Fine-tuning Pre-Trained Transformer based model for Hate Speech and Offensive Content Identification in English, Indo-Aryan and Code-Mixed (English-Hindi) languages

Supriya Chanda[1], S Ujjwal[1], Shayak Das[1] and Sukomal Pal[1]

[1]*Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, INDIA, 221005*

Abstract

Hate Speech and Offensive Content Identification is one of the most challenging problem in the natural language processing field, being imposed by the rising presence of this phenomenon in online social media. This paper describes our Transformer-based solutions for identifying offensive language on Twitter in three languages (i.e., English, Hindi, and Marathi) and one code mixed (English-Hindi) language, which was employed in Subtask 1A, Subtask 1B and Subtask 2 of the HASOC 2021 shared task. Finally, the highest-scoring models were used for our submissions in the competition, which ranked our IRLab@IITBHU team 16th of 56, 18th of 37, 13th of 34, 7th of 24, 12th of 25 and 6th of 16 for English Subtask 1A, English Subtask 1B, Hindi Subtask 1A, Hindi Subtask 1B, Marathi Subtask 1A, and English-Hindi Code-Mix Subtask 2 respectively.

Keywords

Hate Speech, Offensive Language, Social Media, Hindi, Marathi, Code-Mixed, Multilingual BERT

1. Introduction

With the ease of access to the internet these days, a large number of people from various ethnic and educational backgrounds interact on social media. Individuals and groups are demonized by using hateful and insulting language for communicating their ideas and disapproval. User-generated content on social media, especially, has been a hotbed of harsh language and hate speech. As a result, people's morale is lowered, and mental anguish and trauma are inevitable. As a response, information extraction from social media data and possible offensive language identification are considered essential. There are regulations against abusive language on almost all social networking sites, but identifying them might be difficult. It is not possible to keep an eye on the situation manually or with a static set of rules. Using natural language processing (NLP) tools to search for offensive content in textual data is possible because hate speech and offensive language belong to natural language.

For a country like India, people tend to use regional language for texting or tweeting. Around

FIRE'21: Forum for Information Retrieval Evaluation, Dec 13–17, 2021

supriyachanda.rs.cse18@itbhu.ac.in (S. Chanda); sujjwal.cse18@itbhu.ac.in (S. Ujjwal); shayakdas.cse18@itbhu.ac.in (S. Das); spal.cse@itbhu.ac.in (S. Pal)
0000-0002-6344-8772 (S. Chanda)

CEUR Workshop Proceedings (CEUR-WS.org)

half of the population speaks Hindi[1]. Grover et al. (2017) [1] studied English-Hindi code-switching and swearing pattern on social networks for multilingual users. They tested the swearing behaviour of multilingual users on a large scale using monolingual Hindi and English tweets as well as code-switched tweets from Indian users. These findings revealed strong language preference among bilinguals, although profanity and swearing can be powerful motivators for code-switching.

The Hate Speech and Offensive Content Identification (HASOC) shared tasks of 2021 focused on Indo-Aryan languages in three different languages: English, Hindi, and Marathi. The shared tasks have two sub-tasks: Subtask-1 and Subtask-2. Again Subtask-1 has two parts: Subtask-1A, a coarse-grained binary classification, and Subtask-1B, a fine-grained classification. The main focus of Subtask-2 is to identify Conversational Hate-Speech in Code-Mixed Languages (ICHCL). In a conversational thread, the comments sometimes do not express any sentiment by themselves, but it is expressed in the context of the main post or parent comments. However, in our study, we take all comments as a standalone tweet. The Subtasks-2 dataset contains English, Hindi, and code-mixed Hindi tweets. Therefore, it gave us an opportunity to address the multilingual issues associated with social media posts. To solve this, we used publically accessible pre-trained transformer-based neural network (BERT) models, which allow for fine-tuning for specific tasks. In addition to this, its multilingual feature allows us to analyze sentiment for the comments with multiple language words and sentences. We participated in both Subtasks, and all three languages, and one Code-Mixed language.

1.1. HASOC SubTask

The aim of HASOC 2021[2] was to provide a testbed facilitating testing of systems that can detect hate speech and offensive content automatically from social media posts. There were three subtasks in HASOC. They are described below with examples in Table 1.

