Hatespeech and Offensive Content Detection in Hindi Language using C-BiGRU

Sudharsana Kannan[1], Jelena Mitrović[2]

Faculty of Computer Science and Mathematics, CAROLL Research Group, University of Passau, Germany

Abstract

In this paper, we present our submission from the team CAROLL_Passau for subtask 1A of the HASOC 2021 workshop. Our presented model, C-BiGRU, is composed of a Convolutional Neural Network (CNN) together with a bidirectional Recurrent Neural Network (RNN). We utilized word embeddings to allow our model to apprehend the correlation between words in the text. The structure of our model enables it to capture the contextual information along with the long-term dependencies in the text in order to perform binary classification on offensive text. We evaluated our model on the test data provided by the HASOC organizers. Our model achieved a macro F1 score of 75.04%, accuracy of 77.48%, precision and recall with the scores of 74.63% and 75.60% respectively.

Keywords

Hate speech, Offensive language, Hindi, C-BiGRU, Embeddings

1. Introduction

Social media is growing continuously to be one of the powerful means of communication around the world, spreading various forms of user-generated content containing various sorts of information[1, 2]. Although most of the time users of social media can post without any restrictions, the posts often require content moderation [3] since they can contain offensive language, abusive messages, or hate speech. The challenges present in automatically identifying and detecting offensiveness contained in these posts increased the attention of the scientific community [4]. Various studies are performed in the Natural Language Processing (NLP) community to ease the process of analyzing and identifying hate speech successfully in text and language used online [5, 6]. In an attempt to improve and develop new methodologies and approaches concerning hate speech and offensive language detection, various workshops, conferences, and competitions in the form of shared tasks are conducted in recent years [7, 8, 9]. In this paper, we propose a system to participate in a subtask to perform binary classification on a dataset provided by the organizers of the challenge containing hateful and non-hateful tweets in the Hindi language. We make the following contributions in our paper:

- Pre-processing strategy for extracting beneficial tweet content for detecting hate speech
- A model that is capable of detecting hate speech in multiple languages

Forum for Information Retrieval Evaluation, December 13-17, 2021, India

sudharsana.kannan@uni-passau.de (S. Kannan); jelena.mitrovic@uni-passau.de (J. Mitrović)

https://ca-roll.github.io/ (J. Mitrović)

0000-0001-7497-9248 (S. Kannan); 0000-0002-6634-0804 (J. Mitrović)

- Implementation details of the approach utilized to secure a position in the HASOC 2021 challenge

The remainder of the paper is organized as follows. In the second section, we present previous work related to hate speech detection. We then provide an overview of the baseline model. In the third section, the system description including the experimental setup of our model C-BiGRU is presented. In the fourth section, we report our results and discussion for the challenge. Lastly, we conclude the paper and provide changes and enhancements that can be applied to our approach in the future.

2. Background

A variety of ideas and algorithms have been proposed over the years to identify and categorize offensive language, aggression, hatespeech, and various abusive language phenomena on social media [10, 11, 12]. Addressing these topics in different languages using different features mainly, the text is predominant [6]. In this section, we look into different approaches and contributions related to our work in the field of hate speech detection.

A classification methodology through transfer learning using CNN is proposed by [13] utilizing the dataset of Hindi-English code switched language. The authors employed the method of transfer learning to train the tweets in English and then reused the system by retraining the model on the code-mixed dataset to detect hate speech successfully. [14] presented two different classifiers: SVM (Support Vector Machine) and Random forest classifier for hate speech detection using the code-mixed text of Hindi and English. The system used various textual features such as character, word, and lexicon-based features. The authors revealed that out of the two classifiers used, the SVM performed better in classifying the tweets. The approach used by [15] involved data augmentation by utilizing a translation strategy to increase the size of the dataset. They presented an ensemble of bidirectional GRU (BiGRU) and TF-IDF approaches using fastText embeddings as their best-performing model to detect aggressive tweets. [16] investigated two models namely the sub-word level LSTM model and hierarchical LSTM model with attention to detect hate speech from Hindi-English code-mixed social media text. A comparative study between aggressive and offensive language detection on three different languages: Hindi, Bangla, and English is presented in [17]. The authors used the SVM and BERT models to perform the study and revealed that the SVM classifier outperformed BERT in non-English datasets due to the lack of adequate pre-trained models for such languages. Other recent approaches [18, 19, 20] to detect hate speech utilize various machine learning and deep learning techniques.

