Offensive Language Identification Using Hindi-English Code-Mixed Tweets, and Code-Mixed Data Augmentation

Md Saroar Jahan[1], Mourad Oussalah[1], Jhuma kabir Mim[2] and Mominul Islam[3]

[1]*University of Oulu, Faculty of Information Tech., CMVS, PO Box 4500, Oulu 90014, FINLAND*
[2]*LUT Univerity, Dept of Computational Engineering 53850 Lappeenranta, FINLAND*
[3]*Daffodil International University, Dhaka 1207, BANGLADESH*

Abstract

The Code-mixed text classification is challenging due to the lack of code-mixed labeled datasets and the non-existence of pre-trained models. This paper presents the HASOC-2021 offensive language identification results and main findings on code-mixed (Hindi-English) Subtask2. In this work, we have proposed a new method of code-mixed data augmentation using synonym replacement of Hindi and English words using WordNet, and phonetics conversion of Hinglish (Hindi-English) words. We used a 5.7k pre-annotated HASOC-2021 code-mixed dataset for training and data augmentation. The proposal's feasibility was tested with a Logistic Regression (LR) used as a baseline, Convolutional Neural Network (CNN), and BERT with and without data augmentation. The research outcomes were promising and yields almost 3% increase of classifier accuracy and F1 scores as compared to baseline. Our official submission showed a 66.56% F1 score and ranked 8th position in the competition.

Keywords

Code-mixed Hindi-Englsih, Offensive language identification, Code-mixed Data Augmentation.

1. Introduction

Social media is a popular and easiest way to express openly and communicate with others online. Unfortunately, it also provides the means for distributing abusive and aggressive content such as sexism, racism, politics, cyberbullying, and blackmailing. Nockleby [1] stated that 'hate speech disparages a person or group based on some characteristics such as race, color, and ethnicity'. Now its become a challenge; offensive language is ubiquitous in social media; scholars and organizations have been focusing on developing an approach that can identify hate speech or abusive languages and flag them for human restraints or elimination [2]. Prior work has studied offensive language detection in Twitter [3, 4, 5], Wikipedia comments and Facebook posts [6], FromSpring posts [7], youtube Dinakar et al.[8], News article [9] and AskFm post [5, 10].

However, the challenges become critical when social posts are written with Code-Mixed (CM) language. The Code-mixing, which is the phenomenon of mixing words from two languages

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

mjahan18@edu.oulu.fi (M. S. Jahan); mourad.oussalah@oulu.fi (M. Oussalah); Jhuma.mim@student.lut.fi (J. k. Mim); mominul15-11992@diu.edu.bd (. M. Islam)

CEUR Workshop Proceedings (CEUR-WS.org)

in a sentence, is getting increasingly commonplace in several bilingual communities, which renders the automatic making detection task more challenging [11].

Many works focused on deep learning-based models to identify the aggressive language in social media texts. For instance, Agrawal and Awekar[12] investigated how learning-based models can capture more dispersed features on various platforms and topics. Bu and Cho[13] provided a hybrid deep learning model that combines CNN and Long-term Recurrent Convolutional Networks (LRCN) to detect offensive in Social Networking Service (SNS) comments. A character-level CNN model with shortcuts was proposed by Lu et al. [14]. In addition, Rosa et al.[15] compared three different deep learning approaches, trained from three different sources for multiple category textual offensive detection.

