International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Detection of Hate Speech in Marathi Using Language Specific Pre-Processing

Ankur Sarode[1] and Nailya Sultanova[2]
[1,] WB Tech, ING Bank, Netherland
[2,] Kazan Federal University, Kazan, Russia
[1]ankur.sarode@gmail.com
[2]NRSultanova@kpfu.ru

Corresponding author email: Ankur.sarode@gmail.com

***Abstract*— The increasing accessibility of social media platforms has exponentially increased the amount of textual content on the internet. Among this textual content, there is also a rapid growth of hate speech on these social media platforms. This offensive and hateful speech are increasingly becoming a cause for self-harm, depression and even suicidal tendencies, motivating social media platforms to invest in strategies that can make social communities safer. Most of the research done on hate speech detection is done in languages like English, Spanish, French and more. There is also a good amount of work done on the Hindi language, but there is not much significant work done on the Marathi language apart from a few notable exceptions. Marathi has 83 million speakers many of whom consume Marathi social media content. This makes hate speech detection in the Marathi language a desirable opportunity to explore. How various pre-processing methods affect hate speech detection in Marathi were explored in this research. Furthermore, transfer learning was used to analyse the efficiency of multilingual hate speech detection models for the Marathi language. Finally, how hate speech detection in large texts can vary from that in shot text data was explored.**

***Keywords*— Marathi language; pre-processing; media; speech detection.**

1. Introduction

Marathi is a regional language that is spoken in western India by over 83 million people (Marathi language - Wikipedia, 2022). There has been a rapid growth of Marathi speech and text on the internet, especially on social media platforms. Along with this growth in the use of the Marathi language on social media, the use of hate speech in Marathi has also increased exponentially. Users are freely expressing their opinions about any and everything without thinking of any repercussions. The victims of hate speech on social media often develop emotional and mental problems like anxiety, depression, suicidal thoughts etc [1]. There is a lot of effort being put to curb hate speech and filter out hateful comments but these are limited to a few high-resource languages like English or Spanish. For languages like Marathi where there is not much research done, there is no proper censoring mechanism for hateful text. Due to this absence, social media is filled with profane and hateful texts in Marathi. Even if these texts are reported, there is no way to automatically verify their profanity and must undergo manual verification. To enable social media platforms to incorporate hate speech detection and filtering mechanisms, it is crucial to explore this topic and develop realistic solutions.

The research done today on hate speech detection for high-resource languages is abundant, with a variety of techniques being proposed from classical machine learning classifiers like SVM, and Random Forests to Deep Learning models which use BiLSTMs with rigorous text preprocessing [2][3]. The latest research introduces pre-trained transformer models called BERT, which has taken over the hate speech detection domain by storm [4]. Following this, a lot of research has been done on BERT models to enhance them. There is also research done using other techniques that can outperform BERT in some natural language processing tasks. One such model is the XLNet [5].

Due to the easy availability of datasets, major work being done on hate speech detection is in the English language. But recently, with the rapid growth of social media platforms, a lot of text content is generated in other languages as well which has triggered research in these languages. Work is being done in Arabic, French, Greek, Dutch, Japanese and more languages [6]; [7]; [8]; [9]; [10].

In terms of Indian native languages, recently there has been some work being done for hate speech detection in the Hindi Language. Techniques for emotion detection and classification of tweets in Hindi have been explored [11]; [12].

Some research is also done on Indian languages like Bengali, Tamil, Telugu, and Malayalam, where the major challenge is to acquire a dataset [13]; [14]. Furthermore, recent work has explored building models that can identify hate speech in multiple languages, such models are called multilingual models [15]; [16].

Moving to hate speech detection in the Marathi language, work has been done to create datasets with a significant amount of annotated text [17]. Moreover, work has been done to gather datasets that are divided into 4 classes of different types of hate speech and train a model using transfer learning on this data [18]. Some research is done comparing monolingual models with multilingual models for hate speech detection in Marathi [19]. The major work done for the Marathi language revolves around gathering a dataset and applying transfer learning to use existing models and validate how well they perform using simple accuracy metrics. Other metrics such as F1-score and AUC have not been used to

evaluate these models. A set of methodologies and techniques that can improve the quality of hate speech classification were proposed. Hate speech classification into two categories – Hate & Offensive (HOT) or Non–Hate nor Offensive (NOT) was targeted.

Hate & Offensive (HOT) – texts that contain hateful, profane, or negative sentiments.

Non – Hate nor Offensive (NOT) – texts that do not contain hateful, negative sentiments.

