Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 76–82
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Aggression Identification in English, Hindi and Bangla Text using BERT, RoBERTa and SVM

Arup Baruah[♢], Kaushik Amar Das[♢], Ferdous Ahmed Barbhuiya[♢], Kuntal Dey[♡]
[♢]IIIT Guwahati,[♡] IBM Research
[♢]Assam India, [♡]New Delhi India
{arup.baruah, kaushikamardas}@gmail.com,
ferdous@iiitg.ac.in, kuntadey@in.ibm.com

Abstract

This paper presents the results of the classifiers that the team 'abaruah' developed for the shared tasks in aggression identification and misogynistic aggression identification. These two shared tasks were held as part of the second workshop on Trolling, Aggression and Cyberbullying (TRAC). Both the subtasks were held for English, Hindi and Bangla language. In our study, we used English BERT (En-BERT), RoBERTa, DistilRoBERTa, and SVM based classifiers for the English language. For Hindi and Bangla language, multilingual BERT (M-BERT), XLM-RoBERTa and SVM classifiers were used. Our best performing models are EN-BERT for English Subtask A (Weighted F1 score of 0.73, Rank 5/16), SVM for English Subtask B (Weighted F1 score of 0.87, Rank 2/15), SVM for Hindi Subtask A (Weighted F1 score of 0.79, Rank 2/10), XLMRoBERTa for Hindi Subtask B (Weighted F1 score of 0.87, Rank 2/10), SVM for Bangla Subtask A (Weighted F1 score of 0.81, Rank 2/10), and SVM for Bangla Subtask B (Weighted F1 score of 0.93, Rank 4/8). It is seen that the superior performance of the SVM classifier was achieved mainly because of its better prediction of the majority class. BERT based classifiers were found to predict the minority classes better.

Keywords: Aggression Identification, Offensive Language, Multilingual, BERT, SVM, RoBERTa

1. Introduction

Partisan antipathy in politics is on the rise. All over the world, societies are getting more and more politically polarized (Thomas Carothers, 2019). It is partly fuelled by the *echo chamber* and *filter bubble* effect of social media. Anger is fast becoming a tool to lure voters. As the world gets polarized, the popularity and convenience of the social media platforms are turning them to a modern-day battlefield. This has led to an increase in aggressive content in social media. Some of the world leaders are also using social media as a platform for displaying their aggressiveness. An example of this is the following tweet addressed to North Korean leader Kim Jong-un by U.S. President Donald Trump, *"Will someone from his depleted and food starved regime please inform him that I too have a Nuclear Button, but it is a much bigger & more powerful one than his, and my Button works!"*

Social media sites are grappling to remove aggressive content from their sites both to promote healthy discussions and also to comply with legal laws. However, the scale involved makes manual moderation a difficult task. The need of the hour is automated methods for detecting aggressive content.

The second workshop on Trolling, Aggression, and Cyberbullying (TRAC-2) (Kumar et al., 2020) is an attempt to promote research in automated detection of aggression in text. This workshop had two shared tasks titled *"Aggression Identification" (Subtask A)* and *"Misogynistic Aggression Identification" (Subtask B)*. Aggression identification is a 3-way classification problem where it is required to determine if a given comment is overtly, covertly or not aggressive. Misogynistic aggression is a binary classification problem where it is required to determine if the comment is gender-based or not. Both the subtasks were held for English, Hindi, and Bangla language.

We participated in both the subtasks for all the three languages. The classifiers we used in this study include En-BERT, M-BERT, RoBERTa, DistilRoBERTa, and XLM-RoBERTa.

2. Related Work

Apart from automatic detection of aggression in text, considerable research has been performed for detection of offensive language, abusive language, hate speech, cyberbullying, profanity, and insults. Fortuna and Nunes (2018) provides definitions of the terms mentioned above, provides statistics of research performed for the detection of hate speech, lists the features, classification methods, and challenges in automated hate speech detection. Schmidt and Wiegand (2017) too discusses the different classification methods, features and the challenges involved in the detection of hate speech.