The Natural Language Processing (NLP) community organized several initiatives and seminars to cope with the scenarios above to stimulate research on hateful speech and offensive content in social media, such as Semeval-2019[16] and 2020[17], HASOC-2019 [18] and 2020[19]. From previous work, BERT model outperformed most of other state-of-the-art models. The rise of BERT is a striking trend that testifies of its popularity in the hate speech detection community (38% share of deep-learning models) in the past five years [20]

In this year, 2021, for the first time, HASOC provides Subtask2[21], which offers a multilingual offensive code-mixed language identification task in social media-Twitter. In this contest, our team participated to SubTask2 for Hindi-English code-mixed identification of Hate/offensive Tweets. We have used the HASOC-2021 shared dataset for training and validation. Since the absence of large-scale code-mixed labeled corpus, an intuitive approach is to seek an appropriate data augmentation strategy. It is challenging to obtain universal transformation rules in natural languages that assure the quality of the produced data and easy automated application procedures in various domains. A common approach for a such a transformation is to replace words with their synonyms selected from a handcrafted ontology such as WordNet [22]. Another synonym replacement approach is based on pre-trained word embeddings such as GloVe, FastText, Sent2Vec, etc. The nearest neighbor words in the embedding space are a replacement for some word in the sentence or word similarity calculation [23]. Back-translation and paraphrasing augmentation also shown higher accuracy for supervised learning [24]. A recent trend is contextual data augmentation that stochastically replaces words with other words predicted by a bi-directional language model at the corresponding word positions. However, code-mixed datasets are an amalgamation of multilingual tokens, making them limited to existing augmentation methods. For example, contextual augmentation with the transformer model requires pre-trained models, and to the best of our knowledge, pre-trained models for code-mixed are still scarce. Furthermore, many tokes are written native phonetically but in a different language; therefore, it is impossible to use synonym replacement without further conversion. We focused on code-mixed data augmentation to overcome these challenges by using synonym replacement with WordNet, phonetics conversion, translation, and back-translation for relevant tweets. The paper posits some main contributions as follow:

- A data augmentation scheme has been put forward that has not been experimented for code-mixed dataset.
- We developed a new python library for data augmentation, which is the end product of our experiment, and would be released under an open-sourced license for the research

community [1].

- We constructed a newly extended code-mixed (Hindi-English) dataset and released it publicly.

The paper is structured as follows. Section 2 describes our methodology, consisting of dataset annotation schema, preprocessing, dataset augmentation, and classifier architectures, including the machine learning models and the associated feature engineering. Section 3 details and comments on our experimental result. In Section 4, an error analysis task of our best models is performed. Finally, conclusive statements and potential future work are drawn in the conclusion section.

2. Methodology

The overall experimentation methodology includes a four-stage process: (i) data collection and preprocessing, (ii) Code-mix data augmentation, (iii) ML model setup, (iv) results comparison before and after augmentation, and (iv) error analysis.

The experiment environment will be the same for all experiments (e.g., data preprocessing, data augmentation, machine learning (ML) architecture, test data, and error analysis).

2.1. Datasets

To train our models and compare our results, we used the Code-mixed twitter dataset from HASOC-2021. The dataset tweet consists of eight different controversial topics: (i) Twitter Conflicts with the Indian Government on new IT rules. (ii) Casteism controversy in India (ii) Charlie Hebdo posts on Hinduism (iv) The Covid-19 crisis in India 2021 (v) Indian Politics (vi)The Israel-Palestine conflict in 2021 (vii) Religious controversies in India and (viii) The coronavirus controversy.

The HASOC task organizer already annotated datasets for Subtask2. For hate/offensive posts, it is labeled as HOF, and for non-hate/offensive, it is labeled as NONE. The Code-mixed dataset consists of approximately 5740 training data from Twitter tweets, re-tweets, comments, and replies; among 2841(45%) are offensive, and 2899(55%) are not-offensive. Table 1 shows the example of annotated code-mixed.

1. (NONE) Non-Hate/Offensive - This post does not contain any Hate speech, profane, offensive content.
2. (HOF) Hate and Offensive - This post contains Hate, offensive, and profane content.

Table 1 shows example of dataset content. At first glance, the handling may seem pretty straightforward; however, in practice, there are several challenges associated with the token type, sentence pattern, and annotations. First, we shall mainly consider three types of tokens that generated four different sentences in this code-mix dataset. Three different types of tokens are as follow:

1. Hindi Token: Hindi words written in Hindi letter.

[1] https://pypi.org/project/nlp-augment/

Table 1
Example of tweets and annotation from code-mixed datasets. The total dataset size 5740.

