad [17] demonstrated the transformation of raw comments with four classification models, namely, logit, NBSVM, FastText-BiLSTM, and XGBoost. The author reveals that the models achieve a relatively decent result without any transformation.

Van Aken et al. [18] presented an ensemble model with multiple approaches for addressing the challenges in toxic comment classification. Especially, this ensemble model outperforms with high data variance and classes with few instances. Saeed et al. [19] studied Deep Neural Network architectures and their comparison for overlapping toxic sentiment data. The authors show that the Bi-GRU model outperforms in the task of overlapping multilabel toxic sentiment classification. Khieu and Narwal [4] studied the LSTM model and Kohli et al. [20] studied the LSTM model with custom embeddings for toxicity identification. Moreover, Rezaeinia et al. [22] proposed multiple block convolutional highways for text categorization. This study improved the performance of CNN by introducing improved word vectors. Quan et al. [24] proposed multichannel CNN for biomedical relation extraction. This study indicates that the proposed MCNN can deal with long sentences. Yoon and Kim [23] applied multichannel lexicon embeddings on CNN-BiLSTM to improve the performance of sentiment classification. Zhang et al. [25] proposed a multichannel convolutional long short-term memory network (CNN-LSTM) for sentiment classification. The authors revealed that the proposed model outperforms all baseline algorithms. Li et al. [26] proposed a two-channel convolutional gated recurrent unit for stance detection. Their experimental results have shown a 15.6% improvement in the SVM (support vector machine) method. In summary, the existing researchers used multichannel deep learning models for the single label and multiclass problems. Therefore, we propose a multichannel convolutional bidirectional gated recurrent unit for multilabel toxic category detection.

3. The proposed method

In this section, we describe the proposed MCBiGRU model for multilabel category detection. The multichannel represents the standard version of the same hybrid model (convolutional bidirectional gated recurrent unit) with different window sizes [24,26] as shown in Fig. 1. In particular, we discuss the main components of the proposed model as follows.

3.1. Dataset

We use the Wikipedia talk pages dataset published by Google Jigsaw on Kaggle [40]. This dataset includes 223,549 instances with six labels, namely, toxic, obscene, severe toxic, insult, threat, and identity hate. These labels define an instance as toxicity or non-toxicity. In particular, it is one of the largest datasets with class imbalance. Moreover, 201,081 instances were assigned with a 'clear' category matching none of the above six labels. 'Threat' is the least category in the dataset as shown in Table 1.

3.2. Multichannel convolutional bidirectional gated recurrent unit

The multichannel convolutional bidirectional gated recurrent unit represents the multiple version of the standard CNN model with different sizes of kernels. This representation allows the instance or document to process in different n-grams such as 1-gram, 2-gram, and 3-grams at the same time [32]. In particular, we define the standard CNN model with a word embedding layer, one-dimensional convolutional layer, dropout layer, max-pooling, bidirectional gated recurrent unit, and dropout layer. This standard version is defined with five channels for different n-grams. Each component of the channel is explained as follows.

3.2.1. Multichannel word embedding

To obtain the quality of data, we remove punctuations and special symbols to represent each word into a numeric vector for toxic comments [21,22,26]. We then use pre-trained word embeddings to generate word vectors to capture semantic information from the training data. In particular, we use the GloVe word embeddings [10] with 100 dimensions to capture the semantic meaning of words. In the multichannel environment, we generate word embeddings for each channel with different contexts or window sizes from the same training data. The advantage of multichannel word embeddings is to extract different input features parallel on the same training data within a model [23]. Moreover, the learned word vectors are not updated during the model training.

