Team_BUDDI at ComMA@ICON: Exploring Individual and Joint Modelling Approaches for detecting Aggression, Communal Bias and Gender Bias

Anand Subramanian *
BUDDI AI
anands@buddi.ai

Mukesh Reghu *
BUDDI AI
mukeshr@buddi.ai

Sriram Rajkumar
BUDDI AI
sriramr@buddi.ai

Abstract

The ComMA@ICON 2021 Shared Task involved identifying the level of aggression and identifying gender bias and communal bias from texts in various languages from the domain of social media. In this paper, we present the description and analyses of systems we implemented towards these tasks. We built systems utilizing Transformer-based models, experimented by individually and jointly modelling these tasks, and investigated the performance of a feature engineering method in conjunction with a joint modelling approach. We demonstrate that the joint modelling approaches outperform the individual modelling approach in most cases.

1 Introduction

Social media has revolutionized how people communicate and engage in discourse and debate regarding various issues in society. In India, regional languages have influenced how content is generated on social media, with English not being the only language in which people interact with each other on forums. However, it is vital to ensure that discourse on social media is civil and respectful and does not serve as an outlet to malign or abuse users. Abusive or hostile content can manifest in various forms, including but not restricted to aggressive or hostile personal comments or posts, content that may be communal and malign religious sentiments or content that may be discriminatory based on gender. Therefore it is imperative to handle these types of content in a time-bound and sensitive manner. Modelling such text could help build automated or human-in-the-loop systems that can assist manual content moderators in reviewing and flagging such objectionable content.

Natural Language Processing can be extremely integral in this regard. However, modelling text from the social-media domain comes with challenges that need to be addressed. First of all, text generated on social media is significantly different from the text in books or newspapers. One such difference is the comparatively short length of texts in social media, such as tweets. Another difference would be the informal nature of discourse on social media forums, which includes the usage of slang, emojis and hashtags. Thirdly, social media text may be code-mixed, further complicating the process. An example of this is Hinglish, a combination of Hindi and English. These factors must be considered when modelling such text.

2 About the Task

The ComMA Project's Shared Task on Multilingual Gender Biased and Communal Language Identification (Kumar et al., 2021a) [1] [2] provided datasets spanning Hindi, Bangla (Indian variety), Meitei and English (Kumar et al., 2021b). The shared task comprised 3 sub-tasks which involved detecting the level of aggression, the identification of gender bias, and the identification of communal bias in a given text.

3 Challenges

Since the task involved data from informal domains of discourse like social media, some factors were to be considered while building systems for these tasks. Some of those considerations were:

1) **The dataset comprises code-mixed text for each language.** For instance, the text as part of the Hindi corpus may contain Hindi words written in English script (Hinglish), purely Hindi and purely English words as part of it. Thus, it is essential to

*Equal Contribution

[1] https://sites.google.com/view/comma-at-icon2021/overview

[2] https://competitions.codalab.org/competitions/35482

Proceedings of the 18th International Conference on Natural Language Processing: Shared Task on Multilingual Gender Biased and Communal Language Identification, pages 13–20
National Institute of Technology - Silchar, December 16-19, 2021.©2021 NLP Association of India (NLPAI)

ensure that models trained toward this task adapt to the code-mixed nature of the text.

2) **The level of aggression, presence of gender bias and communal bias are annotated for each text in the dataset.** The sub-tasks revolve around identifying these labels given a text. Multiple approaches are possible for solving these problems. The sub-tasks can be modeled independently or modeled jointly.

4 System overview

We built systems towards solving the three subtasks, for the Hindi corpus and the Multilingual Corpus, considering the factors mentioned above. To this end, for tackling the Hindi corpus, we utilize a BERT (Devlin et al., 2019) model, which was finetuned on Hinglish tweets with the Language modelling (LM) task (Bhange and Kasliwal, 2020) (Kasliwal and Bhange) (meghanabhange/Hinglish-Bert), hereafter referred to as **Hinglish-BERT**, as our starting point for all systems we submitted for the Hindi Task.

We utilize XLM-Roberta (**XLM-R**) (Conneau et al.) (Hugging Face - XLM-Roberta-Base) as our starting point for the system built as part of our submission towards the Multilingual Corpus as Hindi, Bengali, and English are part of the list of languages used for training the XLM-R model.