- **Subtask A: Hate and Offensive language Identification**
  Subtask A is a coarse-grained binary classification that classifies tweets into two main categories.
  - **Non Hate-Offensive (NOT)** - This post contains no hate speech, profanity, or objectionable content.
  - **Hate and Offensive (HOF)** - This post contains information that is hateful, offensive, and vulgar.
- **Subtask B: Type of Hate and Offensive post**
  Subtask B is a classification task with multiple classes. After a post is categorised as HOF in Subtask A, it is further categorised into one of three types:
  - **Hate speech (HATE):** - The post is directed towards a group or a member of a group who is aware that he or she is a member of that group. Any comments that are hostile due to their political beliefs, sexual orientation, gender, socioeconomic standing, health condition, or something similar.

[1] https://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/Statement4.htm

- **Offensive (OFFN):** - The post contains offensive contents like dehumanizing, insulting an individual or threatening someone.
- **Profane (PRFN):** - The post contains swearwords. (Fuck etc.)

Table 1
Example tweets from the HASOC2021 dataset for all classes

Language	**Sample tweet from the class**	**SubTask-1**	
		A	**B**
English	@Wari_gay Can't expect God to do all the work	NOT	NONE
	@bananapixelsuk That's why the whole thing is a load of crap. Corporate bollocks.	HOF	HATE
	@ndtv Shameless PM. What else can we say? #ShameOnModi #Resign_PM_Modi #ResignPMmodi	HOF	OFFN
	@UtdEIIis Really like how this list started with Dan Shitbag.	HOF	PRFN
Hindi	#किसानों_का_मोदी_को_धोबीपटका #ResignPMmodi https://t.co/nKTi-ocjjMl	NOT	NONE
	@anushka_s2 मूर्ख लड़की	HOF	OFFN
	सवाल यह नही कि वो मुझे वेश्या कहता है सवाल तो यह है कि मुझे वेश्या बनाया किसने ? -नवनीत	HOF	HATE
	धवन मदरचोद ज़िंदा है मर गया	HOF	PRFN
Marathi	तिने तोंड उघडलं ह्यांनी नाक दाबायला सुरुवात केली.	HOF	-
	आयुष्य खूप सोपं आहे आपण ते विनाकारण अवघड करुन ठेवतो...	NOT	-
Language	**Sample tweet from the class**	**SubTask-2**	
English-Hindi	@MovidMukt_India @srivatsayb Kaash tere sochne k hisab se duniya chalti	HOF	-
	@ashokepandit Sir, we will do kafi ninda only or book as per law?	NONE	-

The remaining of the paper is structured as follows. In Section 2, we briefly outline some previous attempts. The dataset description are presented in section 3. Our computational methods, models description and evaluation methodology are presented in Section 4, followed by results and discussion in Section 5 and conclusion in Section 6.

2. Related Work

Over the last few years, there have been several studies on computational method to identify hate and offensive speech. Some prior works have studied blogs, micro-blogs, and social networks like twitter data [3], [4], [5], [6] and [7] as well as Facebook post and Wikipedia comments.

A couple of studies like [8], [9], [4] and [10] have been published where they focused on detecting whether a post contains hate speech or not, only two-way classification. Dinakar et al. [11] proposed an idea where they classify the posts based on the frequency of offensive or socially non-acceptable words. Machine learning algorithms using TF-IDF characteristics are being utilised in social media to identify and categorise hate speech and offensive language [12].

Because of the scarcity of relevant corpora, the vast majority of studies on abusive language have focused on English data. However, a few research works have recently looked into abusive language detection in different languages. Mubarak et al. [13] deal with abusive language detection on Arabic social media, whereas Su et al. [14] offer a method for detecting and reverting profanity in Chinese. Hate speech and abusive language datasets for German and Slovene have recently been annotated by Ross et al. [15] and Fiser et al. [16] respectively, which paved the way for future work in languages other than English. Also many workshops have been organised to identify hate speech. The SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval 2019) [17] was the first competition towards detecting offensive language in social media (Twitter) only on English language. The SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020) [18] organised for the same proposes with four other languages Arabic, Danish, Greek, and Turkish. Germeval Task 2, 2019 [2] - Shared Task on the Identification of Offensive Language, Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC 2019) [3], (HASOC 2020) [4] try to identify Hate speech on English, Hindi and German language.