3.2.2. Convolutional neural network

The CNN is widely used in the applications of image classification, image and video recognition, recommender systems, and NLP [31,32,7,22]. The CNN is passed over an input sequence with many filters in a fixed-length vector to produce new feature maps at different positions [26]. Specifically, we use a 1D-Convolution layer with many filters (W) and five different kernel sizes (h) separately for multichannel environments. Let $W^i \in R^{hd}$ be the filter for channel i in dimension d. Let $V^i \in R^{Nd}$ be the word embeddings for channel i with the maximum input sequence length N. Then, features m_k are generated as in (1).

$$m_k = \left(\sum_{i=1}^{c} V^i[k : k+h-1] \otimes W^i + b \right) \tag{1}$$

$$C = [m_1, m_2, m_3, \ldots, m_k] \tag{2}$$

where $V^i[k : k+h-1]$ denotes the generated input feature vectors by connecting row k to row $k+h-1$ in inputs V^i, f represents an

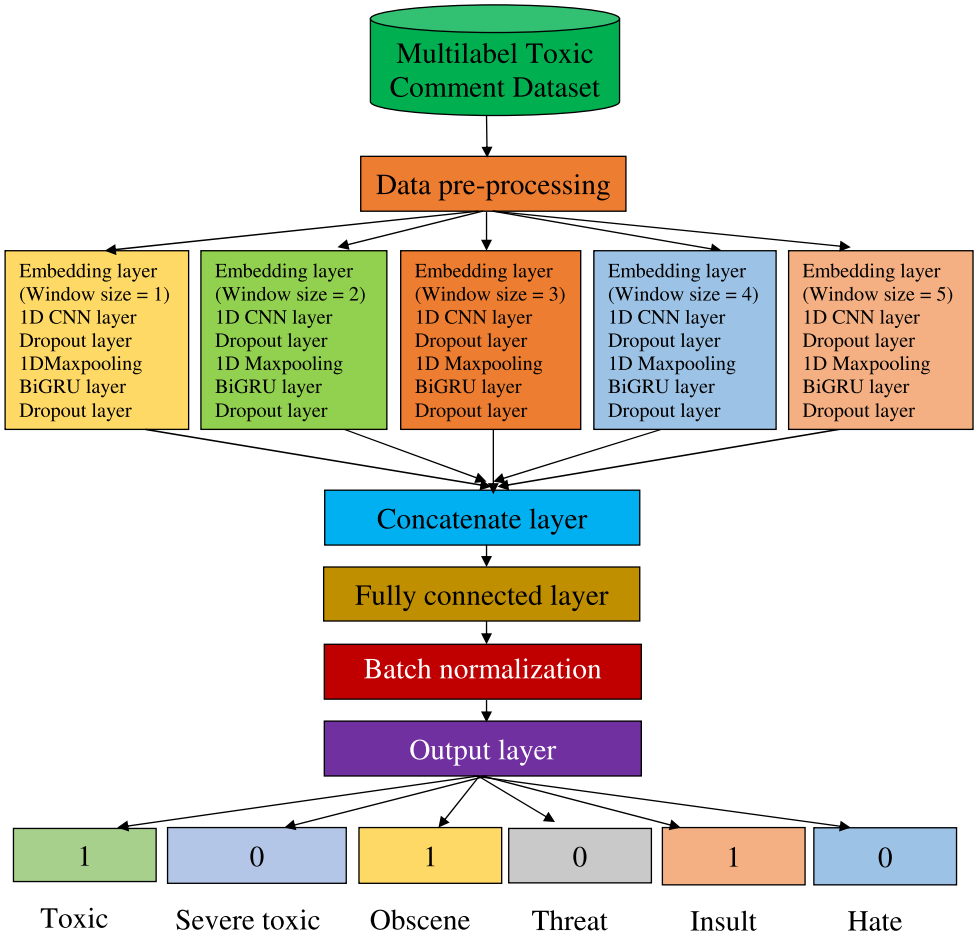

Fig. 1. The proposed MCBiGRU model.

Table 1
Label distribution for Wikipedia talk pages dataset.

Code	Label	Occurrences	%
0	Toxic	21,384	8.53
1	Severe toxic	1962	0.78
2	Obscene	12,140	4.84
3	Threat	689	0.27
4	Insult	11,304	4.51
5	Identity hate	2117	0.84
–	Clean	201,081	80.23

activation function, b denotes a bias term, and $\otimes$ denotes element-wise multiplication. Moreover, a new feature map C is produced by applying a filter to each window for input sequences as in (2).

3.2.3. Pooling layers

Pooling layers are an integral part of CNN. Its main purpose is to reduce the dimension of input feature maps. In particular, the pooling layers produce sub-sampling upon each feature map. This pooled feature map size is smaller than the generated feature map [7]. In this work, we use maximum pooling operation upon different channels to calculate the maximum feature value of the local neighborhoods for each feature map.