A lot of work has been done to categorize hate speech in English. In this study, we have attempted to extend this work to the Marathi language.

2. Materials and Methods

2.1. Dataset

Two datasets were used for this research:

1. Short Text Dataset: Each entry in this dataset was a single sentence that has a size of 1 to 25 words.

The combined dataset had a total of 41,100 annotated text samples which were divided into train and test samples in a 70:30 ratio while maintaining class balance.

2. Long Text Dataset: Each entry in this dataset was a paragraph of two or more sentences that has 25 to 1598 words. This dataset was created using web scraping and gathering texts from news articles and websites. The data was annotated manually by two native Marathi speakers.

The final collected dataset contained 1091 entries which were divided into train and test samples in a 70:30 ratio while maintaining class balance.

2.2. Pre-processing

The text samples were raw texts that contained fillers, punctuations, emojis, stem words and more. Each text sample was pre-processed to remove extra spaces, numbers, unwanted symbols, hashtags. These were removed and the texts were cleaned up.

Further to this, three different pre-processing techniques were used experimentally.

1. Stop-word removal (SWR): Stop words were removed from the text. Stop words were identified using GitHub.
2. Stemming (STM): Stemming is a process of normalizing individual words by reducing them to their base or root form.
3. Emoticon Replacement (EMR): Emoticons are widely used in texts on social media. These emoticons resemble a lot of sentiments which can be very useful. In some cases, emoticons can also be used to identify sarcasm or hidden meanings in texts [20].

There were seven permutations of the short text dataset that were created to perform the experiments on. In the following list, the individual pre-processing techniques mentioned are performed on the short text dataset. For example, EMR + SWR indicates that emoticon replacement and stop word removal were performed in that permutation.

i. EMR
ii. SWR
iii. STM
iv. EMR + STM
v. EMR + SWR
vi. SWR + STM
vii. EMR + STM + SWR

These permutations were applied on the short text dataset after data clean-up, and each permutation was stored as an individual CSV file.

2.3. Word Embeddings

The next step was to convert the text into word embedding vectors. The following word embedding technique was used - Word2Vec. Word embeddings were created by using the genism python library, using the Continuous Bag of Words (CBoW) method [21]; [22].

2.4. Model

The following SOA deep learning models were used:

1. Xlm - RoBERTa: A BERT model that is a multilingual version of the RoBERTa model that is trained in over 100 languages [15].
2. IndicBERT: A BERT model that is a multilingual version of the AlBERT model, and is pre-trained in 12 major Indian languages including Marathi [16].
3. MahaBERT: A multilingual BERT model that has been fine-tuned for the Marathi language [23].

2.5. Exploration

The MahaBERT model was used to predict hate speech separately on both the short-text and long-text datasets. the accuracy, AUC and F1-Score of these classifications were compared to evaluate how the length of the text affects the efficiency of hate speech detection in Marathi.

Since a discrepancy in the behavior was seen, ways to improve the model performance were explored such that it would performs equally well on both short-text and long-text datasets.

The following experiments were conducted on the Long Text Dataset to improve its performance (see Table 1).

Table 1: List of experiments on long-text dataset

No.	Experiment Description
1	Adding BiLSTM Layer after the model output and before final classification layer (**BL**)
2	Experiment 1 + Dropout layer (0.2) + Batch Normalization (**BLBND**)
3	Adding a CNN layer before the model (**CN**)
4	Experiment 3 + Droput Layer (0.2) + Batch Normalization (**CNBND**)
5	Experiment 1 + Experiment 2 (**CNBL**)
6	Experiment 2 + Experiment 4 (**CNBLBND**)

3. Results and Discussion

As described in section 2.4, three models were trained and evaluated on the seven pre-processed CSV files, implying 21 results in total. Each result contains AUC, F1-Score and Accuracy and so there were a total of 63 metrics to compare.

3.1 **Short Text Dataset - Model Evaluation and Metrics**

There was a total of 63 result metrics collected for the 3 models for 3 metrics over 7 pre-processing techniques. The following Table 2 contains a summarized view of all the metrics.

