Battling Hateful Content in Indic Languages HASOC '21

Aditya Kadam[a], Anmol Goel[a], Jivitesh Jain[a], Jushaan Singh Kalra[b],
Mallika Subramanian[a], Manvith Reddy[a], Prashant Kodali[a], T.H. Arjun[a],
Manish Shrivastava[a] and Ponnurangam Kumaraguru[a]

[a]*International Institute of Information Technology, Hyderabad, India*
[b]*Delhi Technological University, Delhi, India*

Abstract

The extensive rise in consumption of online social media (OSMs) by a large number of people poses a critical problem of curbing the spread of hateful content on these platforms. With the growing usage of OSMs in multiple languages, the task of detecting and characterizing hate becomes more complex. The subtle variations of code-mixed texts along with switching scripts only add to the complexity. This paper presents a solution for the HASOC 2021 Multilingual Twitter Hate-Speech Detection challenge by team *PreCog IIIT Hyderabad*. We adopt a multilingual transformer based approach and describe our architecture for all 6 subtasks as part of the challenge. Out of the 6 teams that participated in all the subtasks, our submissions rank 3rd overall.

Keywords

Hate Speech, Social Media, Code Mixed, Indic Languages, Transformer Architecture

1. Introduction

Dissemination of hateful content on nearly all social media is increasingly becoming an alarming concern. In the research community as well, this is a heavily studied research problem [1, 2, 3, 4, 5]. Misconduct such as bullying, derogatory comments based on gender, race, religion, threatening remarks etc. are more prevalent today than ever before. The repercussions that such content can have is profound and can result in increased mental stress, emotional outburst and negative psychological impacts [6]. Hence, curbing the proliferation of this hate speech is imperative. Furthermore, the massive scale at which online social media platforms function makes it an even more pressing issue, which needs to be addressed in a robust manner. Most online social media platforms have imposed strict guidelines [1] [2] [3] to help prevent the spread of hate. In spite of these platform regulations, the dynamics of user-interaction influence the diffusion of (and hence increase in) hate to a large extent [1].

The problem of hate speech has been addressed by several researchers, but the rise in multilingual content has added to the complexity of identification of hateful content. Majority of these

[1]https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy
[2]https://transparency.fb.com/en-gb/policies/community-standards/hate-speech/
[3]https://support.google.com/youtube/answer/2801939

studies deal with high-resource languages such as English, and only recently have low-resource languages – such as several Indic Languages – been more deeply explored [7]. In a country like India, with multitude of regional languages, the phenomenon of Code Mixing/Switching (wherein linguistic units such as phrases/words of two languages occur in a single utterance) is also pervasive.

In this paper we elucidate our approach in solving the six downstream tasks of hate speech identification and characterization in Indian languages as a part of the *'HASOC '21 Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages'* challenge [8]. Motivated by existing architectures, we curate our own pipeline by fusing fine-tuned transformer based models with additional features to solve this challenge and highlight the different methodologies that were adopted for the three languages – English, Hindi, Marathi, and Code Mixed Hindi - English. We also make our code, methodology and approach public to the research community. [4]

2. Literature Review

Discerning hateful content on social media is an already tricky problem given the challenges associated with it, for instance disrespectful/abusive words could be censored in text, some expressions may not be inherently offensive, however they can be so in the right context [9]. Owing to the conversational design of social media wherein users can reply to a given comment (either support, refute or irrelevant to the original message), the build-up of threads in response to a hateful message can also intensify hate even if the reply is not hateful on its own. The evolution of such hate intensity has shown diverse patterns and no direct correlation to the parent tweet which makes the task of hate speech detection more difficult [10].

Significant amount of research has been conducted to evaluate traditional NLP approaches such as character level CNNs, word embedding based approaches and the myriad of variations with LSTMs (sub-word level, hierarchical, BiLSTMs) [11]. Likewise, Machine Learning algorithms including SVMs, K-Nearest Neighbours, Multinomial Naive Bayes (MNB) and their respective performances in multilingual text settings have also been explored [12, 7, 13]. Investigating categories of profane words that are commonly used in hate speech is another non-trivial subtask under the hate detection umbrella, primarily because of the different interpretations of words in different cultures/demographics, adaptation of slangs in newer generations etc [14].