3.2.4. Bidirectional gated recurrent unit

Recurrent Neural Networks takes words in sequential order for interpreting a document. The RNN is difficult to train due to the long-range dependencies. To overcome this problem, the variant of RNN is introduced such as LSTM and GRU [33,35]. The LSTM network controls the input sequence with three gating mechanisms, namely, forget gate, input gate, and update gate. The GRU network controls the input sequences with the update gate and the reset gate [26]. In this paper, we use the bidirectional GRU on the top of CNN and the max-pooling layer for each channel. It memorizes the past and future semantic information. The reset gate combines the previous input and new input, and the update gate preserves the required memory block as in (3)-(7). Moreover, the GRU works faster than LSTM and it takes less computation to update hidden states.

$$u = \sigma(W_u h_{t-1} + U_u x_t + b_u) \tag{3}$$
$$r = \sigma(W_r h_{t-1} + U_r x_t + b_r) \tag{4}$$
$$c = \tanh(W_c(h_{t-1} \otimes r) + U_c x_t + b_c) \tag{5}$$
$$\overrightarrow{h_t} = \sigma(u \otimes c) \oplus (1-u) \otimes h_{t-1}) \tag{6}$$
$$y_t = V\left(\overrightarrow{h_t} : \overleftarrow{h_t}\right) \tag{7}$$

3.3. Output layer

The result of the convolutional bidirectional gated recurrent unit for each channel is fed into the concatenation layer, which takes the same input size and concatenates them in a specified dimension [34]. The output of this layer is passed to the dense layer and normalization layer where the dense layer changes the dimension of vectors for updating the trainable parameters and

the normalization layer allows the model to learn more independently on each layer of this network. We then use the output layer with a sigmoid activation function [36] for predicting multilabel categories. The proposed model uses binary cross-entropy as a loss function to decide whether an instance belongs to a category or not for each label.

4. Multilabel evaluation metrics

In this paper, we describe the multilabel evaluation metrics such as sample-based, label-based, and rank-based metrics [37–39] to deal with toxic classification problem. In particular, we discuss Hamming loss, exact match, and precision, recall, and F1-score with samples, macro, micro, and weighted scores. Let $D = (x_i, Y_i), i = 1, 2, 3 \ldots N, Y_i \subseteq L$, be the training samples assigned with true labels (Y_i). Let L be the set of all true labels and H be the learned model. Then, the predicted labels defined as $Z_i = H(x_i)$.

4.1. Sample-based metrics

Sample-based metrics evaluate the average difference between the true labels and predicted labels over all samples on the evaluation dataset [37]. In these metrics, we discuss Hamming Loss (HL), exact match (EM), precision (P), recall (R), and F1-score (F1) as follows. The HL and EM measure the average of incorrectly classified labels (8) and correctly classified labels (9) overall samples, respectively. Similarly, precision and recall are defined as the fraction between true labels and predicted labels as in (10) and (11). F1-score is calculated by taking the harmonic mean between precision and recall as in (12).

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{|L|} \tag{8}$$

where $\triangle$ denotes the difference between the true and predicted labels.

$$EM = \frac{1}{N}\sum_{i=1}^{N} I(Y_i = Z_i) \tag{9}$$

where,

$$I\left(Y^{(i)} = Z^{(i)}\right) = \begin{cases} 1 & \text{iff} \quad Y^{(i)} \text{ and } Z^{(i)} \text{ are identical} \\ 0 & \text{otherwise} \end{cases}$$

$$P = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{Z_i} \tag{10}$$

$$R = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{Y_i} \tag{11}$$

$$F1 = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \bigcap Z_i|}{|Y_i| + |Z_i|} \tag{12}$$

4.2. Label-based metrics

Label-based metrics evaluate the process for each label separately based on true positives (tp), false positives (fp), false negatives (fn), and true negatives (tn) [37]. We define the precision (P), recall (R), and F1-score ($F1$) with their macro, micro, and weighted scores as follows in (13)-(25).