There have been some work exploring different aspects of offensive content like *abusive language* ([10], [13]), *cyber-aggression* [7], *cyber-bullying* [19] and *toxic comments or hate speech* ([8], [6], [9]).

3. Dataset

The HASOC 2021 dataset[5] [20] was sampled from Twitter for multilingual research with three languages together, i.e., English [21], Hindi [21], Marathi [22] and one Code-Mix (English-Hindi) [23] language. The corpus collection and class distribution is shown in Table 2.

4. Methodology

4.1. Preprocessing

The primary preprocessing phase is carried out using the BERT-specific tokenizer, which divides a phrase into tokens in a WordPiece way. It operates by dividing words into their complete forms (e.g., one word becomes one token) or into word pieces (e.g., one word can be broken down into many tokens). As a example `snowboarding` is a word, which will be tokenize by WordPiece tokenizer like [`snow`] [`##board`] [`##ing`].

The majority of the data collected from Twitter contains Hashtags and emoticons. As a result, two Twitter-specific stages were completed initially.

[2] https://projects.fzai.h-da.de/iggsa/

[3] https://hasocfire.github.io/hasoc/2019/index.html

[4] https://hasocfire.github.io/hasoc/2020/index.html

[5] https://hasocfire.github.io/hasoc/2021/dataset.html

Table 2
Statistical overview of the Training Data and Test Data for determining the final results

			Subtask-1A		Subtask-1B		
Data	Language	# Sentences	NOT	HOF	HATE	OFFN	PRFN
Train	English	3843	2501	1342	683	622	1196
	Hindi	4594	3161	1433	566	654	213
	Marathi	1874	1205	669	-	-	-
Test	English	1281	483	798	224	195	379
	Hindi	1532	1027	505	215	216	74
	Marathi	625	418	207	-	-	-
			Subtask-2				
Data	Language	# Sentences	NONE	HOF			
Train	Hindi-English	5740	2899	2841	-	-	-
Test	Hindi-English	1348	653	695	-	-	-

- Using the `demoji` and `ekphrasis` Python package, replace the emoticons with the equivalent textual representation.
- Normalizing hashtags (for example, "#IndiansDyingModiEnjoying" is segmented into "Indians", "Dying", "Modi", and "Enjoying").

4.2. Implementation

Each subtask can be represented as a text classification issue. Our submission models were developed by fine-tuning a pre-trained language model on shared task data. Because of its recent success and public availability in several languages, we selected BERT [24] as our pre-trained language model. After performing preprocessing steps, we experimented with the bert-base-cased and bert-base-uncased models for both subtasks of the English language. In addition, we submitted a run without performing any preprocessing procedures. We tried with the bert-base-multilingual-cased model for both subtasks of the Hindi language. We applied the same bert-multilingual-cased model for the Marathi subtask. We made use of the BERT implementation included in pytorch-transfomers[6] library. Figure 1 demonstrates our fine-tuned model. On our dataset, we trained the full pre-trained model and fed the result to a softmax layer. The error is back-propagated through the entire architecture in this scenario, and the model's pre-trained weights were adjusted depending on the new dataset. The complete model was fine-tuned.