Source Tweet	Translation	Label	Type
”@HemantSorenJMM Nitin ji ne iskaa kutta bana diya.”	Who make him dog	HOF	Phonetic Hinglish
”Democracy doesn’t have “Kings” ...Dynasties have Kings ..@sambitswaraj Well said sir„ you never disappointed us”	-	NONE	English
चूतयि बनें या पैदा ही चूतयि हुए हैं्कूल	Be a pussy or are you born a pussy	HOF	Hindi
Where is the mask मुर्ख chuthiye	Where is the mask fool pussy	HOF	Hindi, Phonetic, English

2. English token: English words written in English letter.
3. Phonetic Hinglish token: Hindi word written phonetically in English letter.

Four different types of posts/sentence formation are as follow (e.g., posts Table 1):

1. Hindi posts consist of Hindi tokens,
2. English posts consist of English tokens,
3. Phonetic Hindi posts consist of the phonetic Hinglish tokens, and
4. Posts mixture of three types of tokens.

The second challenge of Subtask2 is related to its unique annotation criteria. For example, each tweet in the dataset could have a conversational thread that may contain hate and offensive content, which is not apparent just from a single comment or the reply to a comment but can be identified if given the context of the parent content. Table 2 shows relational annotation; for example, the comment’s reply ’You totally nailed it, can’t stop laughing’ seems not offensive; however, since it is supporting the original post, which was offensive, therefore it becomes offensive as well.

Table 2
Example of relational posts and annotation from datasets.

Posts type	Posts	Translation	Label
Comment	Doctors aur Scientists se manga hai. Chutiyo se nahi. Baith niche.	They have asked Doctors and Scientists. Not fuckers. Sit down.	HOF
Comment’s Reply	You totally nailed it, can’t stop laughing	-	HOF

2.1.1. Data Preprocessing

We have eliminated special characters, numeric values (e.g., @,0-9), newlines, mention tags, and URLs for data preprocessing. We have not removed hashtags since we have found them important. Table 3 shows example tweets before and after preprocessing.

Table 3
Example of posts, before and after preprocessing.

Source posts before preprocessing	**After preprocessing**	**Translation of Source posts**
”@HemantSorenJMM Nitin ji ne iskaa kutta bana diya.”	ji ne iskaa kutta bana diya	who make him dog
”@piyushpallow@HemantSorenJMM #resignmodi kon chutiya ka interview liye ho	#resignmodi kon chutiya ka interview liye ho	Who has interviewed fucker?

Table 4 shows different preprocessing outcomes for classifier accuracy for hate-offensive detection for Subtask2 using LR classifier with TF-IDF word-level feature.

The use of emoji removal in the preprocessing stage does not affect the overall result. However, Newline + Tab Token, mention tag, and URL + Special Characters removal worked well and improved almost .5% in performance accuracy. Since hashtag (#) removal decrease .7% performance, we have not removed hashtag from our dataset.

Table 4
Accuracy scores changes in preprocessing. Result obtain using CNN with fastText embedding.

Preprocessing Type	**Accuracy scores**
All removed except Hash (#) tag	61.3
URL, Special Ch., Newline, Tab Token	61.1
USERNAME (user) mention tag	61.2
RT	61.1
Emoji	60.8
Stop-word	60.6
Steming	60.4
Hash (#) tag	58.6
No Preprocessing	60.8

2.2. Code-mixed Data Augmentation

As discussed in Section 2.1, the dataset has three different types of tokens that have formed four different kinds of posts. Our proposed augmentation methods are followed by three different kinds of approaches that have been employed to cover all types of tokens and posts. For example, if posts consist of all 3 types of tokens, it is impossible to translate the sentence and reform it as a meaningful sentence. In that case, each token was targeted individually and identified by its type for further processing. For Hindi/English token, it is replaced by its synonyms. However, if the token was phonetics, it is converted to Hindi word using python library[2]; finally, synonym replacement is performed. After each token conversion, we have restored the sentence with

