Bias, Threat and Aggression Identification using Machine Learning Techniques on Multilingual Comments

Kirti Kumari, Shaury Srivastav, Rajiv Ranjan Suman

Abstract

In this paper, we presented our team "$IIITRanchi$" for the Trolling, Aggression and Cyberbullying (TRAC-3) 2022 shared tasks. Aggression and its different forms on social media and other platforms had tremendous growth on the Internet. In this work we have tried upon different aspects of aggression, aggression intensity, bias of different forms and their usage online and its identification using different Machine Learning techniques. We have classified each sample at seven different tasks namely aggression level, aggression intensity, discursive role, gender bias, religious bias, caste/class bias and ethnicity/racial bias as specified in the shared tasks. Both of our teams tried machine learning classifiers and achieved the good results. Overall, our team "$IIITRanchi$" ranked first position in this shared tasks competition.

Keywords— Aggression, Multilingual comments, Tokenization, TF-IDF, BoG, Logistic Regression.

1 Introduction

Social media is an open platform where users can interact, share, learn and behave openly with other online users. Due to the high demand and popularity of these media, aggression and its manifestations in different forms have taken unprecedented proportions. Users of these media are generally writing their posts in multilingual forms (Kumar et al., 2022; Kumari and Singh, 2020b,a). So, identification of such kinds of aggression, threats and biases are not an easy task due to various reasons like these comments are unstructured, multilingual, short forms and highly contextual in nature. Due to these challenges, the research communities are very much interested in such kinds of automated identification. We have tried to develop systems that could automatically identify and separate these posts from the normal posts on the aggression shared dataset (Kumar et al., 2022).

For the given tasks, we have attempted all seven different categories of the text and classified them into their classes using different machine learning classifiers. The main motive of the work is to develop an efficient Machine Learning system to detect the aggression, biased and threatening contents on the social media platform which can be removed and altered afterwards. This will prevent the negative impact on many users and hate that may spread in society. The proposed models have different machine learning algorithms and we have fine tuned the models with different hyper-parameters which we have found for testing and cross validation phases. We found better results for all the shared tasks and ranked first. Our team (IIITRanchi) ranked first on all the Task1, Task1 surprise tests, and also in Task2.

In the preceding section, we have discussed a detailed description of the some related works, dataset, the preprocessing steps involved, the initial challenges and the models which we used for our use case.

2 Related Work

The variety of aggression related works have been proposed by researchers in the last few years. Aggression related shared tasks were proposed by the organising team of Shared Tasks on Aggression Identification in every second year 2018 (Kumar et al., 2018), 2020 (Bhattacharya et al., 2020) and 2022 (Kumar et al., 2022). Some of the recent works are discussed as:

At first, we are discussing some of the important works on 2018 aggression dataset (Kumar et al., 2018). The work (Risch and Krestel, 2018) used ensemble learning and data augmentation techniques. They augmented English training dataset with the help of machine translation using three languages (French, German and Spanish) by preserving the meaning of comments with different wording. Their system was not stable for Hindi dataset across the platforms (Facebook and Twitter). Their system is not stable, especially for Hindi dataset for the same domain it was performed well, but for other domain, it fails to classify the tweets with good accuracy. Aroyehun and Gelbukh (Aroyehun and Gelbukh, 2018) used various deep learning models such as Long Short Term Memory (LSTM), CNN, and FastText as word representation and data augmentation techniques by machine-translating the original post into different languages and then translated back to the original language. Their system was not clearly classified covertly aggressive comments from overtly aggressive comments with significant accuracy. Julian and Krestel

Proceedings of the 29th International Conference on Computational Linguistic, pages 30–36
October 12–17, 2022.

(Risch and Krestel, 2018) and Aroyehun and Gelbukh (Aroyehun and Gelbukh, 2018) found that augmentation of training data gives a better result. Raiyani et al. (Raiyani et al., 2018) used dense system architecture and compared several models such as dense neural network, FastText and voting-based ensemble model. They found that simple three-layer dense neural network was performing better than the other two (FastText and voting-based ensemble classification) models. Their system has continued to suffer from false-positive cases and has also overlooked words that are not available in their vocabulary.

Some important works on 2020 aggression dataset (Bhattacharya et al., 2020; Kumar et al., 2020). Julian and Krestel (Risch and Krestel, 2020) uses transformer based multiple fine-tuned BERT models based on bagging technique and found very good results. THe work (Mishra et al., 2020) also used the transformer based BERT models and achieved good performance.