[6] https://github.com/huggingface/transformers

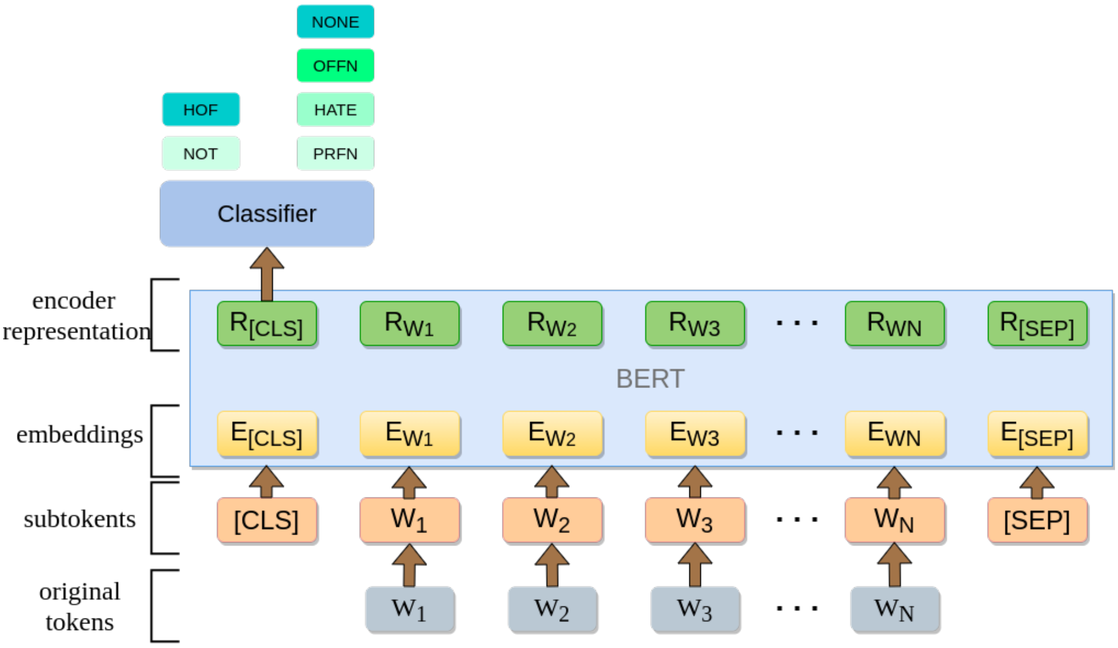


Figure 1: BERT model architecture for sequence classification

In the model described in Figure 1, the input is a sequence of words representing a sentence. The subtokens are generated by appending special tokens, CLS at the beginning and SEP at the end. This is then fed into the BERT model, which produces the embeddings for each word R_{Wi}, and the R_{CLS} vector corresponding to the CLS token for classification. BERT employs Transformer, an attention mechanism that learns contextual associations between words (or sub-words) in a text. In its basic form, the transformer includes two mechanisms: an encoder that reads the text input and a decoder that provides a job forecast. Because BERT's goal is to build a language model, just the encoder approach is necessary. The R_{CLS} vector is then passed through a neural network-based classifier, which gives us the probability distribution of the tokens, thereby corresponding to each class. The number of classes depends on the subproblem at hand.

HuggingFace's transformers library was leveraged for the implementation. HuggingFace transformers is a Python library that provides pre-trained and customizable transformer models that may be used for a range of NLP tasks. It includes the pre-trained and multilingual BERT models, as well as alternative models suited for downstream tasks. We employ the PyTorch library, which enables GPU processing, as the implementation environment. Google Colab was used to run the BERT models. Based on our experiments, we trained our classifier with a batch size of 32 for 5 to 10 epochs. The dropout value is set to 0.1, and the AdamW optimizer with a learning rate of 2e-5 is applied. For tokenization, we applied the hugging face transformers' pre-trained BERT tokenizer. During finetuning and sequence classification, we utilized the HuggingFace library's BertForSequenceClassification module. We have submitted all the different submissions for each subtask. The descriptions of all the runs are following.

Table 3
Evaluation results on test data and rank list (Submission number in bracket)

Language	Subtask	Team Name	Macro F_1 score	Rank
English	1-A	NLP-CIC	0.8305	1 / 56
		IRLab@IITBHU (1)	0.7579	-
		IRLab@IITBHU (2)	0.7581	-
		IRLab@IITBHU (3)	0.7812	-
		IRLab@IITBHU (4)	0.7886	-
		IRLab@IITBHU (5)	**0.7976**	16 / 56
	1-B	NLP-CIC	0.6657	1 / 37
		IRLab@IITBHU (1)	**0.6093**	18 / 37
Hindi	1-A	t1	0.7825	1 / 34
		IRLab@IITBHU (1)	0.7471	-
		IRLab@IITBHU (2)	0.7440	-
		IRLab@IITBHU (3)	**0.7547**	13 / 34
	1-B	NeuralSpace	0.5603	1 / 24
		IRLab@IITBHU (1)	0.4199	-
		IRLab@IITBHU (2)	**0.5127**	7 / 24
Marathi	1-A	WLV-RIT	0.9144	1 / 25
		IRLab@IITBHU (1)	**0.8545**	12 / 25
		IRLab@IITBHU (2)	0.8410	-
English-Hindi Code-Mix	2	MIDAS-IIITD	0.7253	1 / 16
		IRLab@IITBHU (1)	**0.6795**	6 / 16