Davidson et al. (2017) mentions that not all offensive language is hate speech. Their classifier was able to reduce the number of offensive tweets misclassified as hate speech to 5%. Malmasi and Zampieri (2017) worked on differentiating hate speech from profanity by using an SVM classifier trained on features such as character n-grams (2 to 8), word n-grams (1 to 3), and word skip-grams. Malmasi and Zampieri (2018) extended the above work to include Brown cluster features, ensemble classifiers and meta-classifiers in addition to single classifiers.

Zampieri et al. (2019a) introduces a new dataset called Offensive Language Identification Dataset (OLID) where the data has been categorized as offensive or not, targeted or untargeted, and targets individual, group or other. SVM, BiLSTM and CNN classifiers were used in this study to predict the type and target of offensive posts. Zampieri et al. (2019b) summarizes the results from the shared task on

Language	Type	Total	NAG	CAG	OAG	NGEN	GEN	Max Length	Length below 50 words
English	Train	4263	3375 (79.17%)	453 (10.63%)	435 (10.20%)	3954 (92.75%)	309 (7.25%)	806	93.31%
English	Dev	1066	836 (78.42%)	117 (10.98%)	113 (10.60%)	993 (93.15%)	73 (6.85%)	457	93.34%
English	Test	1200	690 (57.50%)	224 (18.67%)	286 (23.83%)	1025 (85.42%)	175 (14.58%)	1390	77.41%
Hindi	Train	3984	2245 (56.35%)	829 (20.81%)	910 (22.84%)	3323 (83.41%)	661 (16.59%)	557	95.41%
Hindi	Dev	997	578 (57.97%)	211 (21.16%)	208 (20.86%)	845 (84.75%)	152 (15.26%)	230	93.98%
Hindi	Test	1200	325 (27.08%)	191 (15.92%)	684 (57.00%)	633 (52.75%)	567 (47.25%)	669	89.92%
Bangla	Train	3826	2078 (54.31%)	898 (23.47%)	850 (22.22%)	3114 (81.39%)	712 (18.61%)	154	98.64%
Bangla	Dev	957	522 (54.55%)	218 (22.78%)	217 (22.68%)	766 (80.04%)	191 (19.96%)	182	98.64%
Bangla	Test	1188	712 (59.93%)	225 (18.94%)	251 (21.13%)	986 (83.00%)	202 (17.00%)	113	99.24%

Table 1: Dataset Statistics

identification and categorization of offensive language held as part of Semantic Evaluation 2019. The best performing system in subtask A of OffensEval 2019 used a BERT based model (Liu et al., 2019b) and obtained a macro F1 score of 0.8286. Zhu et al. (2019) also used a BERT based model and obtained the 3rd rank in subtask A of OffensEval 2019 with a macro F1 score of 0.8136.

The results of the TRAC-1 has been summarized in Kumar et al. (2018). As can be seen, both deep learning (LSTM, BiLSTM, CNN) and traditional machine learning classifiers (SVM, Logistic Regression, Random Forest, Naive Bayes) were used in this shared task.

Similarly, the HASOC [1] (Mandl et al., 2019) workshop organized at FIRE2019 was also aimed at stimulating research the aforementioned areas in Hindi, English and German languages respectively. They note that the most widely used approach was LSTMs coupled with word embeddings. In this workshop, the participants used a wide variety of models such as BERT, SVM, CNN, LSTM with Attention, etc.

3. Data

The dataset for subtask A has been labelled as either overtly aggressive (OAG), covertly aggressive (CAG) or not aggressive (NAG). The dataset for subtask B has been labelled as gendered (GEN) or non-gendered (NGEN). The dataset is further described in Bhattacharya et al. (2020).

Table 1 shows the statistics of the dataset used for the two shared tasks. As can be seen, the dataset is imbalanced with NAG (for subtask A) and NGEN (for subtask B) occurring more frequently in all the three languages. The NGEN category occurred as high as 93.15% in the English development dataset. This, however, is a true reflection of the proportion of aggressive and non-aggressive comments in real life as has been mentioned in Gao et al. (2017). The only exception is the Hindi test dataset. In this dataset, OAG is the most frequently occurring class for subtask A and this dataset is almost balanced for subtask B.

As can be seen, the comments were also of varied length (in terms of the number of words). The longest comment of 1390 words occurred in the English test dataset. However, as can be seen from the table, the majority of the comments were of length less than 50 words.

