AKADÉMIAI KIADÓ

Pollack Periodica • An International Journal for Engineering and Information Sciences

17 (2022) 3, 13–18

DOI: 10.1556/606.2022.00608

ORIGINAL RESEARCH PAPER

Identification of online harassment using ensemble fine-tuned pre-trained Bert

Aadil Gani Ganie* and Samad Dadvandipour

Department of Information Engineering, Faculty of Mechanical Engineering and Information, University of Miskolc, Miskolc-Egyetemváros, Hungary

Received: January 25, 2022 • Revised manuscript received: April 11, 2022 • Accepted: April 16, 2022
Published online: June 29, 2022

ABSTRACT

Identification of online hate is the prime concern for natural language processing researchers; social media has augmented this menace by providing a virtual platform for online harassment. This study identifies online harassment using the trolling aggression and cyber-bullying dataset from shared tasks workshop. This work concentrates on extreme pre-processing and ensemble approach for model building; this study also considers the existing algorithms like the random forest, logistic regression, multinomial Naïve Bayes. Logistic regression proves to be more efficient with the highest accuracy of 57.91%. Ensemble bidirectional encoder representation from transformers showed promising results with 62% precision, which is better than most existing models.

KEYWORDS

bidirectional encoder representation from transformers, online hate, cyber-bullying, natural language processing, machine learning

1. INTRODUCTION

With the increasing parameter of social media usage among all age groups, its erroneous use has led to online harassment. Cyber or Internet bullying is bullying through digital media, mainly social media. According to UNICEF, cyber-bullying has repetitive behavior to scare those targeted, anger, or shame. Examples include spreading lies about someone, sending hurtful messages or threats on social media through messages, impersonating someone, and sending mean messages on their behalf [1]. Social media provides us with a space to discuss various topics related to day-to-day life. There may be narratives and counter-narratives, which is generally regarded as suitable for dissent and discussion; however, some cyber abusers take this opportunity to abuse and shame someone. With several languages, users utilize while interacting online, the cyber world remains global. In linguistically diverse countries like India, Indonesia, etc., the gap between users using their native language and English speakers are noteworthy. Social media giants like Facebook and Twitter took several steps to mitigate or eradicate cyber abuse, but it still exists. This study has been carried out to identify online harassment in multilingual text. Significant work has been done to determine cyber harassment in an automated way using traditional supervised machine learning methods like Support Vector Machine (SVM), Long Short Term Memory (LSTM), logistic regression, decision trees, etc., [2–4]. Though, most of the work has been prepared in the English language. This study used fine-tuned uncased-Bidirectional Encoder Representation from Transformers (BERT) architecture for identifying online harassment in a multilingual dataset. Authors in [5] tried to detect the cyber abuse in multilingual data, but they used simple transformer architecture without fine-tuning and significant preprocessing of the textual data. This study focuses on the famous ensemble approach to attain more accuracy. In preprocessing, lemmatization, stop-word removal, Parts of Speech (PoS) tagging have been evaluated to feed the most accurate data to the model. Before using the pre-trained BERT

*Corresponding author.
E-mail: aadilganiganie@gmail.com

network, the data was provided into various traditional classifiers like SVM, multinomial Niave Bayes, Logistic regression, etc. Almost all the classifiers achieved the same accuracy. This study used TRolling Aggression and Cyber-bullying (TRAC)-1 dataset and showed accuracy close to state-of-art results and more than the baseline without much fine-tuning.