1. **ENSA_submission_1:** BERT multilingual cased (mBERT), 20 epochs without replacing emojis and hashtags, Maximum sequence length of 128 tokens, and batch size of 32. (Macro F1: 0.7579)
2. **ENSA_submission_2:** mBERT, 20 epochs using emoji and hashtag substitution, Maximum sequence length of 128 tokens, and batch size of 32. (Macro F1: 0.7581)
3. **ENSA_submission_3:** mBERT, 25 epochs using emoji and hashtag substitution, replacing commonly occuring short-forms like (it's->it is, don't->do not, hahaha->ha etc.), Maximum sequence length of 128 tokens, and batch size of 32. (Macro F1: 0.7812)
4. **ENSA_submission_4:** BERT Large Cased, 25 epochs using emoji, hashtags substitution, Maximum sequence length of 128 tokens, and batch size of 32.(Macro F1: 0.7886)
5. **ENSA_submission_5:** BERT Large Cased, 25 epochs using emoji and hashtags substitution, Maximum sequence length of 128 tokens, and batch size of 16 (Macro F1: 0.7976)
6. **ENSB_submission_1:** BERT Large Cased, 20 epochs using emoji and hashtags substitution, Maximum sequence length of 128 tokens, and batch size of 16 (Macro F1: 0.6093)
7. **HISA_submission_1:** mBERT, 25 without preprocessing, Maximum sequence length of 128 tokens, and batch size of 32 (Macro F1: 0.7471)

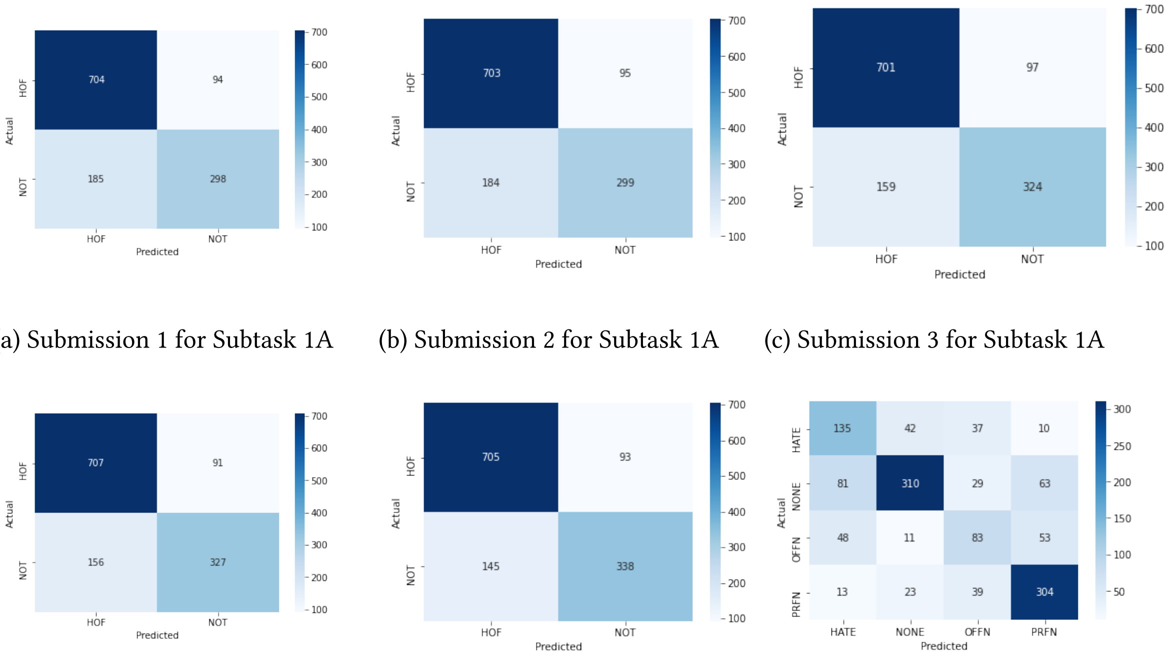
8. **HISA_submission_2:** mBERT, 25 epochs, Maximum sequence length of 256 tokens, using hashtag substitution, and batch size of 16 (Macro F1: 0.7440)
9. **HISA_submission_3:** mBERT, 25 epochs, using emoji and hashtag substitution, Maximum sequence length of 128 tokens, and batch size of 32. (Macro F1: 0.7547)
10. **HISB_submission_1:** mBERT, 25 epochs and without using emoji and hashtag substitution, Maximum sequence length of 256 tokens, and batch size of 32 (Macro F1: 0.4199)
11. **HISB_submission_2:** mBERT, 25 epochs and using emoji and hashtag substitution, Maximum sequence length of 256 tokens, and batch size of 16 (Macro F1: 0.5127)
12. **MRSA_submission_1:** mBERT, 15 epochs, without using preprocessing of emoji and hashtags substitution, Maximum sequence length of 128 tokens, and batch size of 32 (Macro F1: 0.8410)
13. **MRSA_submission_2:** mBERT, 15 epochs, using preprocessing of emoji and hashtags substitution, Maximum sequence length of 128 tokens, and batch size of 32 (Macro F1: 0.8545)
14. **CM_submission_1:** Mbert, 15 epochs, using preprocessing of emoji and hashtags substitution, Maximum sequence length of 256 tokens, and batch size of 16 (Macro F1: 0.6795)