4. Methodology

4.1. Preprocessing

In our work, before performing tokenization, the text was converted to lower case. This conversion to lower-case was performed through the BERT tokenizer and the TFIDF vectorizer. As mentioned in section 3, except for English and Hindi test set, more than 93% of the comments were of length less than 50 tokens. Hence, for En-BERT and M-BERT, the maximum sequence length of 50 was used. Comments of length beyond 50 tokens were truncated. In the RoBERTa models, the long sentences were split into multiple samples [3].

4.2. Classifiers

4.2.1. English BERT (En-BERT)

English BERT (Devlin et al., 2019) is a bi-directional model based on the transformer architecture. The transformer architecture is an architecture based solely on attention mechanism (Vaswani et al., 2017). The transformer architecture overcomes the inherent sequential nature of Recurrent Neural Networks (RNN) and hence they are more conducive for parallelization.

In our study, we used the uncased large version of En-BERT [2]. This version has 24 layers and 16 attention heads. This

[1] https://hasocfire.github.io/hasoc/2019/

[2] https://github.com/google-research/bert

model generates 1024 dimensional vector for each word. We used 1024 dimensional vector of the Extract layer as the representation of the comment. Our classification layer consisted of a single Dense layer.

For subtask A, the dense layer consisted of 3 units and the *softmax* activation function was used. The loss function used was *sparse categorical crossentropy*. For subtask B, the dense layer consisted of 1 unit and the *sigmoid* activation function was used. The loss function used was *binary crossentropy*. The *Adam* optimizer with a learning rate of 2e-5 was used for training the model. The model was trained for 15 epochs. Early stopping with patience of 5 was used for both the subtasks. *Sparse categorical accuracy* was monitored for early stopping.

4.2.2. Multilingual BERT (M-BERT)

Multilingual BERT is BERT trained for multilingual tasks. It was trained on monolingual Wikipedia articles of 104 different languages. It is intended to enable M-BERT fine-tuned in one language to make predictions for another language. In our study, we used the M-BERT model having 12 layers and 12 heads. This model generates 768 dimensional vector for each word. We used the 768 dimensional vector of the Extract layer as the representation of the comment. Just like for the English language subtasks, a single Dense layer was used as the classification model. The hyperparameters used for training the model is the same as mentioned for the English language.

Algorithm 1 Naive Checkpoint Ensemble

$A \leftarrow$ True labels
$P \leftarrow$ Model predictions at each epoch
$N \leftarrow$ Num samples, $C \leftarrow$ Num classes
$reverse \leftarrow boolean$
function ENSEMBLE($P, A, N, C, reverse$)
    $models \leftarrow \{\}, val \leftarrow 0$
    $Z[N][C] \leftarrow$ Zero Matrix
    $\epsilon \leftarrow len(P)$ ▷ Num Epochs
    if $reverse$ **then**
        $range \leftarrow \epsilon$ to 0
    else
        $range \leftarrow 0$ to ϵ
    end if
    for $(e \leftarrow range)$ **do**
        $temp \leftarrow Z$
        $temp \leftarrow temp + P[e]$
        if $metric(A, temp) > val$ **then**
            $Z \leftarrow Z + P$
            $models \leftarrow models \cup e$
            $val \leftarrow metric(A, temp)$
        else
            continue
        end if
    end for
    return $models, val$
end function