[2] https://pypi.org/project/pyhinavrophonetic/

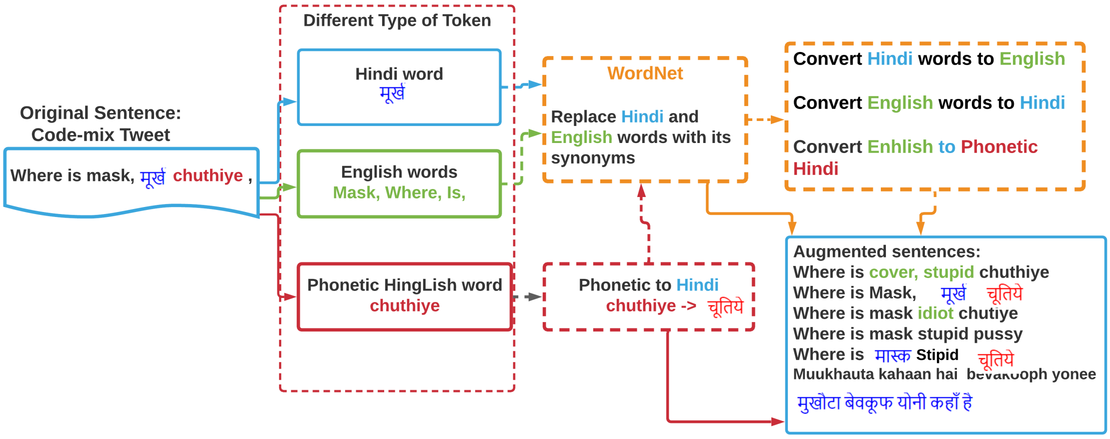
Different Type of Token
Original Sentence:
Code-mix Tweet
Where is mask, मूर्ख chuthiye ,
Hindi word
मूर्ख
English words
Mask, Where, Is,
Phonetic HingLish word
chuthiye
WordNet
Replace Hindi and English words with its synonyms
Phonetic to Hindi
chuthiye -> चूतिये
Convert Hindi words to English
Convert English words to Hindi
Convert Enhlish to Phonetic Hindi
Augmented sentences:
Where is cover, stupid chuthiye
Where is Mask, मूर्ख चूतिये
Where is mask idiot chutiye
Where is mask stupid pussy
Where is मास्क Stipid चूतिये
Muukhauta kahaan hai bevakooph yonee
मुखौटा बेवकूफ योनी कहाँ है

Figure 1: Dataset augmentation using synonym replacement, and phonetics conversion.

different types of the augmented token. Figure 1 shows the token-based augmentation and reformation of new posts.

If posts/sentences contain more than 90% token where all of them are either Hindi or English, a translation, back-translation, and Hindi phonetics conversion have been applied to the whole posts altogether. Figure 2 exhibits an example of translation, back-translation and phonetics conversion of posts.

1. Posts that contain more than 90% of English token: three types of augmentation strategies were performed (i) English to Hindi translation, (ii) Hindi to a phonetics, written in English but pronounced as Hindi, and (iii) Hindi to Bagla and back-translated to English again.
2. Similarly, posts that contain more than 90% Hindi token: three types of augmentation performed (i) Hindi to English translation, (ii) Hindi to phonetic, and (iii) Back-translation, Hindi to Bangla to Hindi .

Finally, both word-level and post-level augmentation were saved to CSV files containing 132k augmented sentences, which were 23.8 times larger than non-augmented ones.