5. Results and Discussion

We validated our model on the training and development sets since we lacked test labels. As our submission for each subtask, we chose the top models from each evaluation. Every system is evaluated using a Macro F_1 score. The overall system's macro F_1 score is the average of the different classes' $F1$ scores. Table 3 shows the best performing team and our official performances on the test data as shared by the organizers vis-a-vis the best performing team for all shared tasks of English, Hindi, Marathi, and code mixed Hindi-English language pair.

For the binary classification, the best-performed model for English subtask-1A was bert-large-cased with preprocessed data (Submission 5). For the system constraints, we took the maximum sequence length of 128 tokens for few sentences whose tokenized length was more than 128. So, we had to truncate it; that could be a reason for some low performance. The best-performed model for Hindi subtask-1A was bert-base-multilingual-cased with preprocessed data. Here also we had to truncate the sequence length up to 256. Although the model gives a comparative score, some of the NOT are still misclassified as HOF. The probable reason could be normalizing the Hashtags, like ResignModi to Resign Modi, which is classified as an attack towards a person. It is possible that the occurance of any curse words or hate words biases the model towards predicting the speech as HOF. The overall meaning of the sentence may still be non-hatred and this is very hard to deduce and requires the overall context to be discovered. Furthermore, for Marathi, preprocessing does not work as expected. It also misclassifies some NOT as HOF. It can be seen in subfigures 4(d) for the multiclass classification on Hindi language submission 1 that the model could not predict PRFN class.

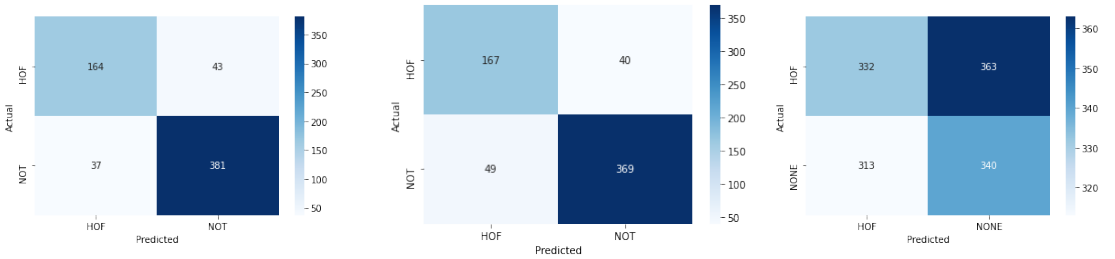
Table 4 shows some of the situations that our best model identified as inaccurate predictions. The expected sentiment, as provided in the gold standard dataset, is compared to the ones predicted by our algorithm in the table's Gold column. It seems that our predicted sentiment was correct.



