

Leveraging Transformers for Hate Speech Detection in Conversational Code-Mixed Tweets

Zaki Mustafa Farooqi, Sreyan Ghosh and Rajiv Ratn Shah

Multimodal Digital Media Analysis Lab, Indraprastha Institute of Information Technology Delhi, India

Abstract

In the current era of the internet, where social media platforms are easily accessible for everyone, people often have to deal with threats, identity attacks, hate, and bullying due to their association with a cast, creed, gender, religion, or even acceptance or rejection of a notion. Existing works in hate speech detection primarily focus on individual comment classification as a sequence labelling task and often fail to consider the context of the conversation. The context of a conversation often plays a substantial role when determining the author's intent and sentiment behind the tweet. This paper describes the system proposed by team MIDAS-IIITD for HASOC 2021 subtask 2, one of the first shared tasks focusing on detecting hate speech from Hindi-English code-mixed conversations on Twitter. We approach this problem using neural networks, leveraging the transformer's cross-lingual embeddings and further fine-tuning them for low-resource hate-speech classification in transliterated Hindi text. Our best performing system, a hard voting ensemble of Indic-BERT, XLM-RoBERTa, and Multilingual BERT, achieved a macro F1 score of 0.7253, placing us 1st on the overall leaderboard standings.

Keywords

Code-Mixed Languages, Hindi-English, Hate Speech, Transformers, Offensive Tweets

1. Introduction

In today's world, hate speech is one of the major issues plaguing online social media websites. Platforms like Twitter and Gab make it easier than ever before for a person to reach a large audience quickly, which results in an increased temptation of users for inappropriate behavior such as hate speech, causing potential damage to the social system and thus possessing major threats which has already led to different types of crimes [1]. Human moderators manually detecting hate speech online have been reported to go through trauma and mental issues. This phenomenon necessitates automated hate speech detection as a crucial task.

A majority of the work on hate speech classification is constrained to the *English* language. The inability of mono-lingual hate speech classifiers to detect the semantic cues in code-mixed languages necessitates an efficient classifier that can detect offensive content automatically from code-mixed languages. Hinglish (formed of the words spoken in Hindi language but written in Roman script instead of the Devanagari script) extends its grammatical setup from native Hindi, accompanied by many slurs, slang, and phonetic variations due to regional

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

✉ zaki19048@iiitd.ac.in (Z. M. Farooqi); gsreyan@gmail.com (S. Ghosh); rajivrtn@iiitd.ac.in (R. R. Shah)

🌐 <https://github.com/zmf0507> (Z. M. Farooqi); <https://github.com/Sreyan88> (S. Ghosh); <http://midas.iiitd.edu.in/> (R. R. Shah)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

influence. Randomized spelling variations and multiple possible interpretations of Hinglish words in different contextual situations make it extremely difficult to deal with for automated classification. Another challenge worth considering in dealing with Hinglish is the demographic divide between Hinglish users relative to total active users globally. This poses a severe limitation as the tweet data in Hinglish language is a small fraction of the large pool of tweets generated, necessitating the use of selective methods to process such tweets in an automated fashion.

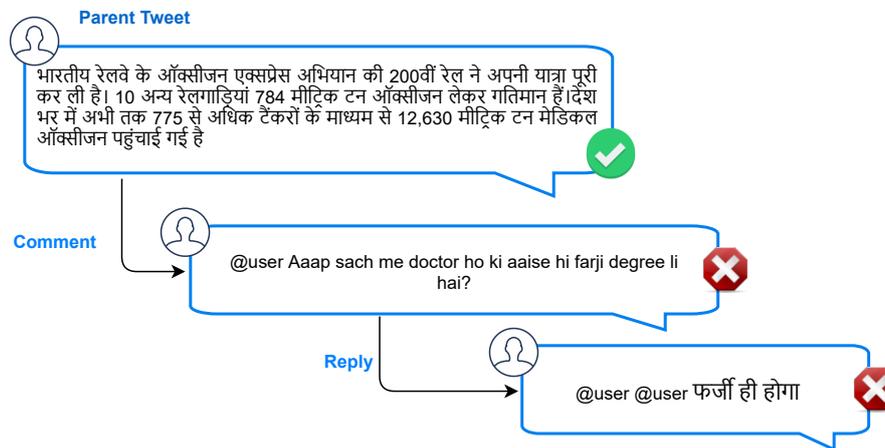


Figure 1: An example of conversation from the dataset where the parent tweet is not hateful but the comment and reply are expressing implicit hate towards the user who posted parent tweet. The post explains the amount of oxygen that has been supplied across the whole country. The hateful comment says “Are you a genuine doctor or have you just acquired a fake degree?” while the hateful reply gives an affirmative response by saying “it must be fake”. The green tick represents “NOT” comments while the red cross represents “HOF” or hateful/offensive comments.