3 Dataset

In this section, we discuss brief descriptions about datasets (Kumar et al., 2022) and given shared tasks.

3.1 Tasks

The following tasks defined by the organizing teams as: (a) Aggression, Gender Bias, Racial Bias, Religious Intolerance and Bias and Casteist Bias on social media and (b) the "discursive role" of a given comment in the context of the previous comment(s). Further these task are subdivided into some subclasses as: Gender Bias: It has three subclasses problem: Gender (GEN), Gender Threat (GENT) and Non-Gender (NGEN). Ethnicity/Racial Bias: It has three subclasses Ethnic/Racial comments (ETH), Ethnic/Racial Threat(ETHT), Non Ethnic/Racial comments(NCOM). Communal bias: It has three subclasses Communal (COM), Communal Threat (COMT), Non-Communal (NCOM). Caste/class bias: It has three subclasses Casteist/Classist comments (CAS), Casteist/classist Threat (CAST), Non-Casteist/Classist comments (NCAS). Aggression Level: It has three subclasses 'Overtly Aggressive'(OAG), 'Covertly Aggressive'(CAG) and 'Non-aggressive'(NAG) text data. Aggression Intensity: This level gives a 4-way classification in between 'Physical Threat'(PTH), 'Sexual Threat'(STH), 'Non-threatening Aggression'(NtAG) and 'Curse/Abuse'(CuAG). Religious Bias: At the level E, the task is to develop a 3-way classifier for classifying the text as 'communal' (COM), 'Communal Threat'(COMT) and 'non-communal'(NCOM).

The dataset (Kumar et al., 2022) is multilingual with a total of over 140,000 samples (over 60,000 unique samples) for training and development and over 15,000 unique samples for testing in four Indian languages Meitei, Bangla (Indian variety), Hindi and English. The dataset consists of comments from a total of 158 videos i.e., it has a comment thread in total. All the data is collected from YouTube. This dataset is manually annotated by multiple annotators. The phenomena of aggression/bias is a function of certain parameters. These parameters have been discussed properly in the article (Agha, 2006). The three contextual factors included in the tasks are aggression, gender bias and communal bias. The training data contains a mixed corpus of multilingual code-mixed comments in four Indian languages:

- Meitei
- Bangla
- Hindi
- English

Language-wise distribution is approximately 26.3% Meitei, 27.8% Bangla, 45.9% Hinglish(Hindi and English). The detailed description of the dataset can be found in the article (Kumar et al., 2022).

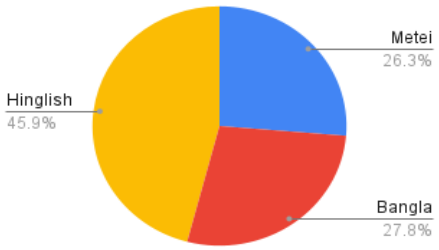

Figure 1: Pie chart for language-wise distribution of the given training dataset.

3.2 Initial challenges

The major challenges in the dataset were the unwanted words and the code-mixed nature of the dataset. We started out by cleaning the dataset with the help of certain techniques. We went for transliteration first but due to the change in meaning of many words we didn't go forward with that method. We then used some of the text data preprocessing techniques and discussed in the following section.

4 Preprocessing

In this section we are going to describe the preprocessing steps we did for cleaning the dataset.

4.1 Removing noise:

The dataset has data from youtube hence mention of users, other hyperlinks are noise for us. Simple regex(regular expression) based rules were used to remove discrepancy : The general method was

- Anything which was followed by @ such as @user was removed as it didn't add much of a context to our tasks.
- Anything starting with https//: was removed.

- Unidentified characters such as emojis with no general meanings were also removed.

4.2 Removing Punctuation and special symbols:

Since, we were going for Bag of Words(BoG) model and Term-Frequency Inverse Document Frequency (TF-IDF) vectorizer concept usage of punctuation didn't have any impact on the dataset, hence we removed the punctuations and other special characters as well.

4.3 Lowering of dataset:

We have lowered the case of the sentences of the dataset to form uniformity in the dataset and we don't have to care about the case of the dataset then for code-mixed data. Words such as 'ACHA' were converted to 'acha' for uniformity and space complexity constraints.

4.4 One Hot Encoding:

Categorical values corresponding to each class were label encoded as 0,1,2,3 to respective tasks having different classes.