Table 2: Model Evaluation Metrics - Short Text Dataset

Pre-processing Technique	Evaluation Metric	Model		
		MahaBERT	XLM-RoBERTa	Indic-BERT
Without Preprocessing*	Accuracy	0.8280	0.8213	0.8331
	F1-Score	-	-	-
	AUC	-	-	-
EMR	Accuracy	0.9223	0.8326	0.8605
	F1-Score	0.9195	0.8446	0.8525
	AUC	0.9222	0.8312	0.8598
SWR	Accuracy	**0.9412**	0.8501	**0.8750**
	F1-Score	**0.9400**	0.8518	**0.8692**
	AUC	**0.9412**	0.8502	**0.8750**
STM	Accuracy	0.9191	0.8766	0.8620
	F1-Score	0.9139	0.8740	0.8641
	AUC	0.9185	0.8769	0.8634
EMR + STM	Accuracy	0.9131	**0.8827**	0.8453
	F1-Score	0.9107	**0.8798**	0.8349
	AUC	0.9132	**0.8833**	0.8446
EMR + SWR	Accuracy	0.9140	0.8855	0.8458
	F1-Score	0.9101	0.8862	0.8522
	AUC	0.9138	0.8858	0.8464
SWR + STM	Accuracy	0.9150	0.8821	0.8312
	F1-Score	0.9095	0.8840	0.8375
	AUC	0.9141	0.8822	0.8322
EMR + STM + SWR	Accuracy	0.9185	0.8432	0.8449
	F1-Score	0.9151	0.8524	0.8519
	AUC	0.9182	0.8517	0.8460

3.1.1 **Model-wise Evaluation of Pre-Processing Techniques**

For each model, it was observed that, using a preprocessing technique does improve the overall performance of the model.
In the case of MahaBERT, the accuracy score without pre-processing was 0.8280, while after using SWR, the accuracy score jumps up by 0.1132 to 0.9412 which was a huge jump.
In the case of XLM-RoBERTa, the accuracy score without pre-processing was 0.8213, while after using EMR+STM, the accuracy score jumps up by 0.0614 to 0.8827.
In the case of Indic-BERT, the accuracy score without pre-processing was 0.8331, while after using SWR, the accuracy score jumps up by 0.0419 to 0.8750. All other preprocessing techniques apart from SWR+STM used on IndicBERT yielded a performance better than what was achieved without pre-processing.

3.1.2 **Overall Evaluation of Pre-Processing Techniques**

Comparing single pre-processing techniques – EMR, STM and SWR, it was observed that SWR performed the best among these three, while EMR and STM performed equally.
For double pre-processing, it was observed that all three double pre-processing techniques performed equally well – EMR + STM, EMR + SWR, SWR + STM.
Furthermore, the triple pre-processing technique – EMR + STM + SWE performed the with the least improvement as compared to any single or double pre-processing technique.

3.2 **Long Text Dataset - Model Evaluation and Metrics**

The following Table 3 shows six experiments carried out on the long text data set and the results.

Table 3: Model Evaluation Metrics - Long Text Dataset

Experiment	Metrics		
	Accuracy	**F1-Score**	**AUC**
MahaBERT	0.8232	0.7514	0.6201
BL	0.8558	0.7490	0.6257
BLBND	0.8160	0.7474	0.6194
CN	0.8349	0.7290	0.6152
CNBND	0.8132	0.7213	0.6082
CNBL	**0.8898**	**0.7912**	**0.6589**
CNBLBND	0.8234	0.7362	0.6103

The following diagram details the flow of experiments conducted on the long text dataset (Fig. 1).

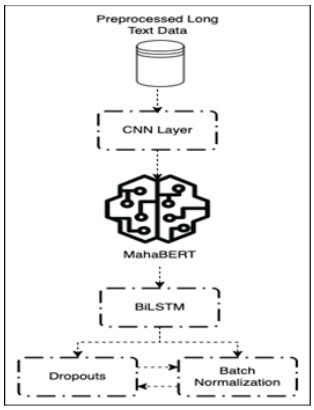

Fig.1 Flow diagram detailing experiments on long text dataset.

3.2.1 Effect of Class Imbalance - Comparison of Metrics

The MahaBERT model had an accuracy of 0.8232 on the long text dataset, but it had an AUC of just 0.6201. Furthermore, when compared, the accuracy score of MahaBERT model on the short text dataset was 0.8280 while that on the long text dataset was 0.8232. But the AUC score of the MahaBERT model on the short text dataset was 0.8132 which was close to the accuracy score, while the AUC of the MahaBERT model on the long text dataset was 0.6201 which was far lower than the accuracy score.

3.2.2 Evaluation of Experiments

The CNBL experiment performed the best, improving the model AUC from 0.6201 to 0.6589. Among the other experiments, only BL improved the model performance, while all other experiments yielded reduction in model performance.
All experiments that had the batch normalization and dropout layers yielded degradation in model performance.