The context of the conversation plays a very crucial role in hate speech identification. We acknowledge the fact that a comment categorized as hate speech may not always contain the subject of hate on its own. As shown in figure 1, agreement or disagreement with a previous comment or the overall ideology in the chain of the conversation might also induce hate towards a particular target group. In addition to this, systems that can efficiently utilize the entire context of the comment chain with a holistic understanding of the entire discussion may also help in detecting “trigger comments” i.e., non-toxic comments in online discussions which lead to toxic replies and implicitly help in mitigating bias in hate speech identification which remains a long-standing problem in this domain [2, 3].

Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2021 [4] proposes two subtasks where subtask 1 aims towards identifying and discriminating hate and profane tweets in English, Hindi, and Marathi, and subtask 2 aims towards detecting hate tweets in conversations primarily in Hindi-English code-mixed texts. This paper illustrates our key contribution to subtask 2 [5] of HASOC 2021 [4]. We present our system based on ensemble of transformers to detect hate speech in code-mixed Hindi-English con-

versations, which helps us achieve the first place in the HASOC Subtask 2 final leaderboard standings.

2. Related Work

Hate speech detection is a challenging task with literature including techniques such as dictionary-based [6], distributional semantics [7] and recent literature exploring the power of neural network architectures for the same [8]. However, a majority of the work done on hate speech detection is constrained to the English language [9, 10, 11, 12, 13, 14, 15, 16] with very limited work on other foreign languages [17, 18, 19, 20, 21] and code-switched text [22, 23, 24, 25, 26]. Although Hinglish has been a major contributor to hate speech online, this area has seen very little work with recent work exploring transformers [27] and author profiling using graph neural networks [25]. Works like [22] uses a Convolutional Neural Network (CNN) architecture together with Glove embeddings and transfer learning, in one of the first attempts to detect hate-speech online in the *Hinglish* language. Similar to our work, [28] explored XLM-RoBERTa and achieved competitive results on the task of detecting hate-speech in Dravidian languages. In HAOSC 2020 shared task, [29] also used XLM-RoBERTa to achieve the third rank on the overall task of detecting offensive content in code-mixed Dravidian text. However, we acknowledge the fact that XLM-RoBERTa was pre-trained on Hindi only text and our work differs from theirs in which we also transliterate words in a different code to Hindi to solve the problem of hate-speech detection.

Hate speech detection has branched into several sub-tasks like toxic span extraction [30, 31], rationale identification [32] and hate target identification [20]. Though recent advancement in the field of NLP has pushed the limits of hate speech identification, like transformers [25] and graph neural networks [33, 25, 34] with people attempting to induce external knowledge leveraging author profiling [25] or ideology [35] but using context of the conversation is still a challenge with very little work exploring this problem. Context of the conversation plays a huge role in hate speech identification with recent literature exploring both the structure and effect of context [36, 37] for the same. One interesting and related direction of work described in [38] relates to building systems that generate text which acts as hate speech interventions in online discussion. Context can both help in detecting trigger comments [39] and implicitly handle bias in hate speech identification which remains a long-standing problem in this domain [2, 3]. To the best of our knowledge, all these works are constrained to the English language with very little work on code-mixed Hindi-English and Hinglish text, considering the context of the conversation.

3. Dataset

The dataset provided for this task has code-switched Hindi-English as well as Hinglish conversation chains taken from twitter as shown in Fig 1. This is a binary classification dataset having two classes HOF and NOT. HOF denotes the tweet, comment, or reply which contains hate, offensive, and profane content in itself or is supporting hate, whereas NOT denotes the tweet, comment, or reply which does not contain any hate speech, profane, or offensive content.