2. MATERIALS AND METHODS

Online harassment can take any form, but predominantly it is rooted in social media. The latest survey by pew research center [6] finds that 75% of the targets of online abuse equaling 31% of Americans overall say their most recent experience of online hate was on social media. Questions have been upraised on the working of social media giants for the elimination or mitigation of online harassment; about 79% say social media companies are not doing a fair job at addressing online harassment bullying on their platforms. Some of the key findings of an online survey conducted by the American trends panel [7] are that 41% of American adults have experienced online hate, and 25% have experienced grave harassment. The above disturbing trends have forced the researchers to automate the detection and subsequent eradication of online harassment, which eventually gives rise to online hate detection using Natural Language Processing (NLP). It is pretty challenging and perplexing to institutionalize the idea of abuse. Mishra [8] used it to discuss racism and sexism, while Nobata [9] referred to hate speech, profanity, and derogatory language. The first reported method for abuse detection was that of Spertus [10] in 1997, who hand-crafted rules over text to generate feature vectors for learning. Dadvar [11] uses a social feature engineering technique that incorporates features and identity traits of a user to the model likelihood of abusive behavior called user profiling Dadvar [11] includes the user's age alongside other lexicon-based features to detect cyber-bullying. In [12], authors used the gender of Twitter users with character n-gram for detection of sexism and racism in tweets F1-score improved from an existing 73.89%–73.93%. Authors [13] were the first to use the deep learning model for online harassment detection. They improved the accuracy of their model from existing 78.89%–80.07%, which outperforms the existing traditional methods significantly. In [4], used LSTM model with GloVe for feature engineering to detect online abuse, they achieved the best (weighted F1 of 93%) results by randomly initializing embeddings. Park and Fung [14] categorize the comments collected by combining two datasets, they concluded that combining the two-granularities using two input channels improves accuracy other researchers like [15–17] acknowledge the same. In GermEval shared task [18], authors made the winning submission with an F1-score of 76.95% and 53.59% for sub-task 1 and sub-task 2. Researchers in [19] have shown that learning about the classification of emotions and detecting abuse leads to improved performance.

3. DATASET

For this study data has been collected from the dataset - the shared task on aggression identification organized at the trolling, aggression, and cyber-bullying workshop [20]. Training data consists of 10,799 randomly selected Facebook comments; these comments have been annotated into three categories Overly AGgressive (OAG), COovertly Aggressive (COA), and Non-AGgressive (NAG). Test data or validation data is 1200 samples.

4. RESULTS AND DISCUSSION

Identification of observations as abusive gives the victims of abuse validation and allows observers to understand the extent of the problem. This study tried to identify online harassment using pre-trained BERT with an ensemble approach on the TRAC-1 dataset. The most recent study by [5] has used simple BERT architecture without considering the importance of preprocessing steps like handling of Not a Number (NaN) values, stopword removal, PoS tagging, contractions, stemming and lemmatization, which suggests that probably their model was not trained on good data, which may have led to model over-fitting[21–24]. The researchers also did not consider the fine-tuning strategies [25–26], which supplement the model to achieve better results. In this study, all the steps mentioned above were performed and try to identify the abuse in the multilingual text as it is shown in Table 1. This experiment has been

Table 1. Data preprocessing

Id	Facebook_corpus_msr_466073
Text	Most Private Banks ATM's Like HDFC, ICICI etc., are out of cash. Only Public sector bank's ATMs working
Label	NAG
Clean	Most of private banks atm like hdfc, icici etc. are out of cash. only public sector bank atm working
No contractions	['most', 'of', 'private', 'banks', 'atm', 'like', 'hdfc,', 'icici', 'etc', 'are', 'out', 'of', 'cash.', 'only', 'public', 'sector', 'bank', 'atm', 'working']
Stopwords	['private', 'banks', 'atm', 'like', 'hdfc', 'icici', 'etc', 'cash', 'public', 'sector', 'bank', 'atm', 'working']
PoSTag	[('private', 'JJ'), ('banks', 'NNS'), ('atm', 'VBP'), ('like', 'IN'), ('hdfc', 'NN'), ('icici', 'NN'), ('etc', 'FW'), ('cash', 'NN'), ('public', 'NN'), ('sector', 'NN'), ('bank', 'NN'), ('atm', 'IN'), ('working', 'VBG')]
Lemmatized	most of private banks atm like hdfc icici etc are out of cash only public sector bank atm working

conducted on Graphic Processing Unit (GPU) Tesla T4, core i5, 12 GB RAM.