In recent times however, with the introduction of Transformer based models and their performance in Natural Language Understanding (NLU) tasks, significant work has been done in order to adapt these for multilingual texts as well to leverage transfer between languages. Models such as XLMR, mBERT, MuRIL, RemBERT have gained much popularity and have shown promising results [15, 16, 17]. Transfer learning based approaches that leverage performance of high resource languages accompanied with CNN classification heads have also shown significant improvements in capturing hateful content on social media platforms [18, 19]. Sharing and re-utilizing the model weights learnt whilst training on a corpus for a high resource language can aid the process of training for languages that are still under explored [20].

---

[4]https://github.com/Adi2K/Precog-HASOC-2021

3. Dataset

3.1. Dataset & Task Description

Subtask 1 consisted of data for 3 languages, namely – English, Hindi and Marathi [21, 22]. For English and Hindi, the task was further subdivided into 2 sub-parts: *a)* Identification of hateful v/s non-hateful content and *b)* Characterizing the kind of hate present in a tweet – either Profane, Hateful, Offensive or None. The distribution of the different data classes for each of the three languages is shown in Table 1.

Table 1
Distribution of the HASOC 2021 dataset for each task and its associated subtasks. For each language and task, the corresponding number of tweets per class is shown below. Tweets pertaining to Subtask 1 are in three languages – English, Hindi, Marathi.

| | Task A | | Task B | | | |
Language	Non-Hateful	Hateful	None	Offensive	Hate	Profane
English	1342	2501	1342	1196	683	622
Hindi	3161	1433	3161	654	566	213
Marathi	1205	669	-	-	-	-
Subtask 2	2899	2841	-	-	-	-

The focus of **Subtask 2** was binary classification: Hate & Offensive or Non Hate-Offensive. The given dataset distribution for this task is shown in Table 1. This data was accompanied by the following additions [23]:

- Tweets are English - Hindi Code Mix sentences, and
- Classification should not be based on the tweet alone, but should also account for the context as well.
 For example : Consider that in a tweet thread, tweet A is a reply to tweet B. For classifying tweet A, the model can leverage the information from the parent tweet - tweet B.

Figure 3b demonstrates the relationship between the tweets to be classified and their contexts.

3.2. Preprocessing Data

As a precursor to applying any NLP models on text data, we pre-processed the dataset with standard techniques. Given that the data from Twitter is bound to have certain amount of noise and unwanted elements such as – URLs, mentions etc, these were removed from the tweet texts. Hashtags have a slightly different contribution to analysis of the tweet since they may or may not contribute positively in the classification task. Through the results from our experiments, we observed that omitting the hashtags proved to work better, and hence they were cleaned from the tweet as well.

Since the data is code mixed, not only in terms of the combination of languages but also with respect to scripts (some English text is written in Roman script, whereas some Hindi text is written in Devanagari apart from Roman), we also normalize the Indic language scripts for
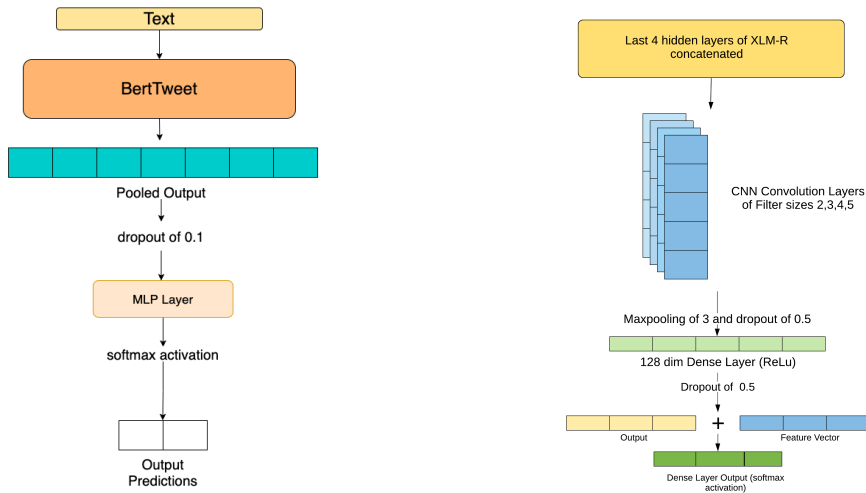
(a) Using BerTweet model with a MLP classifier head.

(b) Combining CNN features over XLM-R output and manually generated feature vectors.

Figure 1: Architecture and pipeline for the models used for the downstream task of hate detection and classification for the English language for Subtask 1.