More details about the dataset provided to us can be found in table 1. We have also provided *Avg. Comments* which tells us about the average number of comments to a *Parent Tweet* in the dataset since all *Parent Tweets* had at least one *comment*. We do not provide the average number of replies for comments since all comments do not have replies.

Table 1
Original Dataset Statistics

Data	Total Conversations	HOF	NOT	Parent Tweets	Total Comments	Total Replies	Avg. Comments
Train	5740	2841	2899	82	3778	1880	46
Test	1348	695	653	16	849	483	53

Table 2
Train-Validation-Test Distribution

Data	Total Conversations	HOF(Hateful/Offensive)	NOT (neither Hateful nor Offensive)
Train	4592	2273	2319
Val	1148	568	580
Test	1348	695	653

Note : **Val** refers to Validation data .

For feeding data into our model, we flatten the given conversations into individual parent-comment-reply unique conversation chains. As mentioned earlier, all parent tweets have atleast one comment, but all comments do not have replies, so, a training instance might end with a comment. For each instance, the final label assigned to the instance was the label of the final comment in the conversation chain. Table 2 describes the dataset distribution with the validation split used in our experiments in section 5.2 .

4. Methodology

Our methodology primarily involves fine-tuning the transformer models pre-trained on a massive multilingual corpus. The following sections further describe the end-to-end approach used in our experiments.

4.1. Data Pre-processing

We first start by concatenating the tweet and its comments and replies, if they are present, to form the final text sequence. Our intuition behind this process is that this concatenation will help the model understand the context better, especially in those cases where the comment or reply may not be hateful but shows support for the hateful parent tweet. While concatenating the tweets, we insert a new separator token “[SENSEP]” between the tweet , comment, and the reply, to differentiate between one tweet and another. Post concatenation, we perform data cleaning by removing hashtags, emojis, URLs and mentions from the tweets. However, we do

not remove punctuation and numbers to preserve the syntactic and semantic coherence of the tweets.

Although the data is code-mixed, having both Hindi and English text, there were several instances where Hindi text is present in Roman script. Since our models are pretrained on a code-mixed corpus, it is essential to deal with Hindi text in the Roman script to make the whole dataset consistent for training purposes. Therefore, we perform transliteration using AI4Bharat library¹ to convert the Hindi text in Roman script to Devanagari script.

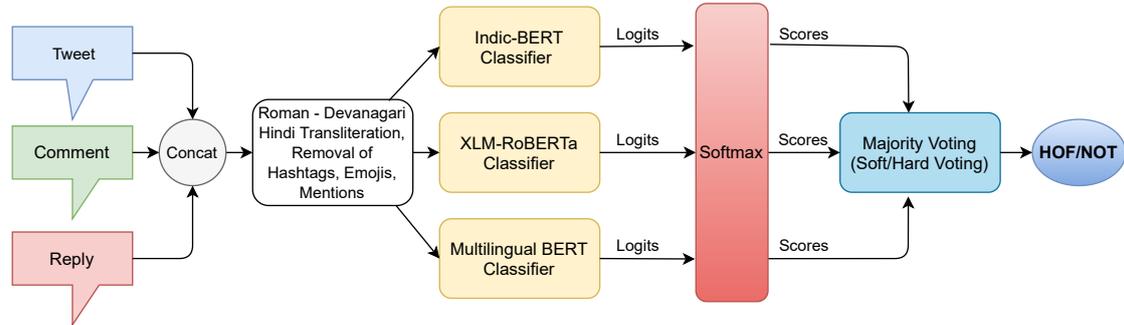


Figure 2: Overview of Ensemble Model. 'Concat' refers to concatenation operation. 'Scores' refer to normalized logits/ probability scores .

4.2. Baseline Approach : Fine-tuning Transformer Models

We perform experiments with three different types of transformer models, which are described below.

- **Indic-BERT** [40] is an ALBERT model pre-trained on a massive multilingual corpus having 12 Indic languages such as Assamese, Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu [40] . The multilingual corpus has about 9 billion tokens.
- **Multilingual BERT** (mBERT) is a BERT [41] model pretrained on Wikipedia data having over 100 languages with a masked language modeling objective.
- **XLM-Roberta** (XLM-R) [42] is a transformer-based masked language model pretrained on Common Crawl data having about 100 languages. It was proposed by Facebook and happened to be one of the best-performing transformer models for multilingual tasks.