After preprocessing, the data has been fed into various famous existing algorithms like SVM, Naïve Bayes, logistic regression, random forest, etc., due to the shallow nature of the network, but the results obtained were not satisfactory. The accuracy achieved is not a milestone, but it is more than the baseline, which is 35.53% as it is shown in Fig. 1. Due to poor performance by the above algorithms deep learning approach has been introduced, the data has been fed into the pre-trained BERT with a multi-head attention model. It works on the mechanism of multi-head attention with a masked language model. BERT is a language representation pre-training method used to create models that are then freely downloaded and utilized by NLP practitioners. There are two ways to approach the problem either use the existing models to extract language features of high quality from text data, or fine-tune them to produce state-of-the-art predictions for a particular task (classification, identification of entities, answering question, etc.).

Three main advantages of BERT are quicker development, fewer data and better results. Fine-tuning the model played an important role in increasing the network's performance. BERT sequence classifier from transformers has been used for classification. BERT comes in two variants base model and large model. The size of training data is only

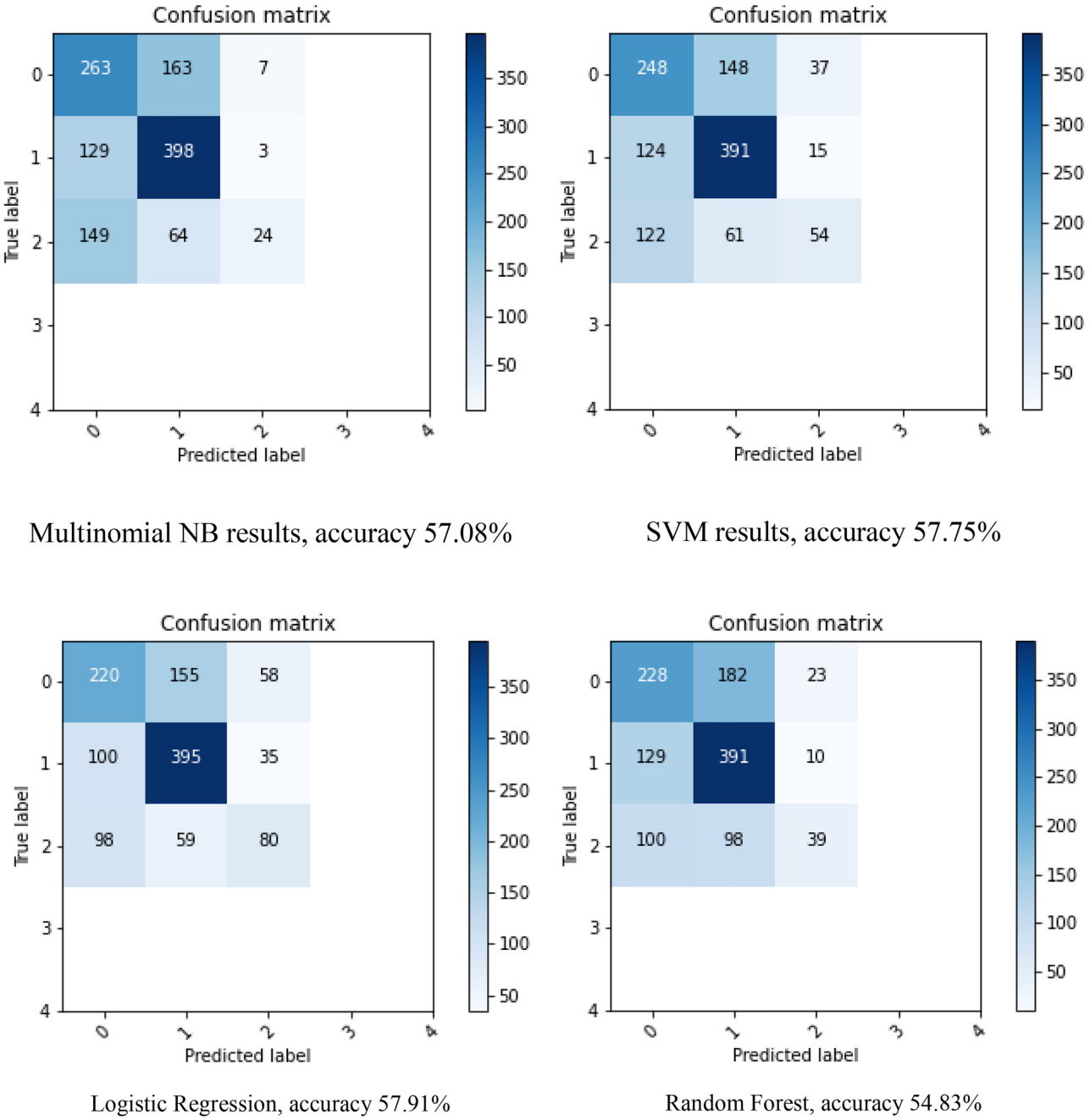

Fig. 1. Result of machine learning algorithms

difference between the two variants. This study used a *bert-based-uncased* model with several labels 3. Results of various fine-tuning parameters are listed below. BERT consists of the encoder and decoder parts. The first encoder layer receives a concatenation of WordPiece embeddings and positional embeddings produced from the input sequence as its input representation. The conversion of a query and a group of key-value pairs to output can be characterized as an attention function, where the question, keys, values, and production are all vectors. The result is a weighted sum of the values, with the weight allocated to each value determined by the query's compatibility function with the relevant key. a Query, Key, and Value vector for each input embedding token are built by multiplying the embedding by three learned matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$, respectively, given an embedded column vector $\mathbf{x}$ for an input sequence. The Query, Key, and Value vectors are stacked into column vectors $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ for concurrent computing. The self-attention function is therefore provided by:

$$\begin{aligned}\text{Attention}(\mathbf{x}) &= \text{Attention}(\mathbf{Q},\ \mathbf{K},\ \mathbf{V})\\ &= \text{softmax}\left\{\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}\right\},\end{aligned} \tag{1}$$

where d_k is the dimension of queries and keys. The transformer performs self-attention function in parallel with multiple attention heads by projecting the queries, keys and values h times with different, learned linear projections to d_k; d_k and d_v dimensions, respectively. Attention function is performed in parallel on each of these projected versions of queries, keys and values, resulting d_v -dimensional output column vector values,

$$\begin{aligned}\mathbf{MultiHead}(\mathbf{x}) &= \mathbf{MultiHead}(\mathbf{Q},\ \mathbf{K},\ \mathbf{V})\\ &= \mathbf{Concat}(head_1,\ \ldots,\ head_h)\mathbf{W}^o,\end{aligned} \tag{2}$$

this operation $\mathbf{Concat}(head_1,\ \ldots,\ head_h)\mathbf{W}^o$ results in row vector, because $\mathbf{W}$ is a matrix. Concat is a row vector, so the result is a row vector, it means that $\mathbf{MultiHead}(\mathbf{x})$ is a row vector and here $\mathbf{head_i} = \mathbf{Attention}(\mathbf{W}_i^Q\mathbf{Q}, \mathbf{W}_i^K\mathbf{K}, \mathbf{W}_i^V\mathbf{V})$. **Concat** is the concatenation function; the projections are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{maddel}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{madd}} \times i \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{madd}} \times d_v}$ and $\mathbf{W}^O \in \mathbb{R}^{hd_y \times d_{\text{model}}}$ with $d_{\text{model}} = d_k h$.

Each transformer layer consists of two sub-layers. The first sub-layer is the multi-head attention and its normalized output is fed to the second sub-layer of fully connected feed forward network. The activation function for the feed forward networks is ReLU. Formally, the hidden states of transformer with M number of transformer layers are calculated as follows:

$$Tr_m(\mathbf{x}) = \text{norm}(Att(\mathbf{x}) + FFN(Att(\mathbf{x}))), \tag{3}$$

where

$$\begin{aligned}Att(\mathbf{x}) &= \text{norm}(\mathbf{x}^{\boldsymbol{T}} + \mathbf{MultiHead}(\mathbf{x})),\\ FFN(\mathbf{x}) &= m\ (0, \mathbf{x}^{\boldsymbol{T}}\mathbf{W}_1 + \boldsymbol{b}_1)\mathbf{W}_2 + \mathbf{b}_2,\end{aligned}. \tag{4}$$

where norm is the normalization function with linear connection followed by fully connected feed forward network, $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weights of the first and second fully connected networks with $\mathbf{b}_1$, $\mathbf{b}_2$ as bias values, and $m \in M$.