Marathi and Hindi. In addition to that, we removed stop words for the Marathi dataset using this list. [5] Finally, punctuations were also removed from the dataset texts.

An interesting observation was that for the task of hate detection, the presence of emojis converted to text in the tweets did not improve the performance of our models significantly (rather it reduced the scores by some margin). However, including emojis along with text while classifying hate did have a positive impact since the emoji-text conversion was able to capture hints of sentiment and indirect offensive/profane content.

4. Methodology

4.1. Subtask 1: Identifying Hate, offensive and profane content from the post

4.1.1. English Classifiers

For the English Subtask 1, the architecture that resulted in the best performance is an ensemble of the following models:

- Fine-tuned **BERTweet** model [24]
- Fine-tuned **XLM-Roberta** [16] with CNN Head

We use XLM-R, a multilingual model, along with the monolingual model in the ensemble as we found that some of the text in the training set has transliterated Hindi along with some Devanagari text. We extracted textual features such as distribution of '?', '!', capital letters etc.

---

[5]https://github.com/stopwords-iso/stopwords-mr

We also use the percentage of profane words and sentiment of the text as a feature. We use profane words list curated from various sources such as words/cuss [6], zacanger/profane-words [7], t-davison/lexicons. [8] For sentiment analysis we use the TweetEval [25] model and use its softmax output as a feature to our models.

Inspired by Kim [26] we pass the embedding (concatenated last 4 hidden layers) to a CNN and max-pool convolution layers of various widths to a fully connected layer of size 128 with dropout. We concatenate this 128 dimensional vector with our feature vector. We pass this output onto a dense output layer with softmax activation and cross entropy loss as shown in Figure 1.

Along with the previous models, we fine-tune BERTweet, a pre-trained language model for English tweets. BERTweet has the same architecture as BERT and is trained on the pre-training procedure of RoBERTa, but it is trained solely on tweets, thus, making it a viable alternative and suitable for our task. This model has shown state-of-the-art results on tasks based on tweets [24]. We use the encoder architecture and pass the pooled output through a linear layer for the classification which uses softmax activation and cross-entropy loss as shown in the Figure 1.

We also train the models on the previous years datasets but notice that this does not increase the performance of the models but actually degrades the performance in Task 1B due to skewed distribution of classes. Transliteration of emojis didn't improve the performance. The class imbalance in Subtask 1-B degraded the performance of our models hence we tried to improve upon it by using a weighted loss function but we notice that this decreases the performance and that the domain specific distribution is actually helping the models. We also perform K-Fold Validation and use early stopping to avoid over-fitting. We average the probabilities of each class across folds and the two models in our ensemble.

4.1.2. Hindi & Marathi Classifier

For both the Hindi and Marathi language, the architecture that performed the best utilized the **XLM-R** transformer model. This model was able to capture the code-mixed and multilingual nature of the tweets dataset. To amplify the results, we leveraged intermediary representations of the language model as well as textual features that were extracted from the tweets. In particular, we utilized the Multilingual MiniLM language model for fine-tuning on Hindi Subtask 1-B. We observed that MiniLM with Focal Loss instead of Cross Entropy Loss performed better than other baselines in the imbalanced multi-class setting of Hindi Subtask 1-B. Focal Loss compensates for class imbalance with a factor that increases the network's sensitivity towards mis-classified samples.
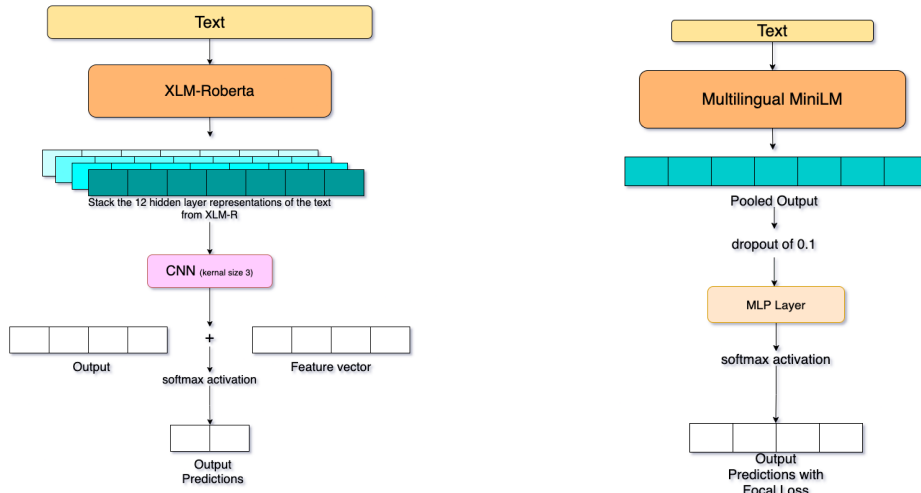
Inspired by Mozafari et al. [18] we use the pre-trained representations of the text from 12 hidden layer of XLM-R model (each of 768 dimensions) and then apply a CNN layer with a kernel size of 3. The output is then passed through a soft-max following which the cross-entropy loss is computed whilst training. This model architecture is represented in Figure 2. Tuning hyperparameters such as optimizers, loss functions and dropout layers, we experiment with