On top of these pre-trained transformer models, we add a dropout followed by a fully connected layer of size two which takes in the transformer's CLS token's representation of size 768. The fully connected layer returns logits for the two classes, i.e., HOF and NOT, which are then passed to a softmax layer to predict the class of the input text. This is explained as a part of the ensemble model in figure 2 .

¹<https://pypi.org/project/ai4bharat-transliteration/>

4.3. Our Approach : Models Ensembling

We perform model ensembling on the top of our three fine-tuned models as explained in the section 4.2. Since we had three different transformer models fine-tuned for our task, we decided to combine the output of the three models using the majority voting system, i.e., Hard Voting and Soft Voting. We denote our two ensemble models as **Hard Voting Ensemble** and **Soft Voting Ensemble**. **Hard Voting Ensemble** model takes in the class predictions from each of the fine-tuned transformer models and selects the class with maximum votes. Similarly, **Soft Voting Ensemble** model takes in the class probabilities from each of the fine-tuned transformer models and sum the same class probabilities and selects the class having higher probabilities sum. Figure 2 shows the end-to-end model pipeline used in our experiment.

5. Experiments and Results

5.1. Experimental Setup

For fine-tuning our transformer models, we set a learning rate of $2e-5$ for all our experiments with AdamW [43] optimizer and a linear learning rate scheduler. We train our models with a batch size of 8. Our experiments use the Hugging-Face Transformers library [44] for fine-tuning all the pre-trained transformer models.

5.2. Evaluation with Validation data

In order to evaluate our models, we split the train dataset in an 80:20 ratio where 80% of the data is the new train data, and the rest of 20% becomes the validation data as shown in table 2. The split is done in random order while maintaining the same class ratio in both train and validation data. We train our models for ten epochs and keep track of the validation loss at each epoch. For reporting results on validation data, We use the model checkpoint corresponding to the epoch with minimum validation loss. At the same time, we make a note of that epoch for which we later train our model on the whole train dataset in 5.3.

Table 3

Results obtained on Validation split

Model	F1	Precision	Recall	Accuracy(%)
Indic-BERT	0.7150	0.7159	0.7154	71.51
Multilingual BERT	0.7438	0.7438	0.7439	74.39
XLM-RoBERTa	0.7262	0.7277	0.7268	72.64
Soft Voting Ensemble (Our approach)	0.7682	0.7687	0.7684	76.82
Hard Voting Ensemble (Our approach)	0.7621	0.7628	0.7624	76.21

Note : F1, Precision and Recall are Macro Scores

From Table 3 , we can see that our model ensembling approaches **Soft Voting Ensemble** and **Hard Voting Ensemble** outperform rest of the baseline transformer models with Macro F1 score of **.7682** and **.7621** respectively.

Table 4

Results obtained on Test Data

Submission	Model	F1	Precision	Recall	Accuracy(%)
NA	Indic-BERT	0.6811	0.6881	0.6821	68.47
NA	Multilingual BERT	0.7031	0.7031	0.7033	70.33
submit-1	XLM-RoBERTa	0.6970	0.6970	0.6970	69.73
submit-2	Soft Voting Ensemble (Our approach)	0.7223	0.7236	0.7222	72.32
submit-3	Hard Voting Ensemble (Our approach)	0.7253	0.7267	0.7251	72.62

Note : F1, Precision and Recall are Macro Scores

5.3. Evaluation with Test data

Following the results obtained in section 5.2 , we train our models on the whole train dataset for the best number of epochs identified through the minimum validation loss in section 5.2 . These epochs vary from 3 to 6 for each of the three transformer models discussed in 4.2. We submit the test results for three of our models as marked in table 4 which reports the results obtained on the test data. It can be inferred that the test results follow the same trend as was with the validation data. However, **Hard Voting Ensemble** with Macro F1 of **0.7253** is slightly better than **Soft Voting Ensemble** with Macro F1 score of **0.7223**. However, the overall performance of the model ensembling technique is better than the Indic-BERT, XLM-RoBERTa, and Multilingual BERT by a significant margin where the highest possible Macro F1 score for baseline models is **0.7031**.