5 Methods

Each of these three tasks of aggression prediction **(AG)**, gender bias prediction **(GEN)** and communal bias prediction **(COM)** in the dataset are multi-class problems where the set of possible classes for each of the tasks are given by $\mathcal{Y}_{AG} = \{NAG, CAG, OAG\}$, $\mathcal{Y}_{GEN} = \{NGEN, GEN\}$ and $\mathcal{Y}_{COM} = \{NCOM, COM\}$. Refer to Table 1 for legend of the classes.

We are given a multi-task text classification dataset given by $\mathcal{D} = \left\{(x^{(i)}, y^{(i)})\right\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the set of all preprocessed texts in the dataset and $\mathcal{Y} = \mathcal{Y}_{\text{AG}} \times \mathcal{Y}_{\text{GEN}} \times \mathcal{Y}_{\text{COM}}$ is the set of all possible annotations or predictions for the text.

Also, for a given sample $(x, y) \in \mathcal{D}$ and task $t \in \{AG, GEN, COM\}$, we define y_t as the true class annotation corresponding to the task t.

We first preprocess each of the text in the raw dataset using the preprocessing steps from (Bhange and Kasliwal, 2020) to obtain the preprocessed texts $x \in \mathcal{X}$.

Short Form	Description
Aggression (AG)	
NAG	Non-aggressive
CAG	Covertly aggressive
OAG	Overtly aggressive
Gender Bias (GEN)	
NGEN	Non-gendered
GEN	Gendered
Communal Bias (COM)	
NCOM	Non-communal
COM	Communal

Table 1: Legend of short forms and descriptions for each of the classes for each of the tasks.

We then embed each of these preprocessed texts $x \in \mathcal{X}$ into a hidden representation $\mathbf{h}_x$ by feeding x to the Hinglish-BERT backbone and extracting its hidden representation corresponding to the **[CLS]** (classification) token (which is used as the representation for text **x**). For the **XLM-R** model, we use the representation of the $\langle \mathbf{s} \rangle$ token as the representation of the text.

We now define the function **Hinglish-BERT** which embeds a given preprocessed text x into a hidden representation $\mathbf{h}_x$ as described above.

$$\mathbf{h}_x = \textbf{Hinglish-BERT}(x) \tag{1}$$

For each of the tasks, $t \in \{AG, GEN, COM\}$, we then use *task-specific head layers* $\mathbf{H}_t$ to obtain the **prediction probabilities** $\hat{\mathbf{y}}_t$ for each of the classes in the task from the hidden representations of the texts as given by:

$$\hat{\mathbf{y}}_t = \mathbf{H}_t(\mathbf{h}_x) \in \{0, 1\}^{|\mathcal{Y}_t|} \tag{2}$$

where, the task specific head $\mathbf{H}_t$ is two fully connected layers stacked on one another with **ReLU** activation in between and **softmax** at the output as graphically represented in Fig 1.

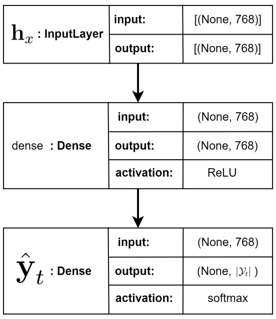

Figure 1: Architecture of the task-specific head $\mathbf{H}_t$

We train all our models end-to-end. We use the **Cross Entropy** loss for each of the individual tasks t, and define the task-specific loss $\mathcal{L}_t$ as:

$$\mathcal{L}_t = \textbf{CrossEntropyLoss}(\textbf{y'}_t, \hat{\textbf{y}}_t) \tag{3}$$

where, $\textbf{y'}_t$ is the one-hot probability vector corresponding to the true class annotation for the task t, y_t.

5.1 Three Individual Task-Specific Models

In this approach, we fine-tune three independent **Hinglish-BERT** models with *task-specific head* for each of the tasks and optimize them for their corresponding task-specific loss $\mathcal{L}_t$ (Equation 3).