$$P = \frac{tp}{tp + fp} \tag{13}$$

$$R = \frac{tp}{tp + fn} \tag{14}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{15}$$

$$A = 2 \times \frac{tp + tn}{tp + fp + fn + tn} \tag{16}$$

$$P_{mac} = \frac{1}{L}\sum_{i=1}^{L} P \text{ of } i \tag{17}$$

$$P_{mac} = \frac{1}{L}\sum_{i=1}^{L} R \text{ of } i \tag{18}$$

$$F1_{mac} = \frac{1}{L}\sum_{i=1}^{L} 2 \times \frac{P_{mac} \times R_{mac}}{P_{mac} + R_{mac}} \tag{19}$$

$$P_{mic} = \frac{\sum_{i=1}^{L} tp \text{ of label } i}{\sum_{i=1}^{L} (tp \text{ of label } i + fp \text{ of label } i)} \tag{20}$$

$$R_{mic} = \frac{\sum_{i=1}^{L} tp \text{ of label } i}{\sum_{i=1}^{L} (tp \text{ of label } i + fn \text{ of label } i)} \tag{21}$$

$$F1_{mic} = 2 \times \frac{P_{mic} \times R_{mic}}{P_{mic} + R_{mic}} \tag{22}$$

$$P_{weighted} = \sum_{i=1}^{L} (P \text{ of } i \times \text{Weight of } i) \tag{23}$$

$$R_{weighted} = \sum_{i=1}^{L} (R \text{ of } i \times \text{Weight of } i) \tag{24}$$

$$F1_{weighted} = \sum_{i=1}^{L} (F1 \text{ of } i \times \text{Weight of } i) \tag{25}$$

4.3. Rank-based metrics

Rank-based metrics measure the ranking of the predicted labels on a classifier. It includes zero-one loss, coverage, ranking loss, and average precision [37–39]. The zero-one loss strictly calculates the fraction of misclassifications, if the subset is set to one. Otherwise, it calculates the number of misclassifications over samples. Ranking loss measures the average fraction of incorrectly ordered label pairs for an instance. Label average precision computes the fraction of truth labels that are assigned to each sample. The value of one indicates the best average precision score. Coverage estimates how far on average a model needs to go through in the ranked prediction to cover all true labels of a sample.

5. Results and discussion

We evaluate the proposed MCBiGRU model on Kaggle's toxic comment dataset. This large dataset is randomly split into 80:10:10 for training, validation, and testing. In particular, we used 181074 instances for training, 20120 instances for validation, and

Table 2
Hyperparameters.

Hyperparameters	Size	Hyperparameters	Size
Number of channel	5	Kernel sizes	1, 2, 3, 5, 6
Sequence length	150	Activation function	ReLU
Number of words	20000	Pooling size	4
Embedding dimension	100	Dropout	0.6
Word embedding	GloVe	Fully connected units	32
GRU units	200	loss	Binary crossentropy
Trainable	False	optimizer	Adam (0.003)
Filters	128	Epochs	100

22355 instances for testing. The proposed model is evaluated in multilabel environments using Google Colaboratory and Keras libraries. In our model settings, we converted the instances from upper case to lower case letters and removed the hyperlinks and punctuations. A tokenization method is employed to represent word sequences to integer sequences and then padded into a fixed-length vector. Then, the pre-trained GloVe word embedding method was applied to generate word vectors present in the training data. Moreover, we used one embedding layer, one 1D convolutional layer, one max-pooling layer, one gated recurrent unit layer, and two dropout layers for each channel. Similarly, we defined five channels with different kernel sizes such as 1, 2, 3, 4, and 5. These channels merged with one concatenate layer and then one fully connected layer and one normalization layer. Finally, one output layer with a sigmoid activation function.

A random approach method was used for fine-tuning the hyperparameters (Table 2). Specifically, we evaluated the model based on multilabel metrics such as sample-based metrics, rank-based metrics, and label-based metrics [37–39] with 100 epochs. The accuracy and loss curve for MCBiGRU with training, validation, and testing is shown in Fig.2. The confusion matrix for the proposed MCBiGRU model with training, validation, and testing is shown in Table 3. Table 4 shows the Precision, recall, and F1-score for each label. Table 5 shows the overall multilabel metrics results such as exact match, Hamming loss, zero-one loss, label ranking loss, ROC_AUC score, average precision score, and precision, recall, F1 score with micro, macro, and weighted. In particular, the proposed MCBiGRU model produces better mean training and validation accuracy with multichannel. The model achieves 98.8% and 99.1% for validation and testing, respectively. Also, the proposed model achieves a 71.7% F1-micro score and 98.2% ROC_AUC score.