(a) Submission 1 for Subtask 1A (b) Submission 2 for Subtask 1A (c) Submission 3 for Subtask 1A

(d) Submission 4 for Subtask 1A (e) Submission 5 for Subtask 1A (f) Submission 1 for Subtask 1B

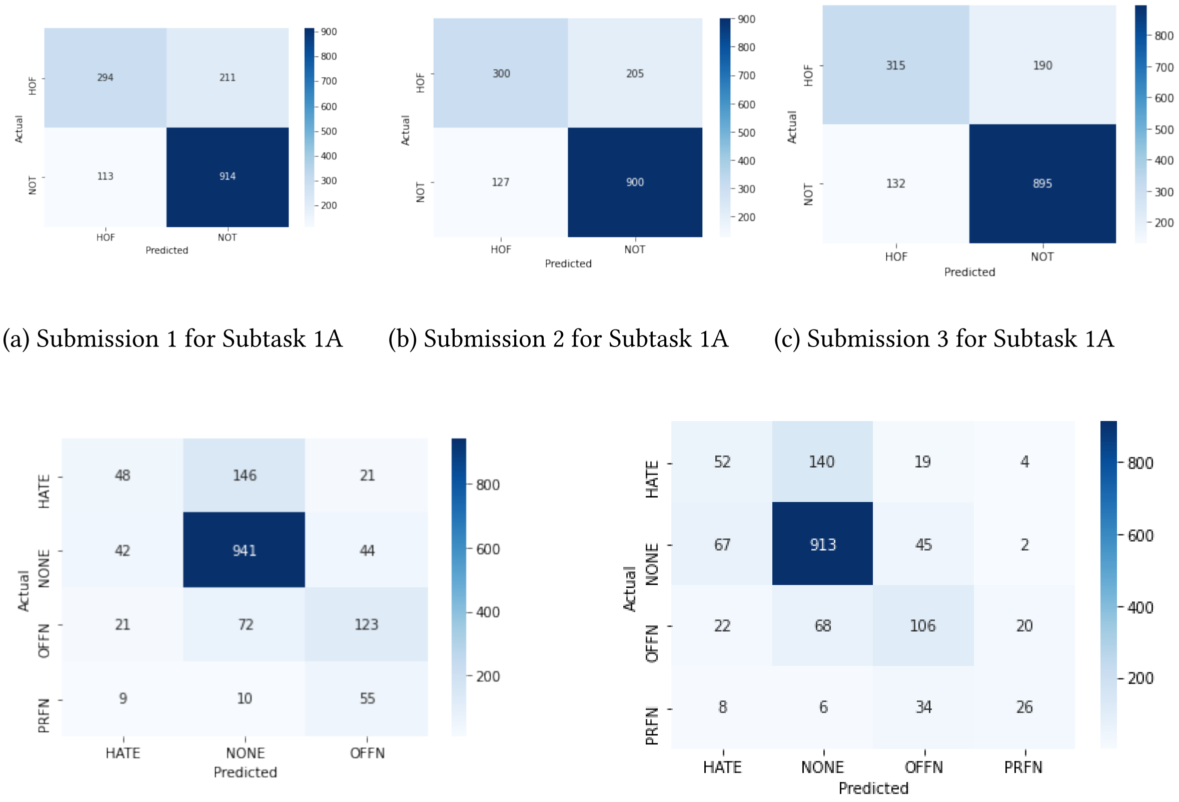
Figure 2: Confusion matrix on the given test data for the English language



(a) Submission 1 for Subtask 1A (b) Submission 2 for Subtask 1A (c) Submission 1 for Subtask 2

Figure 3: Confusion matrix on the given test data for the Marathi language and CodeMix English-Hindi language

Figures 2, 4 and 3 demonstrate the confusion matrix of the BERT model for subtasks 1A, 1B for the English, Hindi, and Marathi datasets, and 2 for the code mixed English-Hindi dataset. We submitted several number of submissions based on preprocessing procedures, batch sizes, and BERT model types.



(a) Submission 1 for Subtask 1A (b) Submission 2 for Subtask 1A (c) Submission 3 for Subtask 1A

(d) Submission 1 for Subtask 1B (e) Submission 2 for Subtask 1B

Figure 4: Confusion matrix on the given test data for the Hindi language

Table 4
Error Analysis

Sample Tweets from dataset	Gold	Predicted
Saw her shag rug and said ““I can wear that””	HOF	NOT
@bosco_rosco Mate I’m the life and soul of them because I’m not a twat.	NOT	HOF
Just had a phone call from the NHS National immunisation recall centre wanting to discuss my ““ #CovidVaccine plans”” - what the heck are they doing with my phone number ???	HOF	NOT