The task-specific model $\textbf{model}_t$ and its *prediction probabilities* $\hat{\textbf{y}}_t$ corresponding to each of the tasks, $t \in \{AG, GEN, COM\}$ are given by:

$$\hat{\textbf{y}}_t = \textbf{model}_t(x) = \textbf{H}_t\big(\textbf{Hinglish-BERT}_\textbf{t}(x)\big) \tag{4}$$

5.2 Joint Modelling Approaches

We also build systems that jointly model the tasks using a single model architecture to investigate if performance improvements are possible due to joint modelling. A different method of jointly modelling such tasks was attempted in (Mishra et al., 2020).

The tasks are significantly intersectional, i.e., a text with *communal bias* present in it, may have *aggressive content* present, or a text may have *aggressive content* with an *overt gender bias*, etc. It is possible for the model to potentially learn better representations when these tasks are modeled jointly. These approaches also have significantly fewer parameters than training individual task-specific models due to a **shared Hinglish-BERT Backbone**.

5.2.1 Joint Model for Tasks (Three Heads)

In this approach, we jointly model the three tasks using a single model architecture that has a **common Hinglish-BERT backbone** with three task-specific heads (each corresponding to one of the tasks).

For each of the tasks, $t \in \{AG, GEN, COM\}$, the prediction probabilities for the classes in task are given by:

$$\hat{\textbf{y}}_t = \textbf{model}_t(x) = \textbf{H}_t\big(\textbf{Hinglish-BERT}_{\textbf{common}}(x)\big) \tag{5}$$

Method	Notation
Three Individual Models - Hindi Data	HIN-3-IND
Joint Model with Three Heads - Hindi Data	HIN-JNT-3H
Joint Model with Three Heads and Feature Engg - Hindi Data	HIN-JNT-3H+FE
Joint Model with Hierarchical Heads for Aggression Task - Hindi Data	HIN-JNT-4H
Joint Model with Three Heads - Multilingual Data	MULTI-JNT-3H

Table 2: Notations for denoting each of the systems

We then use the true class annotations and predicted class probabilities for each of the tasks to compute the task specific losses $\mathcal{L}_{AG}$, $\mathcal{L}_{GEN}$ and $\mathcal{L}_{COM}$. We then combine these losses by averaging them to get our overall loss $\mathcal{L}$, which we optimize for in our model.

$$\mathcal{L} = \frac{\mathcal{L}_{AG} + \mathcal{L}_{GEN} + \mathcal{L}_{COM}}{3} \tag{6}$$

In the multilingual case, we fine-tune an **XLM-R model** instead of a *Hinglish-BERT model* and jointly model the tasks using the same approach.

5.2.2 Joint Model for Tasks with Feature Engineering (Three Heads)

In this approach, we build upon our previous Joint Model for Tasks (Three Heads) approach. However, in the preprocessing step, we introduce a special token (i.e., an unused token from the BERT vocabulary) to act as a marker to surround words that could be informative to the model while learning the three tasks. These words could be obtained through a curated lookup.

For example: We used ”ye sabse bdi **[unused1] chutiya [unused1]** aurat h **[unused1] bc [unused1]**” to replace our preprocessed text ”ye sabse bdi **chutiya** aurat h **bc**” in which the words ”chutiya” and ”bc” are present in our lookup of curated words.

The usage of marker tokens has been widely explored for tasks like Relation Extraction (RE) in NLP (Wu and He, 2019) (Baldini Soares et al., 2019) (Shen and Huang, 2016).

This approach could also potentially reduce the necessity to retrain the model to address failure cases in unseen data by adding those words that may be informative to the model from these failed cases to our lookup. Since the marker token is used

System	Instance-F$_1$	Overall micro-F$_1$	Aggression micro-F$_1$	Gender Bias micro-F$_1$	Communal Bias micro-F$_1$
HIN-3-IND	0.3461 ± 0.0136	0.7004 ± 0.0119	0.6074 ± 0.0276	0.781 ± 0.0111	0.7128 ± 0.0287
HIN-JNT-3H	0.3832 ± 0.0317	0.7151 ± 0.0144	0.6142 ± 0.0203	0.7868 ± 0.0199	0.7443 ± 0.0211
HIN-JNT-3H+FE	0.3749 ± 0.0355	0.7131 ± 0.0133	0.6084 ± 0.0232	0.7894 ± 0.0161	0.7415 ± 0.0174
HIN-JNT-4H	0.383 ± 0.0337	0.7083 ± 0.0159	0.611 ± 0.0178	0.7591 ± 0.0258	0.7549 ± 0.0129