2.3. Classifier architecture

Initially, we employed a random split of the original dataset into 80% for training and 20% for testing and validation, ensuring the same proportion of dataset for all kinds of model learning. Three classifiers were implemented for training and testing: Logistics regression (LR) with word-level TF-IDF, Convolution Neural Network (CNN) with word-level TF-IDF, and fine-tuned BERT pre-trained model. During the experiment, the test data has not been changed for both augmented and non-augmented datasets.

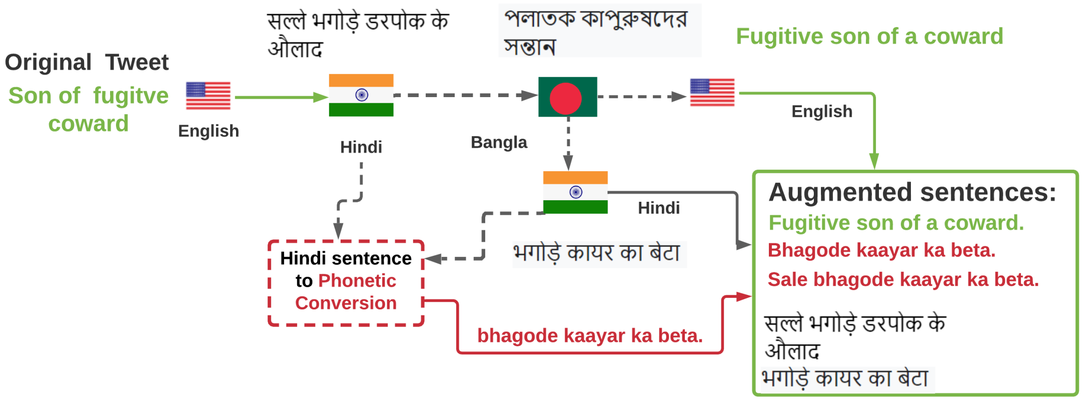

Figure 2: Dataset augmention using translation, back-translation and phonetic conversion.

Table 5
Example of generated sentences by word-level augmentation (synonym replacement and phonetics word conversion) and posts-level augmentation (translation, BT, and phonetics conversion). BT refers to back-translation, Aug refers to augmentation, Eng refers to English, and Hi refers to Hindi.

Original Sentence	Word level Aug.	Sentence level Aug.	Augmentation steps
Son of fugitive coward	-	Fugitive son of a coward	Eng. > Hi. > Eng. BT.
	-	Bhagode Kaayar ka beta	Eng. > Hi. > Phonetics
	-	Sale bhagode Kaayar ka beta	Eng. > Hi. > Phonetics
	-	भगोड़े कायर का बेटा	Eng. >Hi. translation
Where is the mask मुर्‌ख chuthiye	Where is the mask <fool> <pussy>	-	Hi. > Eng, Phonetic > Hi. > Eng.
	Where is the <cover> मुर्‌ख chuthiye	-	Eng. synonym (mask > cover)
	Where is the mask <idiot> chuthiye	-	Hi. > Eng.

2.4. Experiment Setup CNN

We adopted [25] a CNN, architecture, where the input layer is represented by a concatenation of the words forming the post (up to 70 words), except that each word is now a TF-IDF vector representation. Though several experiments showed using word embedding (e.g., fastText) performed better when used with CNN in the embedding layer[20]; however, no available pre-trained word embedding was found for code-mixed Hindi-English. A convolution 1D operation with a kernel size 3 was used together with a max-over-time pooling operation over the feature map with a layer dense 50. Dropout on the penultimate layer with a constraint on 12-norms of the weight vector was used for regularization.

The details of the implementation are reported on our GitHub page of this project with datasets and codes[3].

2.5. Transformer model

BERT – Bidirectional Encoder Representations from Transformers: this seminal transformer-based language model employs an attention mechanism that enables the mode to learn contextual relations between (sub-)words in a text sequence [26]. BERT uses two training strategies:

1. MLM: where 15 % of the tokens in a sequence are replaced (masked) for which the model learns to predict the original tokens, and
2. NSP where the model receives pairs of sentences as input and learns to predict whether or not the second sentence is a successor of the first one in their original document context.