4.5 Vectorization:

For vectorization, we used TF-IDF vectorizer as we used classical Machine Learning (ML) models for classification. Since the tasks were binary in nature (mostly as we had to predict whether a certain comment was aggressive or not and then its extent) so just the presence of certain words made it offensive and non-offensive and aggressive and non-aggressive . Therefore a simpler approach such as TF-IDF was used instead of the word2vec technique . Example : @user Agiye cholo aamra aachhi #goodfeels After cleaning we get - Agiye cholo aamra aachhi, goodfeels The final sparse matrix after vectorization was then feeded to the models for predictions.

5 Models

We have considered Naïve Bayes, Decision Tree based Random Forest, Logistic Regression and Support Vector Machines (SVM) for text classifications. We have used training data on each model by performing GridSearchCV for all the combinations of feature parameters. We have analyzed performance on the basis of a weighted average micro f1-score of the cross validation.

5.1 Support Vector Machine

The first model with which we started out was SVM as its state of the art for classification tasks. The parameters we used were C = 1 and the kernel was linear but it didn't perform well mostly due to the non linearly separable nature of the dataset .Then we moved on to the polynomial kernel . With degrees as high as 8-9 also the model didn't perform well as it failed to generalize. The next change we made was : kernel = 'rbf' C = 1 , this performed well on train data. This model generalized well and gave better classification results as compared to above methods.

5.2 Multinomial Bayes (MNB)

The next model we used was Multinomial Bayes which is commonly used for text classification. The initial model didn't perform well with value of = 0 , as the minimum count of many words was zero in many of the cases so the joint probability was returning zero and hence the classes were being misclassified. We used grid search cross validation for finding the best and found best results on train data with = 1e-03 But due to a sparse dataset from TfidfVectorizer it had some limitations and even with hyper parameter tuning the performance didn't improve much.

5.3 Decision Trees and Random Forest

We started the tree methods with a decision trees classifier but even with higher values of max depth and other parameters it didn't perform well. Then we moved to ensemble techniques such as random forest. Random Forest overfitted on the training dataset, and hence it was not able to capture a general trend in the dataset and failed to provide good results on validation set. We choose the criterion for split to be entropy and the max depth of each tree to be 4 in Random Forest.

5.4 Logistic Regression (LR)

The last model, we used was Logistic Regression. A simple classifier model with different C values. Due to its (almost) binary nature and multi-class solver newton-cg logistic regression performed really well on the training dataset. We got a very generalized model for all tasks. We modified the value of penalty parameter C to higher values also and got the best result at C = 5. With the generalized models ready, we used these models to assess the results on the unseen testing data which contained three types - dataset with surprise text, COVID comments dataset and data with no surprise text.

5.5 Ensemble techniques

We then moved on with ensemble based boosting methods. We used XGB Classifier and Adaboost techniques in view of better variance and better results.

5.5.1 XGB classifier

Due to the sparse nature of the data, single decision trees couldn't perform well. We chose many hyperparameters for this model but it failed to provide better results . Hence we didn't move forward with this model in the training phase. We choose the criterion for split to be entropy and the max depth of each tree to be 4 in XGB classifier and the final criteria to be Softmax.

5.5.2 AdaBoost

The last model we used was Ada Boost. Being a boosting method we expected the variance to be better in this case. For base learners we chose a decision tree with max-depth =3 and the criterion of split to be entropy. Since the model was unable to make proper decisions on the basis of sparse data the performance was not par with the models we used before.

Task	SVM	LR	RF	MNB	ADB	XGB
Aggression	0.76	0.78	0.54	0.74	0.52	0.54
Aggression Intensity	0.79	0.76	0.66	0.72	0.63	0.66
Discursive Role	0.91	0.86	0.71	0.85	0.68	0.71
Gender Bias	0.92	0.91	0.82	0.89	0.81	0.82
Communal Bias	0.96	0.95	0.85	0.94	0.82	0.85
Ethnicity-Racial Bias	0.99	0.99	0.867	0.99	0.82	0.867
Caste bias	0.99	0.99	0.87	0.98	0.74	0.87
Overall	**0.90**	0.89	0.75	0.87	0.717	0.75

Table 1: Micro averages of training dataset tasks with different models

Task	SVM	LR	MNB
Aggression	0.66	0.70	0.70
Aggression Intensity	0.66	0.67	0.69
Discursive Role	0.71	0.87	0.82
Gender Bias	0.87	0.89	0.90
Communal Bias	0.93	0.95	0.95
Ethnicity-Racial Bias	0.98	0.99	0.99
Caste bias	0.96	0.98	0.98
Overall	**0.76**	**0.86**	0.84

Table 2: Micro averages of testing dataset on task-1 with different models.