---

[6] https://github.com/words/cuss
[7] https://github.com/zacanger/profane-words
[8] https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/lexicons

(a) The base architecture for Hindi & Marathi languages for Subtask 1 using XLM-R with CNN augmented with textual features vector followed by a softmax layer.

(b) Multilingual MiniLM architecture adopted to overcome class imbalance while characterizing hate for the Hindi Subtask 1-B.

Figure 2: Architecture and pipeline for the models used for the downstream task of hate detection and classification for the Hindi & Marathi language for Subtask 1.

different options. For the optimizers we try Adadelta and Adam optimizers with Adam working out better. Amongst all loss functions, the Cross Entropy Loss performed the best. As for the dropout layers we explore dropouts in the range 0.1-0.5 and use 0.5 as the final dropout for the model architecture.

We further augment the model features, with two kinds of textual features – fraction of profane words and sentiment of the tweet. Due to lack of resources for Marathi we catalogue [9] a list of profane words in Marathi and use this to find the fraction of profane words in a tweet. For Hindi, we curate a list of profane words by collating and appending to existing lists [10], and use this to score each tweet. As for the sentiment of the tweet, we incorporated off-the-shelf HuggingFace models to obtain the positive, negative and neutral scores for a tweet [11] [12]. Although the textual features improved the performance for Hindi only by a small margin, for Marathi, manually extracted textual features helped in achieving a significant boost.

For the Marathi Subtask 1, we experimented with a voting ensemble of the **XLM-Roberta** with CNN Head using the following features:
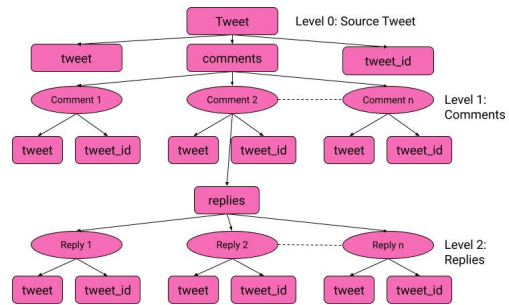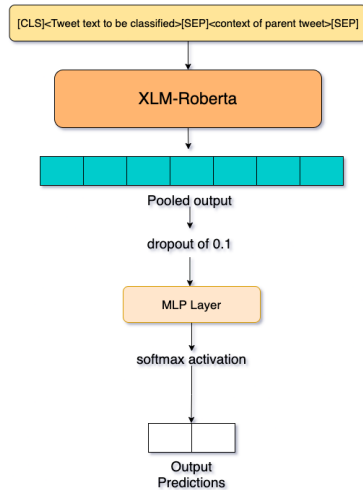
- Word Embedding + Fraction of Profane Words + Sentiment Polarity
- Word Embedding + Sentiment Polarity

---

[9]https://github.com/Adi2K/MarathiSwear

[10]https://github.com/neerajvashistha/online-hate-speech-recog/blob/master/data/hi/Hinglish-Offensive-Text-Classification/Hinglish_Profanity_List.csv

[11]https://huggingface.co/l3cube-pune/MarathiSentiment

[12]https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment

(a) Model pipeline for hate detection in conversational threads for Subtask 2.

(b) Hierarchy of a conversation thread and its associated comments.

Figure 3: Model Pipeline and tweet conversation thread example for Subtask 2.

- Word Embedding

However we noticed that the base model with the embedding and the textual features performed better on the leaderboard.