2.6. Experiemnt setup with BERT model

We fine-tuned different transformer models with the HASOC-2021 training data using the corresponding test data for validation. The following models were tested: BERT-base (uncased) and BERT-multilingual. Each model was fine-tuned for 6 epochs with a learning rate of 5e-6, maximum sequence length of 128, and batch size 4. After each epoch, the model was evaluated on the validation set.

Table 6
Accuracy and F1 scores for Hate-Offensive detection code-mixed Subtask2 with augmented and non-augmented dataset.

Classifier	Not-expanded Dataset		Expanded dataset	
Feature Name	**Acc**	**F1**	**Acc**	**F1**
LR+Word Level TF-IDF	57.8	57.2	60.1	59.8
CNN + Word Level TF-IDF	66.1	65.3	**69.3**	**68.5**
BERT-base-uncased	65.8	65	69	68.1
BERT-base-multilingual-uncased	64.3	64.2	67.5	67.3

3. Results

Table 6 shows the results of binary offensive language detection (assuming all tweets as either hate or non-hate) using Logistic Regression (LR) as a baseline, CNN model, as well as BERT-base

[3] https://github.com/saroarjahan/Hasoc_2021_subtask2 (accessed September 20, 2021)

Table 7
Official results of our HASOC-21 test set submissions for code-mixed (Hindi-English) Subtask2.

Team name	Task	Macro F1	Rank
TeamBd	Subtask2	66.56	8

and BERT-multilingual. The results exhibit model accuracy and F1 scores for both augmented and non-augmented dataset for comparison. In both cases, we see that CNN with word-level TF-IDF largely outperform baseline LR. Comparing BERT and CNN model reveals that CNN model slightly (.5%) outperform both BERT-base-uncased and BERT-multilingual. It is unusual that CNN showed much better performance compared to BERT models. The possible explanation is that we need a pre-trained model that would reflect our dataset; however, no pre-trained models were found that can be useful for code-mixed. Though BERT multilingual was trained with 104 languages; however, the BERT-multilingual model has not outperformed CNN. This agreed with our intuition since the multilingual pre-trained model was trained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective; however, it contains only a small percentage of English and Hindi tokens. Therefore, it might have fallen short as the HASOC-2021 code-mixed dataset is related to Hindi, English, and amalgamation of phonetics tokens.

Our data augmentation results showed success since it has shown a better performance in all models. After using data augmentation for model training, it resulted in almost 3% increase compared to the non-expanded dataset. This improvement has been shown for baseline, CNN, and BERT models. Since we have kept the experiment test dataset identical for all models, this improvement justifies the proposed augmentation method that has helped the model to train better.

We have submitted our best performing model (CNN) results to the HASOC-21 competition, and an official result [27], we have received was 66.56% F1 score (Table 7).

4. Error Analysis

Our final submission F1 macro score was 66.56%, which has outperformed most submissions; however, the F1 score seems low, which indicates the model exhibits a large portion of false detection. We performed in this section an analysis of the model's performance and train dataset evaluation to understand this phenomenon better. For this purpose, we randomly prepared 200 subsets of test data then manually inspected each annotation.