6. Analysis

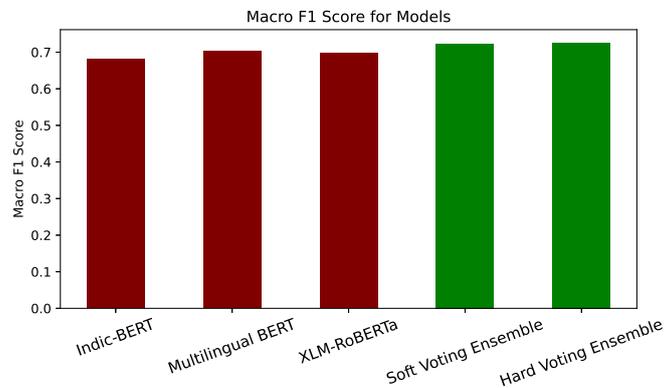
**Figure 3:** Macro F1 Scores on Test Data

Figure 3 compares the results in table 4. It can be noted that the Indic-BERT model has the lowest F1 score among all the baseline and ensemble models. **Soft Voting Ensemble** and **Hard Voting Ensemble** models yield better results than all the baseline transformer models.

However, merely having a better F1 score is not enough since it is crucial to understand where our approach fails and where it performs better than the baselines.

We start our error analysis with table 5 where we have shown the total number of misclassified samples from each class. In addition to it, it has the percentage of total samples misclassified from each class, which helps us develop a better understanding of the direction where models are making more mistakes. Figure 4 shows the detailed results of the performance of each model for samples of both classes. A closer look reveals that Multilingual BERT and XLM-RoBERTa misclassify almost equal number of samples from both classes HOF and NOT where the percentage ranges from 29.35% to 31.24%. However, the case is quite different for Indic-BERT, where the misclassification rate is very high for the NOT class, which is 40.27% compared to the misclassification rate of 23.30% for the HOF class. This could also be because Indic-BERT is trained on 12 Indian languages and has a less diverse pre-training corpus than the other two transformer models.

Table 5
Total Number and Percentage of Misclassified samples for HOF and NOT classes

Model	Misclassified HOF	% MR (HOF)	Misclassified NOT	% MR (NOT)
Indic-BERT	162	23.30	263	40.27
Multilingual BERT	207	29.78	193	29.55
XLM-RoBERTa	204	29.35	204	31.24
Soft Voting Ensemble	168	24.17	205	31.39
Hard Voting Ensemble	165	23.74	204	31.24

Note : % MR (HOF) and % MR (NOT) refer to percent misclassification rate for HOF and NOT classes respectively. % MR (class) is the percentage of total misclassified samples of the class out of the total samples of the class. These are calculated with total HOF samples count of 695 and total NOT samples count of 653 in test data.

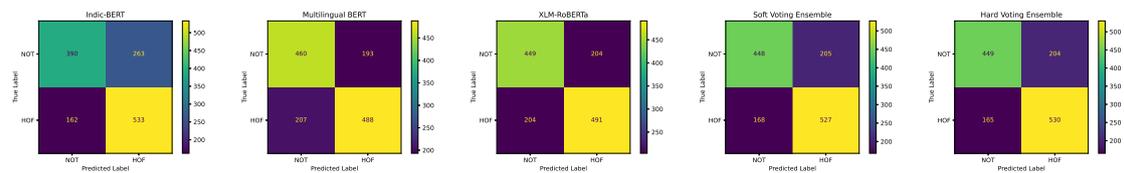


Figure 4: Confusion Matrix on Test Data for Baseline and Ensemble Models

As far as our ensemble models are concerned, we observe that the misclassification rate is very balanced for both classes. To compare it with the baseline transformer models, we can see that the ensemble models have the lowest misclassification rate for HOF class ranging between 23% to 24%, which was also the case with Indic-BERT, and similarly, the misclassification rate for NOT class is also close to the lowest among the baseline models. This suggests that model ensembling minimized the misclassification rate for both classes, and we can further conclude that a single model's mistake is likely to be corrected by the other two models in an ensemble

model. However, the ensemble models still make 7-8% more mistakes in identifying NOT class compared to HOF class.