4. Conclusions

There were two main studies conducted as a part of this research. The first experimented with different pre-processing techniques using three different models on the short text dataset. Accuracy, F1-Score, and AUC metrics were evaluated and compared.
The second study experimented with how size of texts affected model performance. For this, experiments were conducted using the MahaBERT model on the long text dataset.
The pre-processing techniques applied on the short text dataset yielded promising results. The accuracy of the models was improved on an average by 13.67%. It can be concluded that language specific pre-processing plays an important role in the performance of the model and a significant increase in model performance can be achieved by focusing on using the correct pre-processing techniques according to the language.
For the study conducted on the short-text dataset, it the difference between accuracy, F1-Score and AUC was negligible. But the same metrics evaluated in the study on the long text dataset yielded high accuracy but average F1-Score and AUC metrics. This indicated that studies conducted in this field should be cautions enough to consider all valid metrics before publishing results as using only accuracy is not enough.
Furthermore, the tests conducted on the MahaBERT model using longer texts from the long text dataset yielded a decrease in performance by 24%. This is a big drop in the model performance. Considering that a lot of hate speech is present on platforms that are not just social media sites and can contain large texts, these models won't perform as efficiently and would yield incorrect results.
Pre-processing techniques when chosen correctly can significantly improve model performance in the domain of hate speech detection. The nuances of the language should be considered when choosing the pre-processing technique.
Moreover, hate speech detection for Marathi currently has some significant work done for texts in social media websites. These texts are shorter in size lighter to process as well. But when it comes to longer texts that might be found in blogs, articles and news websites, the same models perform poorly. It is possible to improve the model performance on such longer texts by adding external components while keeping the model intact.
Although pre-processing techniques improve the model performance significantly, it is still up to the researcher to choose the correct pre-processing techniques. Work needs to be done to develop techniques that can identify the best pre-processing techniques based on nuances of a language. This can then be integrated with the hate-speech detection model and can function as one single model.
Furthermore, work needs to be done to build models that can better identify hate-speech in longer texts. These models need to have the intelligence to identify which part of the large text has hate-speech, if any. A good model will be one that can identify hate-speech in short as well as long texts equally efficiently.

References

[1] S.Wachs, M. Gámez-Guadix, and M.F.Wright, "Online Hate Speech Victimization and Depressive Symptoms Among Adolescents: The Protective Role of Resilience", vol. 257, pp.416–423, 2022. [Online] Available: https://www.liebertpub.com/doi/10.1089/cyber.2022.0009 [Accessed 1 Nov. 2022].

[2] N.Sevani, I.A. Soenandi, Adianto and J.Wijaya, "Detection of Hate Speech by Employing Support Vector Machine with Word2Vec Model". 7th International Conference on Electrical, Electronics and Information Engineering: Technological Breakthrough for Greater New Life, ICEEIE 2021.

[3] S.Khan, M.Fazil, V.K. Sejwal, M.A. Alshara, R.M. Alotaibi, A. Kamal, and A.R. Baig, "BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection", Journal of King Saud University - Computer and Information Sciences, vol. 347, pp.4335–4344, 2022.

[4] J.Devlin, M.W.Chang, K. Lee and K. Toutanova "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, NAACL HLT 2019. 1, pp.4171–4186, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805v2 [Accessed 29 Oct. 2022].

[5] Z. Yang, Z. Dai, Y.Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding". Advances in Neural Information Processing Systems, vol. 32, 2019. [Online]. Available: https://arxiv.org/abs/1906.08237v2 [Accessed 29 Oct. 2022].

[6] T. Fuchs, and F., Schäfer, "Normalizing misogyny: hate speech and verbal abuse of female politicians on Japanese Twitter." https://doi.org/10.1080/09555803.2019.1687564, vol. 334, pp.553–579, 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/09555803.2019.1687564 [Accessed 29 Oct. 2022].

[7] W. Aldjanabi, A. Dahou, M.A.A. Al-Qaness, M.A. Elaziz, A.M. Helmi, and R. Damaševičius, "Arabic Offensive and Hate Speech Detection Using a Cross-Corpora Multi-Task Learning Model". Informatics 2021, vol. 8, p. 69, 2021. [Online]. Available: https://www.mdpi.com/2227-9709/8/4/69/htm [Accessed 29 Oct. 2022].