4.2.3. RoBERTa and DistilRoBERTa

RoBERTa (Liu et al., 2019c) improves upon BERT by adding a few modifications to the original model such as training on a larger dataset, dynamically masking out tokens compared to the original static masking, etc. DistilRoBERTa (Sanh et al., 2019) is a compressed version of the same which trains faster and preserves up to 95% of the performance of the original. For both of these models, we make use of the pre-trained *base* versions made available by the HuggingFace Transformers library (Wolf et al., 2019). We make use of the RoBERTa model for English Task A and DistilRoBERTa for English Task B. We use an attention layer (Zhou et al., 2016) on top of the embeddings of the underlying pre-trained model. However, instead of the $tanh$ activation function used in the original work, we used $penalized - tanh$ which is demonstrated to work better for NLP tasks (Eger et al., 2019) combined with a cross-entropy loss function. We also do not apply $softmax$ on the output of the classifying layer as done in the original work and instead use $argmax$ directly on the final layer outputs to make the prediction. We make use of the Ranger Optimizer which is a combination of RAdam (Liu et al., 2019a) wrapped with Lookahead (Zhang et al., 2019) to train the model. The entire model is fine-tuned with a tiny learning rate of $1e-4$ for both of the English classification tasks. For task A and task B, lookahead's (k, α) is set to $(5, 0.5)$ and $(6, 0.5)$ with a weight decay of $1e-5$ respectively. The models were set to run for 20 epochs with early stopping patience of 4. We made use of a naive checkpoint ensembling method (Chen et al., 2017) where we save the model weights and dev-set predictions (i.e. the final layer output) at each epoch. The method is given in Algorithm 1. The method is called once with $reverse$ set to $True$ and once with $False$. The ensembled model which maximize our chosen metric (weighted–f1) value is chosen. If the ensemble does not improve the metric, we simply choose the best model found during training. Once we have chosen the model, we use Algorithm 2 to make the final prediction on the test set. This Algorithm 2 simply describes adding the weights of the final classifying layer of the model and using argmax along each row to get the prediction. Naive ensembling increases the weighted f1 on the dev–set on English task A from 0.8070 to 0.8124. We did not use it for English task B as it degraded the performance.

Algorithm 2 Make Prediction

$m \leftarrow$ model ids chosen for ensemble
$E[N][C] \leftarrow$ Zero Matrix
for i in m **do**
    Load model with weights at epoch i
    $p \leftarrow model.predict(samples)$
    $E \leftarrow E + p$
end for
$preds \leftarrow$ Index of max element in each row of N

4.2.4. XLM-RoBERTa

XLM-RoBERTa (Conneau et al., 2019) is a cross-lingual model that aims to tackle the *curse-of-multilinguality* problem of cross-lingual models. It is inspired by (Liu et al., 2019c) and is trained on up-to 100 languages and outperforms M-BERT in multiple cross-lingual benchmarks.

Similar to Section 4.2.3, we use[3] the *base* version coupled with an attention head classifier, the same optimizer, epochs, and early stopping. Lookahead's (k, α) is set to $(6, 0.5)$ with weight–decay of $1e-5$. Batch-size is set to (22,24) for Bangla tasks (A, B) and 32 for both Hindi tasks. This model is used in the sub-tasks of the Hindi and Bangla languages. For the Hindi models, we use the naive checkpoint ensembling method described in Section 4.2.3. This increased the weighted f1 from 0.7146 to 0.7160 for Hindi task A and from 0.8908 to 0.8969 for Hindi task B. Naive ensembling did not yield any performance boosts in Bangla tasks.

4.2.5. SVM

We also used the Support Vector Machine (SVM) model for both the subtasks in all the 3 languages. The SVM model was trained using TF-IDF features of word and character n-grams. Word n-grams of size 1 to 3 and character n-grams of size 1 to 6 were used. The *linear* kernel was used for the classifier and hyperparameter C was set to 1.0.

5. Results

As has been mentioned in section 4, the classifiers we used include En-BERT, RoBERTa, DistilRoBERTa and SVM for the subtasks in the English language, and M-BERT, XLM-RoBERTa and SVM for the subtasks in Hindi and Bangla language.

Table 2 and 3 show the results we obtained on the development and test set respectively. Both the table shows the precision, recall, macro F1, weighted F1, and accuracy. Weighted F1 score is the metric that has officially been used to rank the submissions. As can be seen from table 2, the best performing classifiers on the development set were RoBERTa for English subtask A, En-BERT for English subtask B, XLM-RoBERTa for Hindi subtask A, Bangla subtask A, and Bangla subtask B, and M-BERT for Hindi subtask B.

As can be seen from table 3, the SVM classifier which was not the best on the development set, actually performed well on the test set for English subtask B (ranked 2^{nd}), Hindi subtask A (ranked 2^{nd}), Bangla subtask A (ranked 2^{nd}), and Bangla subtask B (ranked 4^{th}). The other best-performing classifiers are En-BERT for English subtask A (ranked 5^{th}), and XLM-RoBERTa for Hindi subtask B (ranked 2^{nd}). The results of M-BERT for Hindi subtask A are not shown as an error was made for this run (binary classification was performed instead of performing 3-class classification).