7. Conclusion

In this paper, we deal with a novel problem of detecting hateful tweets in twitter conversations where the comment or reply might not be offensive and toxic but contributes to the hate associated with the parent tweet. We performed a thorough experimental analysis with state-of-the-art models such as XLM-RoBERTa, Indic-BERT, and Multilingual BERT to show that pre-trained multi-lingual transformer models can achieve decent performance on this task. We further demonstrate that this performance can be improved with model ensemble techniques such as Soft Voting and Hard Voting. As this problem is dealt with for the first time, there could be many ways to improve these numbers and build a more robust system by incorporating other factors such as emojis and hashtags, which may equally or partially contribute to the hatefulness of a tweet. Additionally, we aim to explore better architectures for taking into account the context of a comment like parent comment and replies to judge the nature of the comment. Author profiling also would be a potential area of research to detect implicit hate in conversations.

References

- [1] M. L. Williams, P. Burnap, A. Javed, H. Liu, S. Ozalp, Hate in the machine: anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime, *The British Journal of Criminology* 60 (2020) 93–117.
- [2] L. Dixon, J. Li, J. Sorensen, N. Thain, L. Vasserman, Measuring and mitigating unintended bias in text classification, in: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 67–73.
- [3] D. Borkan, L. Dixon, J. Sorensen, N. Thain, L. Vasserman, Nuanced metrics for measuring unintended bias with real data for text classification, in: *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 491–500.
- [4] 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.
- [5] 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.
- [6] R. Guerhazi, M. Hammami, A. B. Hamadou, Using a semi-automatic keyword dictionary for improving violent web site filtering, in: *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, 2007, pp. 337–344. doi:10.1109/SITIS.2007.137.
- [7] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, N. Bhamidipati, Hate speech

- detection with comment embeddings, in: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, Association for Computing Machinery, New York, NY, USA, 2015, p. 29–30. URL: <https://doi.org/10.1145/2740908.2742760>. doi:10.1145/2740908.2742760.
- [8] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759–760.
 - [9] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, N. Kourtellis, Large scale crowdsourcing and characterization of twitter abusive behavior, 2018. arXiv:1802.00393.
 - [10] S. Carta, A. Corrigan, R. Mulas, D. R. Recupero, R. Saia, A supervised multi-class multi-label word embeddings approach for toxic comment classification., in: KDIR, 2019, pp. 105–112.
 - [11] H. H. Saeed, K. Shahzad, F. Kamiran, Overlapping toxic sentiment classification using deep neural architectures, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 1361–1366.
 - [12] A. Vaidya, F. Mai, Y. Ning, Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 14, 2020, pp. 683–693.
 - [13] T. Tran, Y. Hu, C. Hu, K. Yen, F. Tan, K. Lee, S. Park, Habertor: An efficient and effective deep hatespeech detector, 2020. arXiv:2010.08865.
 - [14] H. Hitkul, R. R. Shah, P. Kumaraguru, S. Satoh, Maybe look closer? detecting trolling prone images on instagram, in: 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), IEEE, 2019, pp. 448–456.
 - [15] Hitkul, K. Aggarwal, P. Bamdev, D. Mahata, R. R. Shah, P. Kumaraguru, Trawling for trolling: A dataset, 2020. arXiv:2008.00525.
 - [16] S. Ghosh, S. Lepcha, S. Sakshi, R. R. Shah, Speech toxicity analysis: A new spoken language processing task, arXiv preprint arXiv:2110.07592 (2021).
 - [17] O. Kamal, A. Kumar, T. Vaidhya, Hostility detection in hindi leveraging pre-trained language models, 2021. arXiv:2101.05494.
 - [18] J. A. Leite, D. F. Silva, K. Bontcheva, C. Scarton, Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis, arXiv preprint arXiv:2010.04543 (2020).
 - [19] A. Saroj, S. Pal, An indian language social media collection for hate and offensive speech, in: Proceedings of the Workshop on Resources and Techniques for User and Author Profiling in Abusive Language, 2020, pp. 2–8.
 - [20] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, M. Sanguinetti, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 54–63. URL: <https://aclanthology.org/S19-2007>. doi:10.18653/v1/S19-2007.
 - [21] A. Ghosh Chowdhury, A. Didolkar, R. Sawhney, R. R. Shah, ARHNet - leveraging community interaction for detection of religious hate speech in Arabic, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Florence, Italy, 2019, pp. 273–280.