Table 3: Summary of results on the test set of the **Hindi** dataset averaged across runs on 5 random seeds for various approaches with **Hinglish-BERT**

System	Instance-F$_1$	Overall micro-F$_1$	Aggression micro-F$_1$	Gender Bias micro-F$_1$	Communal Bias micro-F$_1$
MULTI-JNT-3H	0.371	0.713	0.539	0.767	0.834

Table 4: Results on the test set of the **Multilingual** dataset with **XLM Roberta** as reported by the organizers on the leaderboard

to highlight a word from the lookup, it could act as a predictive signal in the data.

5.2.3 Joint Model for Tasks with Hierarchical Heads for Aggression Prediction (Four Heads)

In this approach, we break the task of aggression prediction into a *hierarchy of two binary classification sub-tasks*. We first predict whether the text is *aggressive* or *non-aggressive* (this sub-task is referred to as **A1**). Then we further predict each of the texts predicted as aggressive as being either *covertly* or *overtly aggressive* (this sub-task is referred to as **A2**).

Similar to Joint Model for Tasks (Three Heads), we jointly model each of the four tasks/sub-tasks (GEN, COM, A1, A2) using a single model architecture with a **common Hinglish-BERT backbone** and task-specific (or sub-task specific) heads (each corresponding to one of the tasks). Therefore, for each of the tasks, the prediction probabilities for the classes in the task are as given by Equation 5.

For the tasks **GEN** and **COM**, we compute class prediction probabilities $\hat{\mathbf{y}}_{GEN}$ and $\hat{\mathbf{y}}_{COM}$, and thereby compute the task-specific losses $\mathcal{L}_{GEN}$ and $\mathcal{L}_{COM}$ as in Joint Model for Tasks (Three Heads).

Further, for each of the samples $(x, y) \in \mathcal{D}$, we define the true annotation for the **A1** subtask, y_{A1}, as follows:

$$y_{A1} = \begin{cases} AG & \text{if } y_{AG} \in \{CAG, OAG\} \\ NAG & \text{if } y_{AG} = NAG \end{cases} \tag{7}$$

We then compute the prediction probabilities for the classes $\{AG, NAG\}$ in the sub-task **A1** using the common Hinglish-BERT with sub-task specific head $\mathbf{H}_{A1}$, and thereby compute the sub-task specific loss $\mathcal{L}_{A1}$ as the Cross Entropy loss of the prediction probabilities ($\hat{\mathbf{y}}_{A1}$) and true annotations (y_{A1}).

For the fourth sub-task **A2**, given a mini-batch (or dataset) $\mathcal{D}_{train}$ for training, we filter the samples for which the true aggression annotation y_{AG} belongs to $\{CAG, OAG\}$ as given by:

$$\mathcal{D}_{train,A2} = \{(x, y) \big| (x, y) \in \mathcal{D}_{train} \wedge \\ y_{AG} \in \{CAG, OAG\}\} \tag{8}$$

We then compute the sub-task specific loss $\mathcal{L}_{A2}$ for the mini-batch (or dataset) $\mathcal{D}_{train}$ using only the samples in $\mathcal{D}_{train,A2}$ (Eq. 8).

During training on mini-batch D_{train}, we compute the prediction probabilities for the classes $\{CAG, OAG\}$ in the sub-task **A2** for only the samples in $D_{train,A2}$ using the common Hinglish-BERT with sub-task specific head $\mathbf{H}_{A2}$, and thereby compute the sub-task specific loss $\mathcal{L}_{A2}$ as the *Cross Entropy* loss of the prediction probabilities ($\hat{\mathbf{y}}_{A2}$) and true annotations (y_{A2} which equals y_{AG})[3].