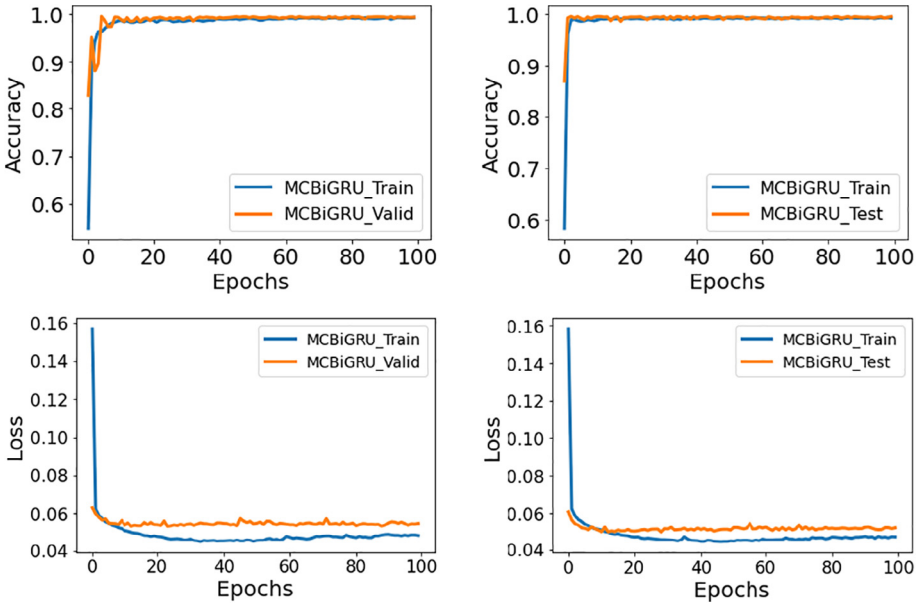

Fig. 2. The accuracy and loss curve for the MCBiGRU model with training, validation, and testing.

Table 3
Confusion matrix for MCBiGRU model with training, validation, and testing.

Label		Train		Valid		Train		Test	
		NT	T	NT	T	NT	T	NT	T
Toxic	NT	161730	2067	17830	316	161882	1915	19903	319
	T	4666	12611	603	1371	4550	12727	671	1462
Severe toxic	NT	179259	258	19881	38	179046	471	22082	69
	T	1136	421	153	48	945	612	125	79
Obscene	NT	169745	1577	18737	218	169944	1378	20907	225
	T	1785	7967	309	856	1808	7944	316	907
Threat	NT	180461	65	20041	3	180435	91	22283	7
	T	397	151	66	10	382	166	47	18
Insult	NT	170187	1775	18793	231	170168	1794	20958	301
	T	2273	6839	377	719	2109	7003	349	747
Identity hate	NT	178997	366	19883	45	179109	254	22093	48
	T	885	826	114	78	945	766	131	83

Table 4
Precision, recall, and F1-score for each label.

Class	Valid			Test		
	P	R	F1	P	R	F1
Toxic	0.813	0.695	0.749	0.821	0.685	0.747
Severe toxic	0.558	0.239	0.334	0.534	0.387	0.449
Obscene	0.797	0.735	0.765	0.801	0.742	0.770
Threat	0.769	0.132	0.225	0.720	0.277	0.400
Insult	0.757	0.656	0.703	0.713	0.682	0.697
Identity hate	0.634	0.406	0.495	0.634	0.388	0.481

Table 5
The model performance based on multilabel evaluation metrics.

Metrics	Valid	Test	Metrics	Valid	Test
Mean accuracy	0.988	0.991	Recall score macro	0.477	0.527
Exact match	0.915	0.916	F1 score macro	0.545	0.591
Hamming Loss	0.021	0.019	Precision score micro	0.784	0.773
Zero-one loss	0.085	0.084	Recall score micro	0.655	0.668
Label Ranking Loss	0.003	0.003	F1 score micro	0.712	0.717
ROC_AUC score	0.981	0.982	Precision score weighted	0.777	0.771
Average precision score	0.996	0.996	Recall score weighted	0.655	0.668
Precision score macro	0.721	0.704	F1 score weighted	0.706	0.713

Table 6
Result comparison with the existing models.