URL: <https://aclanthology.org/P19-2038>. doi:10.18653/v1/P19-2038.

- [22] 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, 2018, pp. 18–26.
- [23] R. Kapoor, Y. Kumar, K. Rajput, R. R. Shah, P. Kumaraguru, R. Zimmermann, Mind your language: Abuse and offense detection for code-switched languages, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 9951–9952.
- [24] P. Mathur, R. Sawhney, M. Ayyar, R. Shah, Did you offend me? classification of offensive tweets in hinglish language, in: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), 2018, pp. 138–148.
- [25] S. Chopra, R. Sawhney, P. Mathur, R. R. Shah, Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 386–393.
- [26] S. Kamble, A. Joshi, Hate speech detection from code-mixed hindi-english tweets using deep learning models, 2018. arXiv:1811.05145.
- [27] T. Ranasinghe, S. Gupte, M. Zampieri, I. Nwogu, Wlv-rit at hasoc-dravidian-codemix-fire2020: Offensive language identification in code-switched youtube comments, 2020. arXiv:2011.00559.
- [28] K. Yasaswini, K. Puranik, A. Hande, R. Priyadharshini, S. Thavareesan, B. R. Chakravarthi, IIIT@DravidianLangTech-EACL2021: Transfer learning for offensive language detection in Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 187–194. URL: <https://aclanthology.org/2021.dravidianlangtech-1.25>.
- [29] A. Baruah, K. A. Das, F. A. Barbhuiya, K. Dey, Iiitg-adbu@hasoc-dravidian-codemix-fire2020: Offensive content detection in code-mixed dravidian text, 2021. arXiv:2107.14336.
- [30] J. Pavlopoulos, L. Laugier, J. Sorensen, I. Androutsopoulos, Semeval-2021 task 5: Toxic spans detection, in: Proceedings of the 15th International Workshop on Semantic Evaluation, 2021.
- [31] S. Ghosh, S. Kumar, Cisco at semeval-2021 task 5: What’s toxic?: Leveraging transformers for multiple toxic span extraction from online comments, 2021. arXiv:2105.13959.
- [32] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, A. Mukherjee, Hatexplain: A benchmark dataset for explainable hate speech detection, 2020. arXiv:2012.10289.
- [33] P. Mishra, M. D. Tredici, H. Yannakoudakis, E. Shutova, Abusive language detection with graph convolutional networks, 2019. arXiv:1904.04073.
- [34] M. Das, P. Saha, R. Dutt, P. Goyal, A. Mukherjee, B. Mathew, You too brutus! trapping hateful users in social media: Challenges, solutions insights, 2021. arXiv:2108.00524.
- [35] J. Qian, M. ElSherief, E. Belding, W. Y. Wang, Hierarchical cvae for fine-grained hate speech classification, 2018. arXiv:1809.00088.
- [36] J. Pavlopoulos, J. Sorensen, L. Dixon, N. Thain, I. Androutsopoulos, Toxicity detection: Does context really matter?, 2020. arXiv:2006.00998.
- [37] M. Saveski, B. Roy, D. Roy, The structure of toxic conversations on twitter, 2021. arXiv:2105.11596.
- [38] J. Qian, A. Bethke, Y. Liu, E. Belding, W. Y. Wang, A benchmark dataset for learning to intervene in online hate speech, arXiv preprint arXiv:1909.04251 (2019).

- [39] H. Almerexhi, H. Kwak, J. Salminen, B. J. Jansen, Are these comments triggering? predicting triggers of toxicity in online discussions, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020. URL: <https://doi.org/10.1145/3366423.3380074>. doi:10.1145/3366423.3380074.
- [40] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, P. Kumar, IndicNLP Suite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages, in: Findings of EMNLP, 2020.
- [41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.
- [42] 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, 2020. arXiv:1911.02116.
- [43] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. arXiv:1711.05101.
- [44] 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. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. 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://www.aclweb.org/anthology/2020.emnlp-demos.6>.