During training, given a mini-batch of samples D_{train}, we define the corresponding overall loss $\mathcal{L}$ which we optimize for in our model as:

$$\mathcal{L} = \left[\frac{|\mathcal{D}_{train}| \times (\mathcal{L}_{A1} + \mathcal{L}_{GEN} + \mathcal{L}_{COM}) + |\mathcal{D}_{train,A2}| \times \mathcal{L}_{A2}}{3 \times |\mathcal{D}_{train}| + |\mathcal{D}_{train,A2}|} \right] \tag{9}$$

[3] y_{A2} equals y_{AG} for samples in $D_{train,A2}$ as we only have samples with true class annotations **CAG** and **OAG** in it

where, for mini-batch D_{train}, $D_{train,A2}$ is as given in Equation 8.

6 Experimental Setup

We utilize PyTorch[4](Paszke et al., 2019) and the Transformers Library[4] from Hugging Face (Wolf et al., 2020) for implementing our systems. The code and resources used are made available on GitHub[5]. The systems we built and their respective notations are summarized in Table 2.

6.1 Preprocessing of texts

We preprocess each of the texts before feeding them to the models for both training/inferencing.

For the **Hindi** dataset, we preprocess the texts along the lines of (Bhange and Kasliwal, 2020) by performing the following transformations on them:

- Replace ”@” with ”mention”, ”#” with ”hashtag” and retweet related information in texts with the word ”Retweet”; remove http(s) URLs
- Convert emojis to their text equivalent using the emoji packages (Kim et al.)

For the **Multilingual** dataset, the preprocessing involves the removal of retweet related information, mentions of users, http(s) URLs and emojis.

6.2 Hyperparameters

The default hyperparameters we used while training all the systems unless mentioned otherwise below are summarized in the Table 5.

Hyperparameter	Value
Tokenizer max sequence length	128
Training batch Size	32
Learning rate	5e-5
Number of training epochs	10

Table 5: Default hyperparameters for the systems which are to considered unless specified otherwise

We used the AdamW (Loshchilov and Hutter, 2017) optimizer for training all our systems.

For the **HIN-JNT-3H+FE** system, we use a learning rate of 3e-4 for the parameters in the task-specific heads and a lower learning rate of 5e-5 for the parameters in the common Hinglish-BERT backbone.

For the **HIN-JNT-3H+FE** system, we train the models for 20 epochs.

We evaluate the model checkpoints for each of the systems after each epoch using the validation set and pick the checkpoint with the *best instance-*F_1 for joint modelling (**JNT**) systems and the checkpoint with the *best accuracy* for each of the individual models for individual modelling (**IND**) systems. We further evaluate this best model checkpoint which was picked on the test set, and report the scores.

For the **HIN-JNT-3H+FE** system, we used a publicly available lookup of profanity words from (pmathur5k10), (Mathur et al., 2018) in combination with a set of words that could be indicative of profanity or used in a profane manner, (tabulated in Table 7) which were manually curated by analyzing some of the samples from the corresponding train and validation splits.

6.3 Evaluation Metrics

The shared task uses instance-F_1 as the primary evaluation metric and overall micro-F_1 as the secondary evaluation metric for the systems[6].

7 Results

For the **Hindi set**, we initially performed one run of each of the systems and submitted the results to the leaderboard. The scores for these runs are present in the first row of Tables 9, 10, 11 and 12 [7]. In these runs, we observed that the **HIN-JNT-4H** system performed the best, followed by the **HIN-JNT-3H** system.

We further re-ran the systems four more times, with different seeds for each run to account for the impact of randomness in our systems' performances. In terms of instance-F_1, we observe that the joint modelling approaches often outperform the system of individually trained models across the runs. This is also evident in the mean scores reported in Table 3, and it highlights the potential benefits of jointly modelling the tasks.

However, when we further compare the performance within the different joint modelling approaches, we observe no clear winner under all circumstances, as the performance often varies with

[4] We use PyTorch library version 1.10.0+cu111 and Transformers library version 4.12.5

[5] https://github.com/BUDDI-AI/ComMA-ICON-2021

[6] https://competitions.codalab.org/competitions/35482#learn_the_details-evaluation

[7] The scores corresponding to the run submitted to the leaderboard are marked with an ”(L)” in the ”Run” column in each of these tables.