With the generalized models ready, we used these models to assess the results on the unseen testing data which contained three types - dataset with surprise text , covid comments dataset and data with no surprise text.

6 Results and Discussion

In this section, we present our findings and observations of this work.

6.1 Training data

The results, we present here are based on a weighted average micro F1-Score and some abbreviation used as: *LR - logistic regression *MNB - Multinomial bayes *SVM - support vector machines *RF- random forest *ADB - Adaboost * XGB - XG boost

The SVM model tends to fit perfectly to training data with a weighted average micro f1-score over 0.90 for many of the tasks due to its soft margin nature and flexibility in C value . The kernel used is Gaussian hence the model tends to mimic the training data really well.

6.2 Testing data

The testing data consisted of three tasks which had different datasets for evaluation in the competition.

- Task-1 Data without surprise language
- Task -2 Covid comments data
- Task-3 Data with surprise language

6.2.1 Task-1 Data without surprise data

In the above task we have seen, Logistic Regression performs better in all tasks even though SVM performed better on train dataset.

Task	SVM	LR	MNB
2018 Aggression	0.38	0.47	0.48
2020 Aggression	0.45	0.65	0.66
2022 Aggression	0.40	0.70	0.70
covid Aggression	0.30	0.63	0.60
Overall	0.30	**0.63**	0.60

Table 3: Micro averages of testing dataset on Task-2 with different models.

Task	SVM	LR	MNB
Aggression	0.60	0.62	0.63
Aggression Intensity	0.46	0.46	0.47
Discursive Role	0.69	0.88	0.84
Gender Bias	0.91	0.92	0.92
Communal Bias	0.95	**0.96**	0.96
Ethnicity-Racial Bias	0.99	0.99	0.99
Caste bias	0.98	0.98	1.00
Overall	**0.81**	**0.87**	**0.87**

Table 4: Micro averages of testing dataset on task-3 with different models.

6.3 Task-2 Data with Covid-19 comments

This data contains comments in codemixed languages where the context is based on Covid-19. So many of the texts are offensive and many have negative aspects to it as well. Logistic regression again outperforms all other models.

6.3.1 Task-3 Data with surprise data

In this case Logistic Regression and MNB both perform equally but MNB performs well on each of the subtasks individually.

6.4 Reasons for not moving towards deep learning techniques

When we talk about Natural Language Processing task, we directly take into account the popular models such as various forms of BERT models. But since in our case the simpler model (Logistic Regression) was performing well, hence we didn't move on to the deep learning model. While analysing our dataset, we found that the BERT based existing models had tokenizers which use sentence piece methods and hence while our dataset was code-mixed it would break the useful words into irrelevant tokens can be seen in the following example. For example: BERT's tokenizer doesn't have the word 'ANNA' (brother). So, it breaks down the word into 'AN' ,'NA' , which isn't even close to brother. One of the other major reasons was the size of the dataset, since it was small, we couldn't make our own embeddings for better performance as many of the things were required such as sentence piece tokenization and that requires a lot of data. The other reason was, these were simple classification tasks: i.e: whether a sentence is aggressive

or not. So the presence of certain words were the only parameters we had to take care of. Hence, in our case Logistic Regression performed really well. We used Sklearn package (Pedregosa et al., 2011) to develop the models.

7 Error Analysis

In this section, we presented error analysis of our models. The size of training data was sufficient for ML models as we came across a large number of vocabulary. Since the metric used was micro F1-Score and most of the tasks had only 2-3 classes we got good results as the micro F1-Score came out to be above 0.70 on an average therefore class wise classification scores were also better. Model performs well on most of the tasks. By good performance, we mean good class wise F1-Score on all the respective classes. Few of the tasks where there was a surplus of one class had a lesser macro average due to absence of context aware classification but mostly the model has outperformed all other techniques. The confusion matrix and classification reports of Logistic Regression model's performance on training data are given below: All the respective classes are encoded to one categorical numeral below is the dictionary for that.

'Aggression': 'CAG': 0, 'OAG': 1, 'NAG': 2,

	precision	recall	f1-score	support
0	0.50	0.73	0.59	14111
1	0.90	0.79	0.84	50217
2	0.80	0.78	0.79	27757
accuracy			0.78	92085
macro avg	0.73	0.77	0.74	92085
weighted avg	0.81	0.78	0.79	92085

Figure 2: Classification report for Aggression task on training data.