Authors	Data size	Model	MVA	MTA	F1	MACF1	ROC_AUC
Georgakopoulos et al. [7]	–	CNN_rand	91.2	–	–	–	
Elnaggar et al. [27]	159,571	Bi_RNN_CNN	–	–	59.0	–	
Khieu and Narwal [4]	159,571	LSTM	92.7	–	–	70.6	
Chu et al. [28]	159,571	CNN_char_emb	94.0	–	–	–	
Kohli et al. [20]	159,571	LSTM_Cus_emb	97.8	–	–	–	
Chakrabarty [3]	159,571	6 Head_ML	98.1	–	–	98.2	
Anand and Eswari [15]	159,571	GloVe_CNN	97.9	97.3	–	–	
Aken et al. [18]	223,549	Ensemble	–	–	79.1	–	98.3
Koratana and Hu [29]	223,549	Bi_LSTM_Attn_FastT	–	98.9	66.0	–	–
Lessmann [30]	223,549	GRU Models	–	–	75.8	–	97.3
Proposed	223,549	MCBiGRU_Valid	98.8	–	71.2	54.5	98.1
		MCBiGRU_Test	–	99.1	71.7	59.1	98.2

The result comparison of the existing deep learning model is shown in Table 6. In [7,27,4,27,20,3,15], the authors used 159,571 toxic comments in their research works. In [18,29,30], and our proposed work 223,549 comments have been used for experimental study. Specifically, the Aken et al. [18] has described in-depth error analysis in this large dataset with two input word embeddings such as character and n-gram word embeddings. However, we achieve better training and testing accuracy than the existing models using only n-gram word embeddings.

6. Conclusion

In this paper, we presented a multichannel convolutional bidirectional gated recurrent unit to categorize multilabel toxicities in online comments. Especially, the proposed model combines CNN and BiGRU in each channel to extract local features and long-term dependencies within comments using many filters and different kernel sizes. Our results show that the proposed MCBiGRU model outperforms the existing results. In the future, we intend to apply multichannel attention mechanisms in a distributed environment for multilabel toxic detection.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the University Grants Commission (UGC), Government of India for supporting this work under the National Doctoral Fellowship.

References

[1] D. Camacho, A. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, E. Cambria, The four dimensions of social network analysis: an overview of research methods, applications, and software tools, Inf. Fusion 63 (2020) 88–120.
[2] S. Ji, S. Pan, X. Li, E. Cambria, G. Long, Z. Huang, Suicidal ideation detection: a review of machine learning methods and applications, IEEE Trans. Comput. Social Syst. 8 (1) (2021) 214–226.
[3] N. Chakrabarty, A machine learning approach to comment toxicity classification, in: Computational Intelligence in Pattern Recognition, 2020, Springer, Singapore, pp. 183–193.