It can also be seen from table 3 that for subtask B, the best performance of all the classifiers (SVM, BERT-based, and RoBERTa-based) was obtained for the Bangla language. For subtask B, the SVM classifier had the weighted F1 score of 0.87, 0.84 and 0.92, the RoBERTa-based classifiers had a score of 0.86, 0.87 and 0.92, and the BERT-based classifiers had a score of 0.85, 0.84 and 0.92 for English, Hindi and Bangla language respectively. Even for subtask A, the classifiers obtained better score for the Bangla language (except for RoBERTa-based classifier which obtained a slightly better score for Hindi language as compared to Bangla language).

The confusion matrices of the classifiers on the test set are shown in table 4 to 9. As can be seen from table 4, the strength of En-BERT which was our best performing classifier for English subtask A, was that it predicted the minority classes better than the other two classifiers. In fact, it was the worst in predicting the majority NAG class. But because of its correct predictions for the minority classes, it was our best performing classifier for this subtask. RoBERTa too predicted the OAG class better than SVM. However, RoBERTa did not perform well in predicting the CAG class. Detecting covertly aggressive comments is very difficult and En-BERT performed better than the other two classifiers in predicting this class.

As can be seen from table 7, SVM which was our best performing classifier for English subtask B, predicted the majority class better than the other two classifiers. SVM, however, was the worst in predicting the minority class. En-BERT again was the best in predicting the minority class. En-BERT also had the best recall score for this subtask.

As mentioned in section 3, for Hindi subtask A, OAG was the majority class. XLM-RoBERTa performed better than SVM in predicting the majority class. However, SVM performed better in predicting the CAG and NAG class and hence was the best performing classifier in this subtask. For Hindi subtask B, the dataset was quite balanced, and in this dataset, XLM-RoBERTa performed the best.

For Bangla subtask A, SVM performed the best in predicting the majority NAG class as well as the CAG class. As such, it was the best performing classifier in this subtask. For Bangla subtask B, SVM again performed better in predicting the majority class. In this subtask, M-BERT and XLM-RoBERTa performed better than SVM in predicting the minority class. The best performing classifier for this subtask was SVM.

6. Error Analysis

On analysis of the predictions made by our classifiers on the development set, we found that our classifiers were not able to handle intentional or unintentional orthographic variations of toxic words and spelling mistakes. For example, both the SVM and En-BERT classifiers wrongly classified the comment *"Fuuck your music"* as not aggressive. This comment has been labelled by the annotators as overtly aggressive. However, after changing the toxic word *'Fuuck'* to *'Fuck'*, both the classifiers were able to make the correct prediction for the comment. Similarly, both the classifiers were not able to handle the spelling mistake for the word *'prostitute'* in the comment *'So sad she is a professional prostatiut'*. The comment was wrongly classified as not gendered. After correcting the spelling mistake, both the classifiers were able to classify the comment correctly.

Annotators have labelled comments such as *'Im homosexual and really proud of it'* and *'I. Gay'* where the user is attributing homosexuality to oneself as not gendered. However, our SVM wrongly classifies these comments as gendered based on the presence of the words *homosexual* and *gay*. So, the SVM classifier has not been able to detect the

[3]Code for this particular model available at `https://github.com/cozek/trac2020_submission`