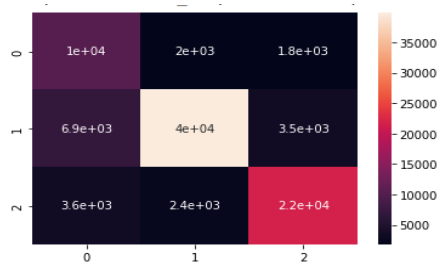

Figure 3: Confusion matrix for Aggression task on training data.

'Aggression Intensity': 'NtAG': 0, 'NA': 1, 'CuAG': 2, 'STH': 3, 'PTH': 4,

'Discursive Role': 'NA': 0,'CNS': 1,'ATK': 2,'AIN': 3,'DFN': 4,'GSL': 5,

'Gender Bias': 'NGEN': 0, 'GEN': 1, 'GENT': 2,

'Communal Bias': 'NCOM': 0, 'COM': 1, 'COMT': 2,

	precision	recall	f1-score	support
0	0.81	0.72	0.76	41573
1	0.78	0.80	0.79	26516
2	0.72	0.81	0.76	21981
3	0.42	0.80	0.55	698
4	0.52	0.86	0.65	1317
accuracy			0.76	92085
macro avg	0.65	0.80	0.70	92085
weighted avg	0.77	0.76	0.77	92085

Figure 4: Classification report for Aggression Intensity task on training data.

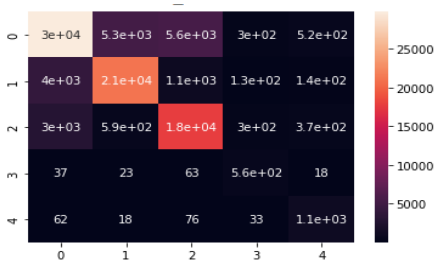

Figure 5: Confusion matrix for Aggression Intensity task on training data.

	precision	recall	f1-score	support
0	0.96	0.87	0.91	70561
1	0.08	0.75	0.15	53
2	0.72	0.84	0.77	20665
3	0.24	0.81	0.37	678
4	0.12	0.78	0.20	128
5	0.00	0.00	0.00	0
accuracy			0.86	92085
macro avg	0.35	0.68	0.40	92085
weighted avg	0.90	0.86	0.88	92085

Figure 6: Classification Report for Discursive Role task on training data.

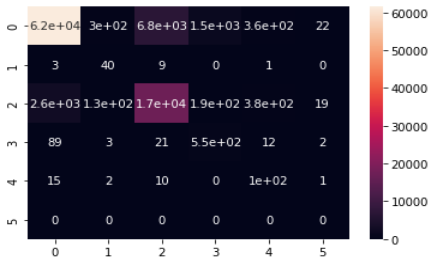

Figure 7: Confusion Matrix for Discursive Role task on training data.

'Caste/Class Bias': 'NCAS': 0, 'CAS': 1, 'CAST': 2,

'Ethnicity/Racial Bias': 'NETH': 0, 'ETH': 1, 'ETHT': 2

8 Conclusion

Our team secured the first position in the competition of the given shared tasks on bias, threat and aggression detection for given datasets. We found that the Logistic

	precision	recall	f1-score	support
0	0.98	0.92	0.95	81216
1	0.58	0.82	0.68	10597
2	0.22	0.85	0.35	272
accuracy			0.91	92085
macro avg	0.60	0.86	0.66	92085
weighted avg	0.93	0.91	0.92	92085

Figure 8: Classification Report for Gender Bias task on training data.

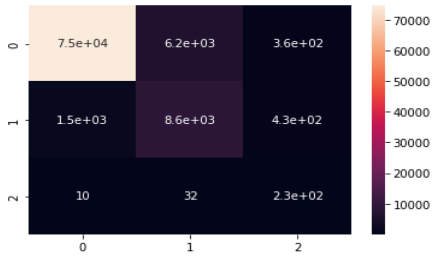

Figure 9: Confusion Matrix for Gender Bias task on training data.

	precision	recall	f1-score	support
0	0.99	0.96	0.98	85176
1	0.65	0.85	0.74	6843
2	0.13	0.80	0.23	66
accuracy			0.95	92085
macro avg	0.59	0.87	0.65	92085
weighted avg	0.96	0.95	0.96	92085

Figure 10: Classification Report for Communal Bias task on training data.