		Predicted		
		NAG	CAG	OAG
True	NAG	309	48	120
	CAG	31	13	41
	OAG	73	53	314

(a) Aggression

		Predicted	
		NGEN	GEN
True	NGEN	740	58
	GEN	134	70

(b) Gender Bias

		Predicted	
		NCOM	COM
True	NCOM	593	47
	COM	199	163

(c) Communal Bias

Table 6: **Confusion matrices** for **test set** predictions by the **HIN-JNT-3H** system's model corresponding to the **5^{th} run** for each of the three tasks of Aggression, Gender Bias and Communal Bias Identification

the random seed used as part of the run.

For the **Multilingual set**, we submitted only one system, **MULTI-JNT-3H**, whose results are presented in Table 4. This system jointly modeled all the three tasks using the approach from Joint Model for Tasks (Three Heads), and we observe that the model performs quite competitively.

7.1 Analysis

From Table 3, we observe that from among all the systems on the *Hindi dataset*, the **HIN-JNT-3H** system has the *best mean Instance-F_1 score across the 5 runs of the system on the test set*. Therefore, we pick the **HIN-JNT-3H** system and select the system's model corresponding to the run with the highest instance-F_1 (i.e., **Run 5** from Table 10). We then analyze the ***confusion matrices*** *of the selected model on the test set for each of the three tasks* (which are tabulated in Table 6).

7.1.1 Aggression Level Identification Task

For this task, we observe that out of the **85** samples with true class annotation **CAG**, only 13 samples **(15.3%)** are correctly predicted by the model as belonging to class **CAG**, whereas 31 samples **(36.47%)** are predicted as **NAG** and 41 samples **(48.24%)** are predicted as **OAG**. This indicates that the model may be struggling to sufficiently identify texts with subtle characteristics of aggression, and instead classifies them into one of the two extremes (**NAG** or **OAG**).

7.1.2 Gender Bias Identification Task

For this task, we observe that the model has a precision of **54.69%** and a recall of **34.31%** for the **GEN** class whereas it has a precision of **84.67%** and a recall of **92.73%** for the **NGEN** class. It indicates that the model performs better in accurately recalling and identifying non-gendered texts than recognizing gendered text.

7.1.3 Communal Bias Identification Task

For this task, the model has a precision of **77.62%** and a recall of **45.03%** on the **COM** class. It indicates that, while the model may face issues in retrieving all the communally-biased text samples (as indicated by its recall), the samples predicted as **COM** by the model are quite likely to be communally biased (as indicated by its precision).

7.1.4 Class Imbalance

As indicated by Table 8, it is clear that there is an imbalance in class distribution across the tasks in the train set of the Hindi data, which could account for some of the problems discussed previously. Techniques from the imbalanced learning literature, such as sampling or weighted loss functions, could be explored.

Conclusion

Thus, we present the description and analyses of the systems we submitted towards these tasks. Future extensions to this work could include assessing the performance of our systems across different folds of the data for more robust evaluation. The performance of other transformer-based models on the corpora could also be analyzed.

Acknowledgments

We would like to thank our colleagues in the Research and Development division and the management of BUDDI AI for their constant support and encouragement.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Meghana Bhange and Nirant Kasliwal. 2020. HinglishNLP at SemEval-2020 task 9: Fine-tuned language models for Hinglish sentiment detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 934–939, Barcelona (online). International Committee for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal Vishrav Chaudhary Guillaume Wenzek, Francisco Guzmán, Edouard Grave Myle Ott Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugging Face - XLM-Roberta-Base. xlm-roberta-base.

Nirant Kasliwal and Meghana Bhange. Nirantk/hinglish: Hinglish text classification.

Taehoon Kim, Kevin Wurster, and Tahir Jalilov. Emoji for python.

Ritesh Kumar, Bornini Lahiri, Akanksha Bansal, Enakshi Nandi, Laishram Niranjana Devi, Shyam Ratan, Siddharth Singh, Akash Bhagat, and Yogesh Dawer. 2021a. Comma@icon: Multilingual gender biased and communal language identification task at icon-2021. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON): COMMA@ICON 2021 Shared Task*, Silchar, India. NLP Association of India (NLPAI).