6. Conclusion

In this paper, we have presented the system submitted by the IRLab@IITBHU team to the HASOC 2021 - Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages shared task at FIRE 2021. Our system is based on fine-tuning monolingual and multilingual transformer networks to categorize social media postings in three distinct languages and an English-Hindi code mixed language for hate speech, offensive, and objectionable content. We have shown from the overview paper of the HASOC track at FIRE 2020 that the best results are achieved with state-of-the-art transformer models. Pre-trained bi-directional encoder representations using transformers (BERT) outperform all the traditional machine learning models. Thatswhy we have used only BERT model with some pre-processing. In Subtask 2: Identification of Conversational Hate-Speech in Code-Mixed Languages (ICHCL), we take all comments as a standalone tweet. In the future, we will like to solve this subtask using a graph.

7. Acknowledgements

We'd want to express our gratitude to the HASOC organisers for organising this interesting shared task and for immediately responding to all of our questions. We'd also like to express our gratitude to the anonymous reviewers for their insightful comments.

References

[1] J. Grover, P. Agarwal, A. Sharma, M. Sikka, K. Rudra, M. Choudhury, I may talk in english but gaali toh hindi mein hi denge: A study of english-hindi code-switching and swearing pattern on social networks, IEEE, 2017.

[2] S. Modha, T. Mandl, P. Majumder, D. Patel, Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages, in: FIRE '19, 2019.

[3] J.-M. Xu, K.-S. Jun, X. Zhu, A. Bellmore, Learning from bullying traces in social media, in: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT '12, Association for Computational Linguistics, USA, 2012, p. 656–666.

[4] P. Burnap, M. L. Williams, Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making, Policy & Internet 7 (2015) 223–242. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/poi3.85. doi:10.1002/poi3.85. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/poi3.85.

[5] M. Wiegand, M. Siegel, J. Ruppenhofer, Overview of the germeval 2018 shared task on the identification of offensive language, Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018, Austrian Academy of Sciences, Vienna, Austria, 2018, pp. 1 – 10. URL: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84935.

[6] T. Davidson, D. Warmsley, M. W. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, CoRR abs/1703.04009 (2017). URL: http://arxiv.org/abs/1703.04009. arXiv:1703.04009.

[7] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Benchmarking aggression identification in social media, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 1–11. URL: https://www.aclweb.org/anthology/W18-4401.

[8] 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.

[9] I. Kwok, Y. Wang, Locate the hate: Detecting tweets against blacks, in: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, AAAI Press, 2013, p. 1621–1622.

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

[11] K. Dinakar, R. Reichart, H. Lieberman, Modeling the detection of textual cyberbullying, in: The Social Mobile Web, Papers from the 2011 ICWSM Workshop, Barcelona, Catalonia, Spain, July 21, 2011, volume WS-11-02 of *AAAI Workshops*, AAAI, 2011. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/3841.

[12] A. Saroj, S. Chanda, S. Pal, IRlab@IITV at SemEval-2020 task 12: Multilingual offensive language identification in social media using SVM, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 2012–2016. URL: https://aclanthology.org/2020.semeval-1.265.

[13] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on Arabic social media, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 52–56. URL: https://www.aclweb.org/anthology/W17-3008. doi:10.18653/v1/W17-3008.

[14] H.-P. Su, Z.-J. Huang, H.-T. Chang, C.-J. Lin, Rephrasing profanity in Chinese text, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 18–24. URL: https://www.aclweb.org/anthology/W17-3003. doi:10.18653/v1/W17-3003.

[15] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, M. Wojatzki, Measuring the reliability of hate speech annotations: The case of the european refugee crisis, CoRR abs/1701.08118 (2017). URL: http://arxiv.org/abs/1701.08118. arXiv:1701.08118.

[16] D. Fišer, T. Erjavec, N. Ljubešić, Legal framework, dataset and annotation schema for socially unacceptable online discourse practices in Slovene, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 46–51. URL: https://www.aclweb.org/anthology/W17-3007. doi:10.18653/v1/W17-3007.

[17] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 75–86.

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

[19] M. Dadvar, D. Trieschnigg, R. Ordelman, F. de Jong, Improving cyberbullying detection with user context, in: P. Serdyukov, P. Braslavski, S. O. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, I. Segalovich, E. Yilmaz (Eds.), Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 693–696.

[20] 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.

[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] 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.