3. System Overview

In this section, we describe our model C-BiGRU along with the pre-processing steps and embeddings utilized in the system. The architecture of the model is shown in Figure 1

3.1. Data

We utilized the data provided by the organizers of the HASOC 2021 workshop [21]. The dataset for subtask 1A for the Hindi language contains tweets annotated with the labels 'NOT' (Non Hate-Offensive) and 'HOF' (Hate and Offensive). The train data consists of 4,594 tweets and is split into 3,161, not offensive tweets and 1,433 offensive tweets. The test data consists of 1,532 tweets in total with a mix of 505 offensive tweets and 1027, not offensive tweets labeled in a similar fashion as that of the train data.

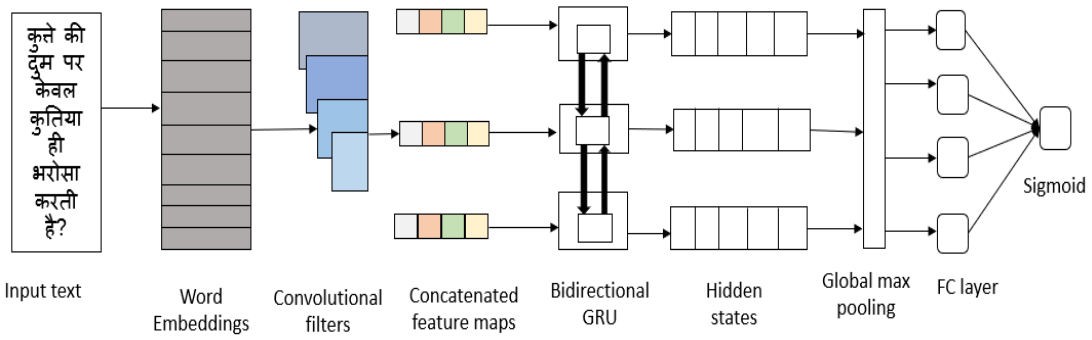

Figure 1: Architecture of C-BiGRU model as presented

3.2. Pre-processing

Pre-processing of tweets includes removal of emojis, URLs, and any additional spaces. In addition, the tweets are converted to lowercase, HTML character encodings are replaced with their respective token representation or literal. Apart from this, TweetTokenizer from NLTK is used to split the tokens containing special characters (e.g. '/', '-').

3.3. C-BiGRU Model

We begin our model construction with the embedding layer. The pre-processed tweets are fixed to a sequence length of 150 tokens, longer sequences are clipped at the end and shorter sequences are padded with a masking token. Following this, we create a dictionary containing all unique tokens that appear more than once and map them to their number of occurrences in the respective corpus. As a next step, we construct the weighting matrix W^{m*dim} to form the embedding layer, where dim is the dimension of the fastText [22] embedding model used in our setup and m the number of unique tokens t_i, $i \in \{1, .., m\}$. The word vector of t_i is stored in W if the token is present in the embedding model. If t_i has no pre-trained word vector, we generate a random vector drawn from the uniform distribution within $\left[-\sqrt{\frac{6}{dim}}, \sqrt{\frac{6}{dim}}\right]$ as suggested by [23].

Next in line is the convolutional layer where n-gram features are extracted from the sequence of tokens. The (kx128) 1-dimensional filters present in this layer assist in the accomplishment of feature extraction. The value of k varies from 2 to 5 and represents different window sizes. The outputs produced by this layer are all of the same sequence lengths which are attained through padding. Furthermore, we also make use of the ReLu activation function. The resulting feature maps are concatenated and then forwarded to the recurrent layer.
GRUs which are the improvised version of standard recurrent neural networks aim to solve the vanishing gradient problem using a gating mechanism. Originally proposed by [24] these are used in capturing long-term dependencies of input-sequence. GRUs have appeared to accomplish practically identical outcomes to LSTM in sequence modeling tasks. They have been reported to outperform LSTM while training on smaller datasets [25]. In our model, we made use of bidirectional GRU as the recurrent layer. As input to one of the GRU layers, the output from the previous layer is received meanwhile, the reversed form of the same output is used for the other GRU layer. The GRU layers return hidden states for each processed feature map. The hidden states from both layers are concatenated. The resultant output of size 150*128 is obtained by setting the hidden layers of both the layers to 64.
The output from the previous layer is then passed through a global max-pooling layer which reduces the output space to (1x128) nodes. Finally, a fully connected layer with 32 neurons that connect to a single output neuron that utilizes the sigmoid activation function is used. In order to prevent overfitting, a dropout layer is introduced with a rate of 0.2 before the single output neuron. Besides that, another dropout layer with a rate of 0.2 is included after the embedding layer.
Moreover, we used cross-entropy as an error function for our model and the Adam optimizer to update our network weights [26]. During the training phase, early stopping is implemented and we perform a split on the data resulting in 10% of the data as validation dataset and 90% of the data for training. The batch size of the gradient update is set to 32 with a maximum of 5 epochs.