[8] K. Perifanos, D. Goutsos, M.Montes-Y-Gómez, and P. Rosso, (2021) "Multimodal Hate Speech Detection in Greek Social Media". Multimodal Technologies and Interaction, vol. 5, p. 34, 2021. [Online]. Available t: https://www.mdpi.com/2414-4088/5/7/34/htm [Accessed 29 Oct. 2022].

[9] I. Markov, I. Gevers, and W. Daelemans, "An Ensemble Approach for Dutch Cross-Domain Hate Speech Detection". Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13286 LNCS, pp.3–15, 2022. [Online]. Available:

https://link.springer.com/chapter/10.1007/978-3-031-08473-7_1 [Accessed 29 Oct. 2022].

[10] M.S.A. Sanoussi, C. Xiaohua, G.K. Agordzo, M.L. Guindo, A.M.M.A. Al Omari, and B.M. Issa, "Detection of Hate Speech Texts Using Machine Learning Algorithm". 2022 IEEE 12th Annual Computing and Communication Workshop and Conference, CCWC 2022, pp.266–273, 2022.

[11] P. Mathur, R. Sawhney, M. Ayyar, and R. Shah, "Did you offend me? Classification of Offensive Tweets in Hinglish Language". Proceedings of the 2nd Workshop on Abusive Language Online (ALW2). Brussels, Belgium: Association for Computational Linguistics, pp.138–148, 2018. [Online] Available: https://aclanthology.org/W18-5118.

[12] T.T. Sasidhar, P. B and S.K. P, "Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text". Procedia Computer Science, vol. 171, pp.1346–1352, 2020. [Online] Available: https://www.sciencedirect.com/science/article/pii/S1877050920311236.

[13] M. Romim Naurosand Ahmed, T.H. and S.I.M., "Hate Speech Detection in the Bengali Language: A Dataset and Its Baseline Evaluation". J.C. Uddin Mohammad Shorifand Bansal, ed., Proceedings of International Joint Conference on Advances in Computational Intelligence. Singapore: Springer Singapore, pp.457–468. 2021

[14] P.K. Roy, S. Bhawal, and C.N. Subalalitha, "Hate speech and offensive language detection in Dravidian languages using deep ensemble framework". Computer Speech & Language, vol. 75, p.101386, 2022.

[15] A. Conneau, K.Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M.Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale", pp.132–135, 2019. [Online]. Available: https://arxiv.org/abs/1911.02116v2 [Accessed 29 Oct. 2022].

[16] D. Kakwani, A. Kunchukuttan, S. Golla, N.C. Gokul, A. Bhattacharyya, M.M. Khapra, and P. Kumar, "Findings of the Association for Computational Linguistics IndicNLPSuite: Monolingual Corpora". Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. vol.3, p.5, 2020. [Online]. Available: https://github.com/AI4Bharat/indicnlp_catalog [Accessed 29 Oct. 2022].

[17] S. Gaikwad, T. Ranasinghe, M. Zampieri, and C.M. Homan "Cross-lingual Offensive Language Identification for Low Resource Languages: The Case of Marathi". [Online]. Available: http://arxiv.org/abs/2109.03552. (2021)

[18] A. Velankar, H. Patil, A. Gore, S. Salunke and R. Joshi, "L3Cube-MahaHate: A Tweet-based Marathi Hate Speech Detection Dataset and BERT models", 2022. [Online]. Available: https://arxiv.org/abs/2203.13778 [Accessed 8 Oct. 2022].

[19] A. Velankar, H. Patil, and R. Joshi, "Mono vs Multilingual BERT for Hate Speech Detection and Text Classification: A Case Study in Marathi", 2022. [Online]. Available: http://arxiv.org/abs/2204.08669 [Accessed 8 Oct. 2022].

[20] LahotiPawan, MittalNamita and SinghGirdhari, "A Survey on NLP resources, tools and techniques for Marathi Language Processing". Transactions on Asian and Low-Resource Language Information Processing, 2021. [online] Available: https://dl.acm.org/doi/10.1145/3548457 [Accessed 29 Oct. 2022].

[21] R. Rehurek, and P. Sojka, "Gensim--python framework for vector space modellin"g. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 32, 2011.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space". 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, (2013). [Online] Available: https://arxiv.org/abs/1301.3781v3 [Accessed 29 Oct. 2022].

[23] R.Joshi, "L3Cube-MahaCorpus and MahaBERT: Marathi Monolingual Corpus". Marathi BERT Language Models, and Resources, 2022. [Online] Available: https://huggingface.co/l3cube-pune/marathi-bert [Accessed 29 Oct. 2022].