From Figure 3, we see that 232 of the error is related to false-negative (FN), and 421 hate samples were correctly identified. In contrast, 477 non-hate samples were correctly detected, and 218 resulted as false positive (FP). This indicates that our model was not performing as good as detecting non-hate samples. Since our train dataset in its majority (55%) contains non-hate samples, it seems our model is better trained or biased towards non-hate classes. Another possible explanation of the overall model's low performance is due to the fact that errors might be coming from the training samples. For example, our manual inspection of 200 samples showed that 4% of the test sample was annotated wrongly (Table 4). Since this test data was a

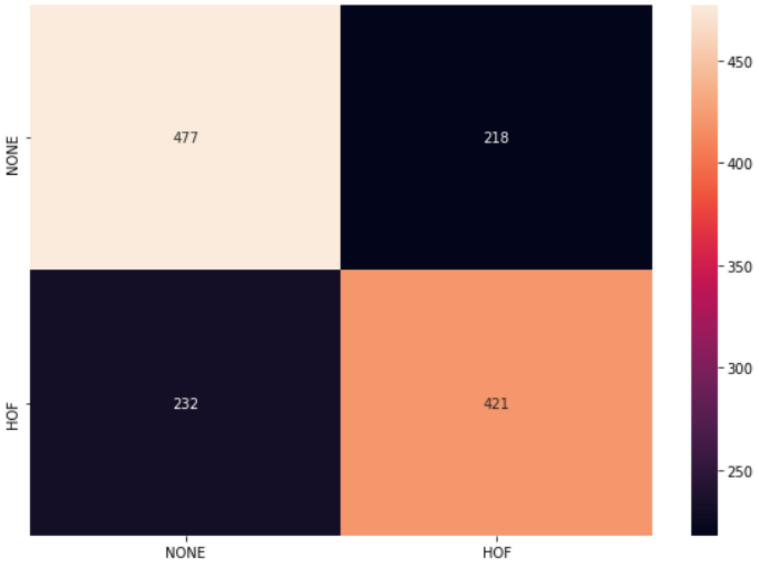

Figure 3: Confusion matrix of 1348 test samples.

split of original data, the wrong annotation of test data raises the question of overall training data quality, which might have affected model training. Furthermore, the training data were mostly relational data of tweets, retweets, comments, and replies. Therefore a single post was an amalgamation of hate and not-hate samples, which made the classifier difficult to train.

5. Conclusion

In the HASOC-2021 competition, we have worked on Code-Mixed (Hindi-English) dataset Subtask2 for hate-offensive post identification. We performed different experiments of twitter preprocessing, classification with LR, CNN, and BERT-finetuned models. Our tweet preprocessing showed removing the mentioned tag and removing special characters and URLs were useful and yield an increased of almost 1% in classification accuracy. However, eliminating hashtags and stemming reduced the overall performance. This provides a basis for optimal preprocessing pipeline. For classification purpose, we proposed a data augmentation strategy for code-mix dataset. Since the code-mixed dataset is a combination of Hindi, English, and HingLish tokens, we applied data augmentation that employs WordNet synonym replacement, conversation phonetics to Hindi, translation, and back-translation of sentences. After training with the augmented dataset, all models showed an approximate 3% improvement in classification accuracy and F1 accuracy. Among BERT and CNN, CNN performed little better than BERT models since there was no available pre-trained model for the code-mix dataset. Our best test CNN models showed 69.6 F1 scores, and the official submission result showed 66.65% f1 scores. In the future, it would be interesting to experiment with contextual augmentation using Code-Mixed pre-trained model.

Table 8
Sample test dataset, example of wrong annotation.

Tweets	**Original Label**	**label should be**
Picking up any matter, come with casting the cast, now stop, it is not India of 1947, 2021 has come, now leave this thing	HOF	NONE
watch video randeep hooda make dirti joke nation leader mayavati dalit woman voic oppress toler castiest peopl arrest #arresterandeephood chup teri ka bsda	HOF	
watch video randeep hooda make dirti joke nation leader mayavati dalit woman voic oppress toler castiest peopl arrest #arresterandeephood chup kar saal ka baccha joke funni	NONE	HOF
watch video randeep hooda make dirti joke nation leader mayavati dalit woman voic oppress toler castiest peopl arrest #arresterandeephood Supper	HOF	NONE
watch video randeep hooda make dirti joke nation leader mayavati dalit woman voic oppress toler castiest peopl arrest #arresterandeephood The show should be cancelled	HOF	NONE
watch video randeep hooda make dirti joke nation leader mayavati dalit woman voic oppress toler castiest peopl arrest #arresterandeephood God is Ram Rahim for you, now we don’t expect you to use your brain	NONE	HOF

6. Acknowledgments

We would like to thank the HASOC-2021 sharing task organizers for giving us the opportunity to submit this work. Secondly, this project was partially funded by EU Project WaterLine (Downscaling Remotely Sensed Products to Improve Hydrological Modelling Performance), which is gratefully acknowledged.