4.2. Subtask 2 : Identification of Conversational Hate-Speech in Code-Mixed Languages (ICHCL)

The tweets for Subtask 2 are code mixed. While the Transformer based encoder models have performed well on various monolingual NLU tasks, their performance does not reach the same level on code mixed sentences. Multilingual transformer based models, have been applied for various code mixed NLU tasks, and have performed better than monolingual transformer based models [27]. For this task, we use XLM-RoBERTa [16]. To capture the context and the tweet itself, we modify the input in the following manner, where [CLS] , [SEP] are part of the vocabulary of model, and are used to classify and take multiple sentences as input, respectively.

> **[CLS]** <Tweet text to be classified> **[SEP]** <context of parent tweet> **[SEP]**

Here, *<Tweet text to be classified>* is the text of the tweet/comment/reply that is being classified, while *<Context of parent tweet>* is either just the parent tweet or concatenation of parent tweet and comment, depending on weather the text to be classified is a tweet or a comment or a reply. While classifying a standalone tweet, the context is left empty. The Hindi corpus used to train XLM-Roberta is in Devanagari script, while there is only a small portion of the corpus which is in Romanised form. With the hypothesis that the performance of model will improve if the

Hindi tokens are in Devanagari script, we used CSNLI tool [13] to convert the Romanised tokens to Devanagari script. However, this normalisation only had a marginal impact on the final performance of the model. We used Huggingface's Trainer API to train the XLM-R model, and the hyperparameters were chosen using the hyperparameter search functionality offered by Trainer API.

Table 2
Performance scores for each of the six subtasks in terms of their test accuracy percentage and Macro F1 scores. All the architectures that were experimented and tested out are tabulated here. We can observe form the results that XLM-R combined with CNN classifier head works best across the languages of Subtask 1, while for Subtask 2, XLM-R with normalised input text performs the best in our experiments.

Language	Subtask	Method	Accuracy	Macro F1
English	A	XLM-R + CNN	62.30	0.62
		Ensemble	79.94	0.78
		XLM-R + CNN + Sentiment Scores	**81.03**	**0.79**
English	B	XLM-R + CNN + Weighted loss	60.50	0.53
		Ensemble	**65.18**	**0.59**
Hindi	A	MuRIL	68.90	0.60
		XLM-R Base	74.00	0.76
		XLM-R + CNN	**80.09**	**0.77**
Hindi	B	**MiniLM with Focal Loss**	**72.64**	**0.51**
Marathi	A	XLM-R Base	84.16	0.86
		Ensemble	88.48	0.87
		XLM-R + CNN	**88.64**	**0.87**
Hi-En Code mix	2	XLM-R without norm	67.58	0.67
		XLM-R with norm	**69.36**	**0.70**

4.3. Experiments

We used Huggingface Transformers [28] library for implementing the classifiers. For hyper parameter tuning we use Optuna Framework [14] library. Exploring multiple architectures simultaneously, we also tried ensembling an odd number of models following a majority rule based selection. For the English Subtask 1 we also did ensembling with averaged softmax probabilities. However, the increase in complexity of the classification pipeline did not necessarily improve performance scores, considering the size and the distribution of the dataset for Hindi and Marathi but helped in English. Table 2 captures the Accuracies and F1 scores (corresponding to submissions made on the leaderboard) of all our models for each of the subtasks.

---

[13]https://github.com/irshadbhat/csnli
[14]https://optuna.org/

5. Conclusion

In this paper, we presented our approaches for Hate Speech detection on Indian Languages and code mix between Hindi-English using multilingual transformer based encoder models. Although, in this work we have employed different models to address individual language specific subtasks, a multi-task single model based approach, which performs well across all the language pairs, would be an interesting challenge, which we wish to explore as a future work. In addition to this, as part of future work, we would like to improve the performance by carrying out an additional step of domain adaptive pre-training of the encoder models, and an efficient ensemble of multilingual encoder models.

Acknowledgments

References

[1] B. Mathew, R. Dutt, P. Goyal, A. Mukherjee, Spread of hate speech in online social media, in: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 173–182. URL: https://doi.org/10.1145/3292522.3326034. doi:10.1145/3292522.3326034.

[2] L. Silva, M. Mondal, D. Correa, F. Benevenuto, I. Weber, Analyzing the targets of hate in online social media, in: Tenth international AAAI conference on web and social media, 2016.

[3] M. Mozafari, R. Farahbakhsh, N. Crespi, Hate speech detection and racial bias mitigation in social media based on bert model, PLOS ONE 15 (2020) 1–26. URL: https://doi.org/10.1371/journal.pone.0237861. doi:10.1371/journal.pone.0237861.