Ritesh Kumar, Enakshi Nandi, Laishram Niranjana Devi, Shyam Ratan, Siddharth Singh, Akash Bhagat, Yogesh Dawer, and Akanksha Bansal. 2021b. The comma dataset v0.2: Annotating aggression and bias in multilingual social media discourse.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium. Association for Computational Linguistics.

meghanabhange/Hinglish-Bert. meghanabhange/hinglish-bert.

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).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

pmathur5k10. pmathur5k10/hinglish-offensive-text-classification: Hinglish offensive text classification.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536, Osaka, Japan. The COLING 2016 Organizing Committee.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 23612364, New York, NY, USA. Association for Computing Machinery.

A Appendix

The words in Table 7 are added to the corpus in a purely research motivated manner since they are words that can potentially be used in a profane manner in text, and we investigate if they could aid systems in better learning to recognize instances of aggressive, communal or gender biased text.

Terms				
bc	mc			
RANDI	RANDY	hore	Dalla	bs
hijra	gay			
chod	chutiya	chutiyo		
pussy	Gand	G**d	Lawde	Lowde
OLAD	harami	bootlicker		
हरामी	पिछवाड़े	भड़वे	गांडू	

Table 7: Set of code mixed Hindi words which are potentially informative for the tasks, and which were manually curated by analyzing some samples from the train and validation splits

Class	# Samples
NAG	1289
CAG	800
OAG	2526
NGEN	3665
GEN	950
NCOM	3598
COM	1017

Table 8: Distribution of classes across tasks for the **train** split of **Hindi** dataset

Run	**Instance-F$_1$**	**Overall micro-F$_1$**	**Aggression micro-F$_1$**	**Gender Bias micro-F$_1$**	**Communal Bias micro-F$_1$**
1 (L)	0.3453	0.6979	0.6457	0.7695	0.6786
2	0.3603	0.7083	0.5988	0.7745	0.7515
3	0.3433	0.6929	0.5768	0.7884	0.7136
4	0.3253	0.6866	0.5908	0.7764	0.6926
5	0.3563	0.7162	0.6248	0.7964	0.7275

Table 9: Results on the test set of the **Hindi** dataset using the **HIN-3-IND system**

Run	**Instance-F$_1$**	**Overall micro-F$_1$**	**Aggression micro-F$_1$**	**Gender Bias micro-F$_1$**	**Communal Bias micro-F$_1$**
1 (L)	0.3603	0.6946	0.6188	0.7585	0.7066
2	0.3972	0.7242	0.6257	0.7924	0.7545
3	0.3473	0.7112	0.5818	0.7994	0.7525
4	0.3832	0.7129	0.6098	0.7754	0.7535
5	0.4281	0.7325	0.6347	0.8084	0.7545

Table 10: Results on the test set of the **Hindi** dataset using the **HIN-JNT-3H system**

Run	**Instance-F$_1$**	**Overall micro-F$_1$**	**Aggression micro-F$_1$**	**Gender Bias micro-F$_1$**	**Communal Bias micro-F$_1$**
1 (L)	0.3413	0.7006	0.5978	0.7794	0.7246
2	0.3683	0.7112	0.5998	0.7784	0.7555
3	0.3473	0.7006	0.5998	0.7754	0.7265
4	0.3882	0.7226	0.5948	0.8094	0.7635
5	0.4291	0.7305	0.6497	0.8044	0.7375

Table 11: Results on the test set of the **Hindi** dataset using the **HIN-JNT-3H+FE system**

Run	**Instance-F$_1$**	**Overall micro-F$_1$**	**Aggression micro-F$_1$**	**Gender Bias micro-F$_1$**	**Communal Bias micro-F$_1$**
1 (L)	0.3982	0.7092	0.6277	0.7425	0.7575
2	0.3932	0.7169	0.6038	0.7914	0.7555
3	0.4242	0.7295	0.6317	0.7824	0.7745
4	0.3373	0.6949	0.6008	0.7435	0.7405
5	0.3623	0.691	0.5908	0.7355	0.7465

Table 12: Results on the test set of the **Hindi** dataset using the **HIN-JNT-4H system**