BERT creates a corrupted version $\widehat{\mathbf{X}}$ by randomly assigning a special symbol [MASK] to 15% of the tokens in $\overline{\mathbf{x}}$. If the masked tokens are denoted as $\mathbf{x}$, the training goal is to reconstruct $\overline{\mathbf{x}}$ from $\widehat{\mathbf{X}}$,

$$\begin{aligned}\max_{\theta} \log \mathrm{p}_\theta\left(\overline{\mathbf{x}} \mid \widehat{\mathbf{x}}\right) &\approx \sum\nolimits_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{m}_{\mathrm{t}} \log \mathrm{p}_\theta\left(\mathbf{x}_{\mathrm{t}} \mid \widehat{\boldsymbol{x}}\right)\\ &= \sum\nolimits_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{m}_{\mathrm{t}} \log \frac{\exp\left(\mathbf{H}_\theta(\widehat{\boldsymbol{x}})_t^{\top} e(\boldsymbol{x}_t)\right)}{\sum_{x'} \exp\left(\mathbf{H}_\theta(\widehat{\boldsymbol{x}})_t^{\top} e(\boldsymbol{x}')\right)},\end{aligned} \tag{5}$$

where $m_t = 1$ denotes token $\mathbf{x}_t$ is masked, $e(\mathbf{x})$ indicates the embedding of $\mathbf{x}$ and $\mathbf{H}_\theta$ is a transformer that transforms a length-T text sequence $\mathbf{x}$ into a series of hidden vectors $\mathbf{H}_\theta(\mathbf{x}) = [\mathbf{H}_\theta(\mathbf{x})_1,\ \mathbf{H}_\theta(\mathbf{x})_2,\ \ldots,\ \mathbf{H}_\theta(\mathbf{x})_T]$. The results with fine-tuning are shown in Tables 2-5, the comparison between the existed methods and this approach is shown in Fig. 2.

Table 2. Learning = $2{\cdot}10^{-5}$, batch size =32

Epoch	Training loss	Validation Loss	Validation Accuracy	Training Time	Validation Time
1	0.90	0.83	0.59	0:02:04	0:00:04
2	0.64	0.85	0.6	0:02:00	0:00:05
3	0.31	1.11	0.6	0:02:01	0:00:05
4	0.12	1.31	0.6	0:02:02	0:00:05

Table 3. Learning rate = $5{\cdot}10^{-5}$, batch size = 64

Epoch	Training loss	Validation Loss	Validation Accuracy	Training Time	Validation Time
1	0.91	0.85	0.60	0:01:31	0:00:08
2	0.61	0.95	0.60	0:01:36	0:00:09
3	0.23	1.31	0.60	0:01:38	0:00:09
4	0.07	1.60	0.60	0:01:39	0:00:09

Table 4. Learning rate = $2{\cdot}10^{-5}$, batch size = 64

Epoch	Training loss	Validation Loss	Validation Accuracy	Training Time	Validation Time
1	0.92	0.85	0.59	0:01:43	0:00:04
2	0.70	0.86	0.60	0:01:48	0:00:04
3	0.45	0.97	0.60	0:01:50	0:00:04
4	0.26	1.07	0.60	0:01:51	0:00:04

Table 5. Learning rate = $5{\cdot}10^{-5}$, batch size = 16

Epoch	Training loss	Validation Loss	Validation Accuracy	Training Time	Validation Time
1	0.90	0.86	0.59	0:02:13	0:00:05
2	0.61	0.94	0.60	0:02:19	0:00:05
3	0.27	1.34	0.61	0:02:21	0:00:05
4	0.11	1.85	0.62	0:02:21	0:00:05