3.4. Baseline

To compare the performance of our model with a baseline setup, we used a classification model, Logistic Regression. The model was evaluated using the validation dataset after carrying out the pre-processing steps as described previously in section 3.2. The baseline model after evaluating resulted in a macro F1 score of 70.03%, accuracy of 75.87%, precision and recall with the scores 64.04% and 51.05% respectively. Afterward, the system was tested using the test data and achieved a macro F1 score of 73.46%, accuracy of 78.13%, precision and recall with the scores 76.14% and 72.22% respectively. The scores of the baseline model are compared with our C-BiGRU system in the following section.

4. Results and Discussion

To study the performance of our C-BiGRU model, we conducted experiments in the Hindi language to detect hate speech utilizing the train and test data as described in section 3.1. The model was trained and thereafter, evaluated using the validation data which is 10% of the train

Table 1
Results for test data

System	F1	Recall	Precision	Accuracy
Baseline	73.46%	72.22%	76.14%	78.13%
C-BiGRU	75.04%	75.60%	74.63%	77.48%

data. During the validation phase, the system achieved an accuracy of 76.30%, recall, precision, and macro F1 score of 72.61%, 61.49%, and 63.64% respectively. Following this, the model was tested using the test data and the system achieved a macro F1 score of 75.04%, accuracy of 77.48%, precision and recall with the scores 74.63% and 75.60% respectively. The results are displayed in the table 1 Our presented C-BiGRU model successfully performed the classification task with noticeable results and surpassed the baseline model. The model also showed consistent F1 scores during the validation phase and test phase.

The overview of the results and findings of HASOC 2021 is presented in [27]
This experiment helped us evaluate the performance of the C-BiGRU model in another language as a follow-up to the previously impressive performances of the model in other languages - English, German, Danish, and Turkish [28, 29, 30].

5. Conclusion and Future work

In this submission, we described our approach to detecting hate speech present in social media posts and we presented the architecture of our model. We further provided the results of our model evaluated on the Hindi language with significant performance. Although our model performed well, there are a few limitations such as handling of unknown tokens, and distinguishing between explicit and implicit hate speech, which is an important task that is not easy to overcome, as explained and investigated by [31, 32].
In the future, we plan to extend our approach to work on identifying rhetorical figures and multi-word expressions containing abusive language. We will also work on improving the model to identify the fine-grained difference between implicit and explicit offensive posts. Apart from this, domain-specific word embeddings can help in handling the unknown tokens. Other potential features such as POS (Parts of Speech) tagging will also be explored.

Acknowledgments

The project on which this report is based was funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049. The author is responsible for the content of this publication.

SPONSORED BY THE

Federal Ministry
of Education
and Research

References

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: Proceedings of the 25th International Conference on World Wide Web, WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016, p. 145–153. URL: https://doi.org/10.1145/2872427.2883062. doi:10.1145/2872427.2883062.

[2] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion (2017). URL: http://dx.doi.org/10.1145/3041021.3054223. doi:10.1145/3041021.3054223.

[3] T. Gillespie, Content moderation, ai, and the question of scale, Big Data & Society 7 (2020) 2053951720943234. URL: https://doi.org/10.1177/2053951720943234. doi:10.1177/2053951720943234. arXiv:https://doi.org/10.1177/2053951720943234.

[4] M. Mozafari, R. Farahbakhsh, N. Crespi, Hate speech detection and racial bias mitigation in social media based on bert model, PLOS ONE 15 (2020) e0237861. URL: http://dx.doi.org/10.1371/journal.pone.0237861. doi:10.1371/journal.pone.0237861.

[5] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, S. Villata, A multilingual evaluation for online hate speech detection, ACM Trans. Internet Technol. 20 (2020). URL: https://doi.org/10.1145/3377323. doi:10.1145/3377323.

[6] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 1–10. URL: https://aclanthology.org/W17-1101. doi:10.18653/v1/W17-1101.

[7] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, c. Çöltekin, SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020), in: Proceedings of SemEval, 2020.

[8] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, FIRE 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 29–32. URL: https://doi.org/10.1145/3441501.3441517. doi:10.1145/3441501.3441517.