[4] K. Khieu, N. Narwal, Detecting and classifying toxic comments. Web: https://web. stanford. edu/class/archive/cs/cs224n/cs224n, 1184.
[5] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Sentic computing: exploitation of common sense for the development of emotion-sensitive systems. Development of Multimodal Interfaces: Active Listening and Synchrony, pp. 148–156.
[6] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Common sense computing: from the society of mind to digital intuition and beyond, Biometr. ID Manage. Multimodal Commun. (2009) 252–259.
[7] S.V. Georgakopoulos, S.K. Tasoulis, A.G. Vrahatis, V.P. Plagianakos, Convolutional neural networks for toxic comment classification, in: Proceedings of the 10th Hellenic Conference on Artificial Intelligence, 2018, pp. 1–6.
[8] M. Ibrahim, M. Torki, N. El-Makky, Imbalanced toxic comments classification using data augmentation and deep learning, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, IEEE, pp. 875–878.
[9] G Chen, D Ye, E Cambria, J Chen, Z Xing. Ensemble Application of Convolutional and Recurrent Neural Networks for Multi-Label Text Categorization. Proceedings of IJCNN, 2377-2383, 2017.
[10] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning based text classification: a comprehensive review, ACM Comput. Surv. 54 (2021).
[12] E. Cambria, Y. Li, F. Xing, S. Poria, K. Kwok, SenticNet 6: ensemble application of symbolic and subsymbolic AI for sentiment analysis, Proc. CIKM (2020) 105–114.
[13] M. Grassi, E. Cambria, A. Hussain, F. Piazza, Sentic web: a new paradigm for managing social media affective information, Cogn. Comput. 3 (3) (2011) 480–489.
[14] H. Hosseini, S. Kannan, B. Zhang, R. Poovendran, Deceiving google's perspective api built for detecting toxic comments, 2017. arXiv preprint arXiv:1702.08138.
[15] M. Anand, R. Eswari, Classification of abusive comments in social media using deep learning. in: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, IEEE, pp. 974–977.
[16] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deep learning for user comment moderation, 2017. arXiv preprint arXiv:1705.09993.
[17] F. Mohammad, Is preprocessing of text really worth your time for online comment classification?, 2018. arXiv preprint arXiv:1806.02908.
[18] B. van Aken, J. Risch, R. Krestel, A. Lser, Challenges for toxic comment classification: an in-depth error analysis, 2018. arXiv preprint arXiv:1809.07572.
[19] H.H. Saeed, K. Shahzad, F. Kamiran, Overlapping toxic sentiment classification using deep neural architectures, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, IEEE, pp. 1361–1366.
[20] M. Kohli, E. Kuehler, J. Palowitch, Paying attention to toxic comments online. Web: https://web. stanford. edu/class/archive/cs/cs224n/cs224n, 1184.
[21] F. Gargiulo, S. Silvestri, M. Ciampi, G. De Pietro, Deep neural network for hierarchical extreme multilabel text classification, Appl. Soft Comput. 79 (2019) 125–138.
[22] S.M. Rezaeinia, A. Ghodsi, R. Rahmani, Text classification based on multiple block convolutional highways, 2018. arXiv preprint arXiv:1807.09602.
[23] J. Yoon, H. Kim, multichannel lexicon integrated CNN-BiLSTM models for sentiment analysis, in: Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017), 2017, pp. 244–253.
[24] C. Quan, L. Hua, X. Sun, W. Bai, Multichannel convolutional neural network for biological relation extraction, BioMed Res. Int. 2016 (2017).
[25] H. Zhang, J. Wang, J. Zhang, X. Zhang, Ynu-hpcc at semeval 2017 task 4: using a multichannel cnn-lstm model for sentiment classification, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 796–801.
[26] W. Li, Y. Xu, G. Wang, Stance detection of microblog text based on two-channel CNN-GRU fusion network, IEEE Access 7 (2019) 145944–145952.
[27] A. Elnaggar, B. Waltl, I. Glaser, J. Landthaler, E. Scepankova, F. Matthes, Stop illegal comments: a multitask deep learning approach, in: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference, 2018, pp. 41–47.
[28] T. Chu, K. Jue, M. Wang, Comment abuse classification with deep learning, 2016. Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen.
[29] A. Koratana, K. Hu, Toxic Speech Detection.
[30] S. Lessmann, Antisocial Online Behavior Detection Using Deep Learning Elizaveta Zinovyeva Wolfgang Karl Hrdle.
[31] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[32] Y. Kim, Convolutional neural networks for sentence classification, 2014. arXiv preprint arXiv:1408.5882.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[34] F. Chollet, Keras, 2015.
[35] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. arXiv preprint arXiv:1412.3555.
[36] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, 2018. arXiv preprint arXiv:1811.03378.
[37] M.L. Zhang, Z.H. Zhou, A review on multilabel learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2013) 1819–1837.
[38] M.A. Tahir, J. Kittler, A. Bouridane, Multilabel classification using heterogeneous ensemble of multilabel classifiers, Pattern Recogn. Lett. 33 (5) (2012) 513–523.
[39] M.S. Sorower, A Literature Survey on Algorithms for Multilabel Learning, Oregon State University, Corvallis, 18, 2010, pp. 1–25.
[40] Jigsaw/Conversation AI. Wikipedia talk pages dataset.