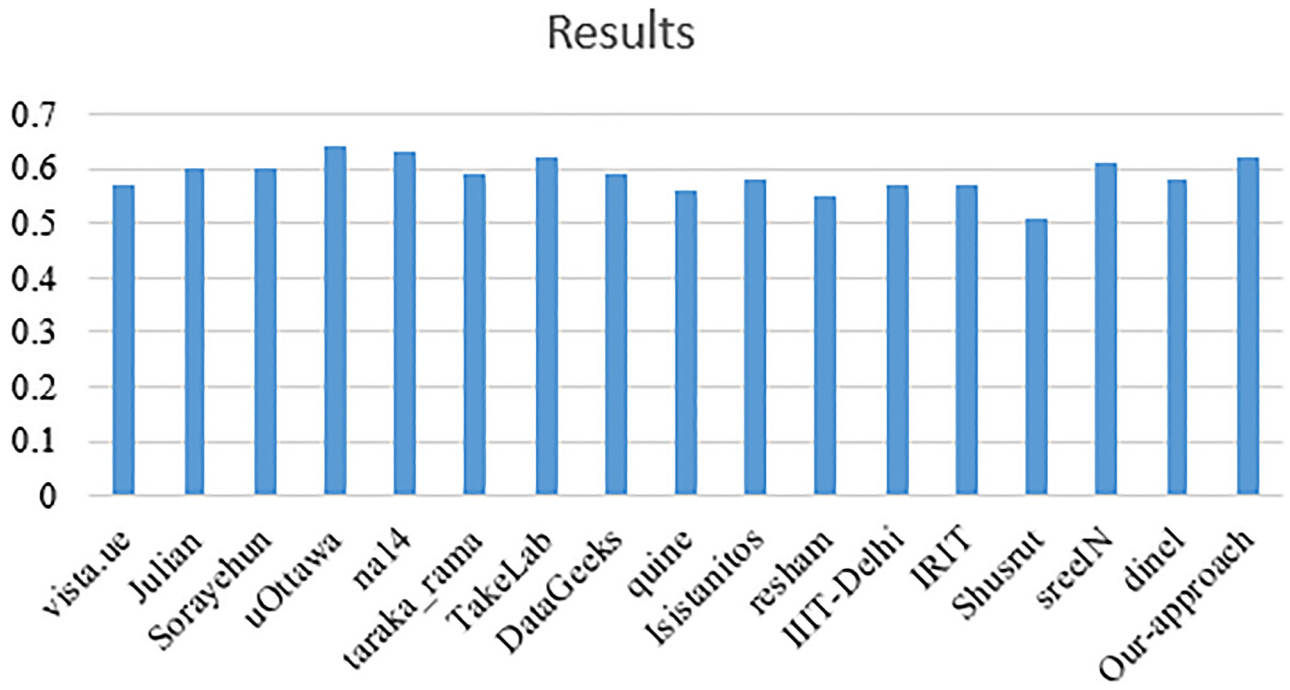

Fig. 2. Result comparison on test data

5. CONCLUSION

The research has been carried out to identify the online harassment on digital media using a famous dataset from the shared task of identifying trolling, aggression, and cyber-bullying workshop (TRAC-1), unlike existing studies, which fed the semi preprocessed data to the model. This study preprocessed the data significantly by applying the techniques like contraction handling, stemming, lemmatization, stop-word removal, etc. The preprocessed data has been fed to the existing algorithm like Naïve Bayes logistic regression, but the accuracy achieved is not par. This work achieved competitive accuracy compared to state-of-the-art models by using fine-tuning strategies for pre-trained BERT with an ensemble approach. However, it can be concluded that with the increase in batch size and learning rate, the accuracy deteriorates, and the model starts to over-fit.

ACKNOWLEDGEMENTS

Author is highly thankful to my supervisor and my faculty for their continuous support.

REFERENCES

[1] Cyberbullying: What is it and how to stop it. [Online]. Available: https://www.unicef.org/end-violence/how-to-stop-cyberbullying. Accessed Apr. 25, 2021.

[2] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, pp. 11–17, 2011.

[3] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *10th International Conference on Machine Learning and Applications and Workshops*, Honolulu, HI, USA, Dec. 18–21, 2011, vol. 2, pp. 241–244.

[4] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, Austin, USA, April 5–6, 2017, pp. 759–760.

[5] A. Malte and P. Ratadiya, "Multilingual cyber abuse detection using advanced transformer architecture," in *TENCON 2019-2019 IEEE Region 10 Conference*, Kochi, India, Oct. 17–20, 2019, pp. 784–789.