[9] B. Cristina, D. Felice, F. Poletto, S. Manuela, T. Maurizio, Overview of the EVALITA Hate

Speech Detection (HaSpeeDe) Task, in: T. Caselli, N. Novielli, V. Patti, P. Rosso (Eds.), Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), CEUR.org, Turin, Italy, 2018.

[10] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: ICWSM, 2017.

[11] M. Mozafari, R. Farahbakhsh, N. Crespi, A bert-based transfer learning approach for hate speech detection in online social media, in: H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, L. M. Rocha (Eds.), Complex Networks and Their Applications VIII, Springer International Publishing, Cham, 2020, pp. 928–940.

[12] J. H. Park, P. Fung, One-step and two-step classification for abusive language detection on Twitter, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 41–45. URL: https://aclanthology.org/W17-3006. doi:10.18653/v1/W17-3006.

[13] P. Mathur, R. Shah, R. Sawhney, D. Mahata, Detecting offensive tweets in Hindi-English code-switched language, in: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 18–26. URL: https://aclanthology.org/W18-3504. doi:10.18653/v1/W18-3504.

[14] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, M. Shrivastava, A dataset of Hindi-English code-mixed social media text for hate speech detection, in: Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, Association for Computational Linguistics, New Orleans, Louisiana, USA, 2018, pp. 36–41. URL: https://aclanthology.org/W18-1105. doi:10.18653/v1/W18-1105.

[15] J. Risch, R. Krestel, Aggression identification using deep learning and data augmentation, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 150–158.

[16] T. Y. Santosh, K. V. Aravind, Hate speech detection in hindi-english code-mixed social media text, in: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, CoDS-COMAD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 310–313. URL: https://doi.org/10.1145/3297001.3297048. doi:10.1145/3297001.3297048.

[17] R. Kumar, B. Lahiri, A. K. Ojha, Aggressive and offensive language identification in hindi, bangla, and english: A comparative study, SN Computer Science 2 (2021) 26. URL: https://doi.org/10.1007/s42979-020-00414-6. doi:10.1007/s42979-020-00414-6.

[18] A. Baruah, K. A. Das, F. A. Barbhuiya, K. Dey, Iiitg-adbu@hasoc-dravidian-codemix-fire2020: Offensive content detection in code-mixed dravidian text, CoRR abs/2107.14336 (2021). URL: https://arxiv.org/abs/2107.14336. arXiv:2107.14336.

[19] P. K. Roy, A. K. Tripathy, T. K. Das, X.-Z. Gao, A framework for hate speech detection using deep convolutional neural network, IEEE Access 8 (2020) 204951–204962. doi:10.1109/ACCESS.2020.3037073.

[20] R. Raja, S. Srivastavab, S. Saumyac, Nsit & iiitdwd@ hasoc 2020: Deep learning model for hate-speech identification in indo-european languages (2021).

[21] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranas-

inghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: http://ceur-ws.org/.

[22] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 427–431. URL: https://aclanthology.org/E17-2068.

[23] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification., in: In Proceedings of the IEEE international conference on computer vision, pages 1026–1034., 2015.

[24] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734. URL: https://aclanthology.org/D14-1179. doi:10.3115/v1/D14-1179.

[25] J. Chung, Çaglar Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, ArXiv abs/1412.3555 (2014).

[26] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[27] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.

[28] J. Mitrović, B. Birkeneder, M. Granitzer, nlpUP at SemEval-2019 task 6: A deep neural language model for offensive language detection, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 722–726. URL: https://aclanthology.org/S19-2127. doi:10.18653/v1/S19-2127.

[29] B. Birkeneder, J. Mitrovic, J. Niemeier, L. Teubert, S. Handschuh, upinf-offensive language detection in german tweets, in: 14th Conference on Natural Language Processing KONVENS 2018, 2018.

[30] O. Hussein, H. Sfar, J. Mitrović, M. Granitzer, NLP_Passau at SemEval-2020 task 12: Multilingual neural network for offensive language detection in English, Danish and Turkish, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 2090–2097. URL: https://aclanthology.org/2020.semeval-1.277.

[31] T. Caselli, V. Basile, J. Mitrović, I. Kartoziya, M. Granitzer, I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language, in: Proceedings of the 12th language resources and evaluation conference, 2020, pp. 6193–6202.

[32] T. Caselli, V. Basile, J. Mitrović, M. Granitzer, Hatebert: Retraining bert for abusive language detection in english, arXiv preprint arXiv:2010.12472 (2020).