[4] Z. Mossie, J.-H. Wang, Vulnerable community identification using hate speech detection on social media, Information Processing & Management 57 (2020) 102087. URL: https://www.sciencedirect.com/science/article/pii/S0306457318310902. doi:https://doi.org/10.1016/j.ipm.2019.102087.

[5] K. A. Qureshi, M. Sabih, Un-compromised credibility: Social media based multi-class hate speech classification for text, IEEE Access 9 (2021) 109465–109477. doi:10.1109/ACCESS.2021.3101977.

[6] K. Saha, E. Chandrasekharan, M. De Choudhury, Prevalence and psychological effects of hateful speech in online college communities, in: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 255–264. URL: https://doi.org/10.1145/3292522.3326032. doi:10.1145/3292522.3326032.

[7] T. Ranasinghe, M. Zampieri, An evaluation of multilingual offensive language identification

methods for the languages of india, Information 12 (2021). URL: https://www.mdpi.com/2078-2489/12/8/306. doi:10.3390/info12080306.

[8] 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.

[9] G. Kovács, P. Alonso, R. Saini, Challenges of hate speech detection in social media, SN Computer Science 2 (2021) 95. URL: https://doi.org/10.1007/s42979-021-00457-3. doi:10.1007/s42979-021-00457-3.

[10] S. Dahiya, S. Sharma, D. Sahnan, V. Goel, E. Chouzenoux, V. Elvira, A. Majumdar, A. Bandhakavi, T. Chakraborty, Would your tweet invoke hate on the fly? forecasting hate intensity of reply threads on twitter, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2732–2742. URL: https://doi.org/10.1145/3447548.3467150. doi:10.1145/3447548.3467150.

[11] 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.

[12] P. Rani, S. Suryawanshi, K. Goswami, B. R. Chakravarthi, T. Fransen, J. P. McCrae, A comparative study of different state-of-the-art hate speech detection methods in Hindi-English code-mixed data, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 42–48. URL: https://aclanthology.org/2020.trac-1.7.

[13] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga, Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions, Computer Science Review 38 (2020) 100311. URL: https://www.sciencedirect.com/science/article/pii/S1574013720304111. doi:https://doi.org/10.1016/j.cosrev.2020.100311.

[14] P. L. Teh, C.-B. Cheng, W. M. Chee, Identifying and categorising profane words in hate speech, in: Proceedings of the 2nd International Conference on Compute and Data Analysis, ICCDA 2018, Association for Computing Machinery, New York, NY, USA, 2018, p. 65–69. URL: https://doi.org/10.1145/3193077.3193078. doi:10.1145/3193077.3193078.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational

Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.

[17] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, P. Talukdar, Muril: Multilingual representations for indian languages, 2021. arXiv:2103.10730.

[18] 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.

[19] I. Bigoulaeva, V. Hangya, A. Fraser, Cross-lingual transfer learning for hate speech detection, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, Association for Computational Linguistics, Kyiv, 2021, pp. 15–25. URL: https://aclanthology.org/2021.ltedi-1.3.

[20] T. Ranasinghe, M. Zampieri, Multilingual offensive language identification for low-resource languages, CoRR abs/2105.05996 (2021). URL: https://arxiv.org/abs/2105.05996. arXiv:2105.05996.

[21] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, 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] S. Gaikwad, T. Ranasinghe, M. Zampieri, C. M. Homan, Cross-lingual offensive language identification for low resource languages: The case of marathi, in: Proceedings of RANLP, 2021.

[23] S. Satapara, S. Modha, T. Mandl, H. Madhu, P. Majumder, Overview of the HASOC Subtrack at FIRE 2021: Conversational Hate Speech Detection in Code-mixed language, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[24] D. Q. Nguyen, T. Vu, A. T. Nguyen, BERTweet: A pre-trained language model for English Tweets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 9–14.

[25] F. Barbieri, J. Camacho-Collados, L. Espinosa-Anke, L. Neves, TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification, in: Proceedings of Findings of EMNLP, 2020.

[26] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1746–1751. URL: https://aclanthology.org/D14-1181.

[27] S. Khanuja, S. Dandapat, A. Srinivasan, S. Sitaram, M. Choudhury, GLUECoS: An evaluation benchmark for code-switched NLP, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3575–3585. URL: https://aclanthology.org/2020.acl-main.329. doi:10.18653/v1/2020.acl-main.329.

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao,

S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.