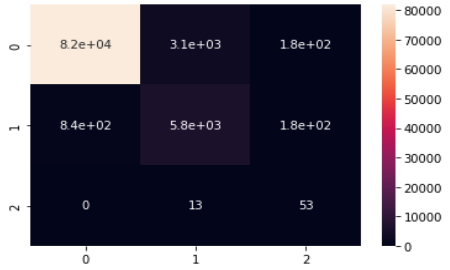

Figure 11: Confusion Matrix for Communal Bias task on training data.

Regression classifier outperforms all the models on test data due to more generalization and better prediction nature in sparse data. The possible reason was that the dataset in sparse form was linearly separable, but the SVM model being soft margin was not a generalized model for our case. SVM model was overfitting on train data but Logistic Regression generalized the metrics and hence performed really well in our case. At the end, we would like to conclude with the possibility that there

	precision	recall	f1-score	support
0	1.00	0.99	0.99	91577
1	0.35	0.86	0.49	508
2	0.00	0.00	0.00	0
accuracy			0.99	92085
macro avg	0.45	0.62	0.50	92085
weighted avg	1.00	0.99	0.99	92085

Figure 12: Classification Report for Caste/Class Bias task on training data.

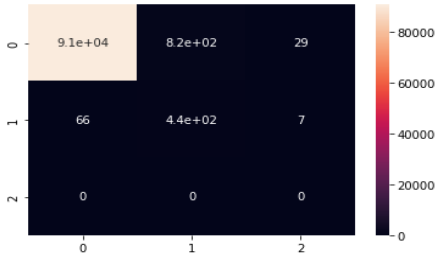

Figure 13: Confusion Matrix for Caste/Class Bias task on training data.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	90554
1	0.50	0.86	0.63	1531
2	0.00	0.00	0.00	0
accuracy			0.98	92085
macro avg	0.50	0.61	0.54	92085
weighted avg	0.99	0.98	0.99	92085

Figure 14: Classification Report for Ethnicity/Racial Bias task on training data.

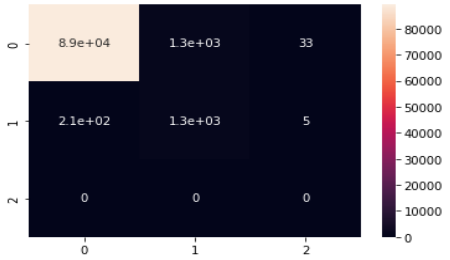

Figure 15: Confusion Matrix for Ethnicity/Racial Bias task on training data.

exists many of the techniques other than what we have presented in this paper. In order to improve the model's performance, we can go for ensemble techniques of the models which have performed well in order to increase the variance and make the model more generalized.

9 Acknowledgement

We would like to thank Mrinmoy Mahato, Amitesh Patel , Aman Kapoor and Ankit Kumar of Department

of Computer Science and Engineering, Indian Institute of Information Technology RANCHI - 834004 for their help in preprocessing steps.

References

Asif Agha. 2006. *Language and social relations*, volume 24. Cambridge University Press.

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Kr. Ojha. 2020. Developing a multilingual annotated corpus of misogyny and aggression. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, Marseille, France. European Language Resources Association (ELRA).

Ritesh Kumar, Enakshi Nandi, Laishram Niranjana Devi, Shyam Ratan, Siddharth Singh, Akash Bhagat, and Yogesh Dawer. 2022. The comma dataset v0. 2: Annotating aggression and bias in multilingual social media discourse. In *Proceedings of the Third Workshop on Trolling, Aggression and Cyberbullying*, pages 4149–4161, Marseille, France. European Language Resources Association (ELRA).

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the second workshop on trolling, aggression and cyberbullying*, pages 1–5.

Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kirti Kumari and Jyoti Prakash Singh. 2020a. Ai_ml_nit_patna@ hasoc 2020: Bert models for hate speech identification in indo-european languages. In *FIRE (Working Notes)*, pages 319–324.

Kirti Kumari and Jyoti Prakash Singh. 2020b. Ai_ml_nit_patna@ trac-2: deep learning approach for multi-lingual aggression identification. In *Proceedings of the second workshop on trolling, aggression and cyberbullying*, pages 113–119.

Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra. 2020. Multilingual joint fine-tuning of transformer models for identifying trolling, aggression and cyberbullying at TRAC 2020. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 120–125, Marseille, France. European Language Resources Association (ELRA).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Kashyap Raiyani, Teresa Gonçalves, Paulo Quaresma, and Vitor Beires Nogueira. 2018. Fully connected neural network with advance preprocessor to identify aggression over facebook and twitter. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 28–41.

Julian Risch and Ralf Krestel. 2018. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 150–158.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).