Task	System	Precision (Macro)	Recall (Macro)	F1 (macro)	F1 (weighted)	Accuracy
English A	SVM	0.6415	0.4807	0.5170	0.7729	0.8105
English A	RoBERTa	**0.6418**	0.5883	**0.6106**	**0.8070**	**0.8148**
English A	En-BERT	0.5866	**0.5884**	0.5871	0.7878	0.7858
English B	SVM	0.8060	0.6056	0.6490	0.9244	0.9390
English B	DistilRoBERTa	0.7201	0.6866	0.7016	0.9260	0.9289
English B	En-BERT	**0.8274**	**0.6962**	**0.7423**	**0.9400**	**0.9467**
Hindi A	SVM	**0.6682**	0.6249	0.6409	0.7074	0.7192
Hindi A	XLM-RoBERTa	0.6602	**0.6376**	**0.6472**	**0.7146**	**0.7207**
Hindi A	M-BERT	0.6147	0.6167	0.6151	0.6846	0.6871
Hindi B	SVM	**0.8415**	0.6906	0.7346	0.8765	0.8917
Hindi B	XLM-RoBERTa	0.8125	0.7565	0.7801	0.8908	**0.8959**
Hindi B	M-BERT	0.7977	**0.7781**	**0.7874**	**0.8919**	0.8937
Bangla A	SVM	0.7096	0.6557	0.6747	0.7197	0.7304
Bangla A	XLM-RoBERTa	**0.7203**	**0.7121**	**0.7137**	**0.7539**	**0.7513**
Bangla A	M-BERT	0.6805	0.6891	0.6844	0.7279	0.7252
Bangla B	SVM	**0.8792**	0.7396	0.7826	0.8723	0.8851
Bangla B	XLM-RoBERTa	0.8580	**0.8319**	**0.8439**	**0.9020**	**0.9039**
Bangla B	M-BERT	0.8585	0.7998	0.8242	0.8920	0.8966

Table 2: Dev Set Results

Task	System	Precision (Macro)	Recall (Macro)	F1 (macro)	F1 (weighted)	Accuracy	Rank
English A	SVM	**0.7923**	0.6077	0.6489	0.7173	**0.7450**	
English A	RoBERTa	0.6722	0.5921	0.6130	0.6986	0.7233	
English A	En-BERT	0.6880	**0.6415**	**0.6501**	**0.7289**	0.7350	5th
English B	SVM	**0.7980**	0.6744	0.7121	**0.8701**	**0.8850**	2nd
English B	DistilRoBERTa	0.7277	0.7101	**0.7183**	0.8623	0.8650	
English B	En-BERT	0.6980	**0.7226**	0.7089	0.8503	0.8458	
Hindi A	SVM	**0.7252**	**0.7592**	**0.7363**	**0.7944**	0.7867	2nd
Hindi A	XLM-RoBERTa	0.7129	0.7269	0.7188	0.7927	**0.7892**	
Hindi B	SVM	0.8597	0.8373	0.8395	0.8408	0.8433	
Hindi B	XLM-RoBERTa	**0.8704**	**0.8673**	**0.8683**	**0.8689**	**0.8692**	2nd
Hindi B	M-BERT	0.8395	0.8363	0.8372	0.8379	0.8383	
Bangla A	SVM	**0.8385**	**0.7171**	**0.7586**	**0.8083**	**0.8199**	2nd
Bangla A	XLM-RoBERTa	0.7434	0.7136	0.7264	0.7880	0.7938	
Bangla A	M-BERT	0.7265	0.6945	0.7074	0.7740	0.7820	
Bangla B	SVM	**0.9299**	0.8167	0.8600	**0.9258**	**0.9310**	4th
Bangla B	XLM-RoBERTa	0.8431	0.8617	0.8519	0.9153	0.9141	
Bangla B	M-BERT	0.8619	**0.8648**	**0.8633**	0.9227	0.9226	

Table 3: Official Results on Test Set

	SVM			RoBERTa			En-BERT		
	Pred CAG	**Pred NAG**	**Pred OAG**	**Pred CAG**	**Pred NAG**	**Pred OAG**	**Pred CAG**	**Pred NAG**	**Pred OAG**
True CAG	86	135	3	64	132	28	122	83	19
True NAG	3	677	10	26	645	19	48	624	18
True OAG	26	129	131	38	89	159	97	53	136

Table 4: Confusion Matrix on Test Set for English Subtask A

	SVM			XLM-RoBERTa		
	Pred CAG	**Pred NAG**	**Pred OAG**	**Pred CAG**	**Pred NAG**	**Pred OAG**
True CAG	121	52	18	101	53	37
True NAG	42	273	10	54	257	14
True OAG	64	70	550	46	49	589

Table 5: Confusion Matrix on Test Set for Hindi Subtask A

	SVM			XLM-RoBERTa			M-BERT		
	Pred CAG	**Pred NAG**	**Pred OAG**	**Pred CAG**	**Pred NAG**	**Pred OAG**	**Pred CAG**	**Pred NAG**	**Pred OAG**
True CAG	116	101	8	115	82	28	100	90	35
True NAG	14	691	7	42	647	23	53	645	14
True OAG	16	68	167	33	37	181	26	41	184