References

[1] J. T. Nockleby, Hate speech, Encyclopedia of the American constitution 3 (2000) 1277–1279.

[2] J. Risch, R. Krestel, Delete or not delete? semi-automatic comment moderation for the newsroom, in: Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018), 2018, pp. 166–176.

[3] 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.

[4] M. Wiegand, M. Siegel, J. Ruppenhofer, Overview of the germeval 2018 shared task on the identification of offensive language (2018).

[5] Y. J. Foong, M. Oussalah, Cyberbullying system detection and analysis, in: 2017 European Intelligence and Security Informatics Conference (EISIC), IEEE, 2017, pp. 40–46.

[6] 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), 2018, pp. 1–11.

[7] K. Reynolds, A. Kontostathis, L. Edwards, Using machine learning to detect cyberbullying, in: 2011 10th International Conference on Machine learning and applications and workshops, volume 2, IEEE, 2011, pp. 241–244.

[8] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, R. Picard, Common sense reasoning for detection, prevention, and mitigation of cyberbullying, ACM Transactions on Interactive Intelligent Systems (TiiS) 2 (2012) 1–30.

[9] Y. Bounab, J. M. Adeegbe, M. Oussalah, Towards storytelling automatic textual summerized, in: Conference of Open Innovations Association, FRUCT, 25, FRUCT Oy, 2019, pp. 434–438.

[10] M. S. Jahan, O. Mourad, Team oulu at semeval-2020 task 12: Multilingual identification of offensive language, type and target of twitter post using translated datasets, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2020, pp. 1628–1637.

[11] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, M. Shrivastava, Iiit-h system submission for fire2014 shared task on transliterated search, in: Proceedings of the Forum for Information Retrieval Evaluation, 2014, pp. 48–53.

[12] S. Agrawal, A. Awekar, Deep learning for detecting cyberbullying across multiple social media platforms, in: European Conference on Information Retrieval, Springer, 2018, pp. 141–153.

[13] S.-J. Bu, S.-B. Cho, A hybrid deep learning system of cnn and lrcn to detect cyberbullying from sns comments, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2018, pp. 561–572.

[14] N. Lu, G. Wu, Z. Zhang, Y. Zheng, Y. Ren, K.-K. R. Choo, Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts, Concurrency and Computation: Practice and Experience (2020) e5627.

[15] H. Rosa, D. Matos, R. Ribeiro, L. Coheur, J. P. Carvalho, A "deeper" look at detecting cyberbullying in social networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[16] 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), arXiv preprint arXiv:1903.08983 (2019).

[17] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, Ç. Çöltekin, Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020), arXiv preprint arXiv:2006.07235 (2020).

[18] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14–17.

[19] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.

[20] M. S. Jahan, M. Oussalah, A systematic review of hate speech automatic detection using natural language processing, arXiv preprint arXiv:2106.00742 (2021).

[21] 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.

[22] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Advances in neural information processing systems 28 (2015) 649–657.

[23] S. Wang, J. Liu, X. Ouyang, Y. Sun, Galileo at semeval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models, arXiv preprint arXiv:2010.03542 (2020).

[24] D. R. Beddiar, M. S. Jahan, M. Oussalah, Data expansion using back translation and paraphrasing for hate speech detection, Online Social Networks and Media 24 (2021) 100153.

[25] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).

[26] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[27] 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.