[6] The state of online harassment. [Online]. Available: https://www.pewresearch.org/internet/2021/01/13/the-state-of-online-harassment/. Accessed: Apr. 27, 2021.

[7] The American Trends Panel. [Online]. Available: https://www.pewresearch.org/methods/u-s-survey-research/american-trends-panel/. Accessed: Apr. 27, 2021.

[8] P. Mishra, M. D. Tredici, H. Yannakoudakis, and E. Shutova, "Author profiling for abuse detection," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, Aug. 31, 2018, pp. 1088–1098.

[9] D. Samad and G. A. Gani, "Analyzing and predicting spear-phishing using machine learning methods," *Multidiszciplináris Tudományok*, vol. 10, no. 4, pp. 262–273, 2020.

[10] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th International Conference on World Wide Web*, Montréal, Québec, Canada, Ap. 11–15, 2016, pp. 145–153.

[11] E. Spertus, "Smokey: Automatic recognition of hostile messages," in *Proceedings of IAAI-97, The 9th Conference on Innovative*

Application of Artificial Intelligence, Providence Rhode Island, Jul. 27, 1997, pp. 1058–1065.

[12] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in *European Conference on Information Retrieval*, Moscow, Russia, Mar. 24–27, 2013, pp. 693–696.

[13] A. G. Ganie, "Private network optimization," *Multidiszciplináris Tudományok*, vol. 11, no. 4, pp. 248–254, 2021.

[14] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL Student Research Workshop*, San Diego, California, June 17, 2016, pp. 88–93.

[15] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embedding," in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, May 18–22, 2015, pp. 29–30.

[16] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," in *Proceedings of the First Workshop on Abusive Language Online*, Vancouver, Canada, Jul. 30–Aug. 4, 2017, pp. 41–45.

[17] V. Singh, A. Varshney, S. S. Akhtar, D. Vijay, and M. Shrivastava, "Aggression detection on social media text using deep neural networks," in *Proceedings of the 2nd Workshop on Abusive Language Online*, Brussels, Belgium, Oct. 31, 2018, pp. 43–50.

[18] C. Wang, "Interpreting neural network hate speech classifiers," in *Proceedings of the 2nd Workshop on Abusive Language Online*, Brussels, Belgium, Oct. 31, 2018, pp. 86–92.

[19] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *European Semantic Web Conference*, Crete, Greece, Jun. 3–7, 2018, pp. 745–760.

[20] A. G. Ganie and S. Dadvandipour, "Sentiment analysis on the effect of trending source less News: special reference to the recent death of an Indian actor," in *International Conference on Artificial Intelligence and Sustainable Computing*, Greater Noida, India, Mar. 22–23, 2021, pp. 3–16.

[21] A. Paraschiv and D. C. Cercel, "UPB at GermEval-2019 Task 2: BERT-Based Offensive Language Classification of German Tweets," in *Proceedings of the 15th Conference on Natural Language Processing*, Erlangen, Germany, Oct. 8, 2019, pp. 395–402.

[22] S. Rajamanickam, P. Mishra, H. Yannakoudakis, and E. Shutova, "Joint modeling of emotion and abusive language detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Jul. 5–10, 2020, pp. 4270–4279.

[23] A. G. Ganie, S. Dadvandipour, and M. A. Lone, "Detection of semantic obsessive text in multimedia using machine and deep learning techniques and algorithms," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 11, pp. 2567–2577, 2021.

[24] S. Modha, P. Majumder, and T. Mandl, "Filtering aggression from the multilingual social media feed," in *Proceedings of the First Workshop on Trolling, Aggression, and Cyberbullying*, Santa Fe, New Mexico, USA, Aug. 18, 2018, pp. 199–207.

[25] M. H. Al-Hafadhi and G. Krallics, "Prediction and numerical simulation of residual stress in multi-pass pipe welds," *Pollack Period.*, vol. 16, no. 2, pp. 7–12, 2021.

[26] L. Kota and K. Jármai, "Improving optimization using adaptive algorithms," *Pollack Period.*, vol. 16, no. 1, pp. 14–18, 2021.