Table 6: Confusion Matrix on Test Set for Bangla Subtask A

	SVM		RoBERTa		En-BERT	
	Pred GEN	**Pred NGEN**	**Pred GEN**	**Pred NGEN**	**Pred GEN**	**Pred NGEN**
True GEN	66	109	86	89	96	79
True NGEN	29	996	73	952	106	919

Table 7: Confusion Matrix on Test Set for English Subtask B

	SVM		XLM-RoBERTa		M-BERT	
	Pred GEN	**Pred NGEN**	**Pred GEN**	**Pred NGEN**	**Pred GEN**	**Pred NGEN**
True GEN	413	154	473	94	453	114
True NGEN	34	599	63	570	80	553

Table 8: Confusion Matrix on Test Set for Hindi Subtask B

	SVM		XLM-RoBERTa		M-BERT	
	Pred GEN	**Pred NGEN**	**Pred GEN**	**Pred NGEN**	**Pred GEN**	**Pred NGEN**
True GEN	130	72	158	44	157	45
True NGEN	10	976	58	928	47	939

Table 9: Confusion Matrix on Test Set for Bangla Subtask B

benign use of these words. The En-BERT classifier however correctly classified these comments correctly as not gendered.

Our classifiers were not able to correctly classify comments such as *'There are only 2 genders'* that require world knowledge. The above comment was labelled by the annotators as gendered. However, because of the absence of any toxic words, the above comment was classified by both the SVM and En-BERT classifier as not gendered.

There were also certain comments such as *'Hot'* that were labelled as gendered by the annotators. These comments are ambiguous and can belong to either of the two categories. Most likely, these comments we labelled so based on some contextual information. In the absence of contextual information, our classifiers did not classify these comments correctly.

7. Conclusion

We used BERT, RoBERTa and SVM based classifiers for detection of aggression in English, Hindi and Bangla text. Our SVM classifier performed remarkably well on the test set and obtained 2nd rank in the official results for 3 of the 6 tests and obtained 4th in another. However, on closer analysis, it is seen that the superior performance of the SVM classifier was mainly due to the better prediction of the majority class. BERT based classifiers were found to predict the minority classes better. It was also found that our classifiers did not handle spelling mistakes and intentional orthographic variations correctly. FastText word embeddings are better in handling orthographic variations. As a future study, it can be checked if FastText embeddings improve performance on this dataset. Another option would be to use automatic methods for correcting grammatical and spelling mistakes. Use of contextual information and world knowledge for automatic detection of aggression needs further investigation.

8. Bibliographical References

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression.

Chen, H., Lundberg, S., and Lee, S.-I. (2017). Checkpoint ensembles: Ensemble methods from a single training process.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional trans-

formers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Eger, S., Youssef, P., and Gurevych, I. (2019). Is it time to swish? comparing deep learning activation functions across nlp tasks.

Fortuna, P. and Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Gao, L., Kuppersmith, A., and Huang, R. (2017). Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach. In *IJCNLP 2017*, pages 774–782, Taipei, Taiwan.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018). Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2020). Evaluating Aggression Identification in Social Media. In Ritesh Kumar, et al., editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*, Paris, France, may. European Language Resources Association (ELRA).

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019a). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Liu, P., Li, W., and Zou, L. (2019b). NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019c). Roberta: A robustly optimized bert pretraining approach.

Malmasi, S. and Zampieri, M. (2017). Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

Malmasi, S. and Zampieri, M. (2018). Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., and Patel, A. (2019). 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*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Schmidt, A. and Wiegand, M. (2017). A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Thomas Carothers, A. O. (2019). How to Understand the Global Spread of Political Polarization. `https://carnegieendowment.org/2019/10/01/how-to-understand-global-spread-of-political-polarization-pub-79893`. [Online; accessed 15-April-2020].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019a). Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019b). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Zhang, M. R., Lucas, J., Hinton, G., and Ba, J. (2019). Lookahead optimizer: k steps forward, 1 step back.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August. Association for Computational Linguistics.

Zhu, J., Tian, Z., and Kübler, S. (2019). UM-IU@LING at SemEval-2019 task 6: Identifying offensive tweets using BERT and SVMs. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 788–795, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.