IIITG-ADBU at HASOC 2019: Automated Hate Speech and Offensive Content Detection in English and Code-Mixed Hindi Text

Arup Baruah¹, Ferdous Ahmed Barbhuiya¹, and Kuntal Dey²

 Dept. of Comp. Sc. & Engg., IIIT Guwahati, India arup.baruah@gmail.com, ferdous@iiitg.ac.in
 IBM Research India, New Delhi, India, kuntadey@in.ibm.com

Abstract. This paper presents the results obtained by using Logistic Regression (LR), Support Vector Machine (SVM), bi-directional long short-term memory (BiLSTM) and Neural Network (NN) models for subtask A of the shared task "Hate Speech and Offensive Content Identification in Indo-European Languages" (HASOC). This paper presents the results for English and code-mixed Hindi language. Embeddings from Language Models (ELMo), Glove and fastText embeddings, and TF-IDF features of character and word n-grams have been used to train the models. Our best models for Hindi and English language obtained F1 score of 81.05 and 74.62 respectively on the official run. The models obtained the 4th and 8th position in the official ranking.

Keywords: Hate Speech \cdot Logistic Regression \cdot Support Vector Machine \cdot Bi-directional Long Short-Term Memory, Glove, fastText, ELMo

1 Introduction

Social media has made it easier for people to communicate with one another. Publishing content to reach a vast number of people has become very easy. However, among the constructive dialogs that take place in social media, there are also a few negative things that are happening in social media. Content that is hateful, offensive or profane is also being published. Such content are harmful for the society. There are evidences where hateful content published via social media has fueled communal riots in different parts of the world.

There has been a growing interest among the research communities to use machine learning and natural language processing techniques to automatically detect hateful and offensive content. As a step towards this direction, the shared

Copyright \odot 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). FIRE 2019, 12-15 December 2019, Kolkata, India.

task "Hate Speech and Offensive Content Identification in Indo-European Languages" (HASOC) has been organized [7]. This paper presents the results obtained by our models for subtask A of HASOC. The goal of subtask A is to detect if a given tweet is free from hateful and offensive content or not.

2 Related Work

Automated detection of offensive, hateful, abusive, aggressive, and profane text has seen the use of rule-based, traditional machine learning, and deep learning techniques. Risch and Krestel [9] used a LR classifier to detect abusive language. Features such as word and character n-grams, word2vec embeddings, word and character count etc. were used in the study. Waseem [12] used SVM and LR classifier to detect racist or sexist content. Nobata et al. [8] used a regression model to detect abusive content. Djuric et al. [3] used a LR classifier to detect hate speech. Among other features, this study used comment embeddings as features. Serra et al. [11] used a character-based RNN to detect hate speech in tweets. Gamback and Sikdar [4] used a CNN to detect racist and sexist content. Badjatiya et al. [1] experimented with LR, SVM, Gradient Boosted Decision Tree (GDBT), CNN, LSTM and FastText based models. Study on hate speech detection in code-mixed Hindi-English data has been performed in Mathur et al. [6], Santosh and Aravind [10], and Kamble and Joshi [5].

3 Dataset

The dataset for Subtask A of HASOC has been labeled as either free from hateful, offensive and profane content or not. Trial, train and test datasets were released for the subtask. Table 1 below shows the details of the dataset for both English and Hindi. As can be seen from the table, the percentage of hate, offensive or profane content was more in the English trial dataset compared to the English train dataset. For Hindi, the distribution of hate and non-hate content was identical in both trial and train dataset. The Hindi dataset was more balanced compared to the English dataset.

It was observed that performance of the models used in this study improved when English trial and train datasets were combined for training the models. However, combining the Hindi trial and train dataset decreased the performance of the models. Thus, only the train dataset was used for training the models for Hindi.

4 Methodology

4.1 Preprocessing

We experimented by removing the URLs, hashtags, and mentions from the English dataset. However, we found that removing each of them degraded the performance of our models. Thus, for our final models the dataset was used as was provided without performing any preprocessing.

Table 1. Data set statistics

Language	Type	Not Hate/Offensive/Profane	Hate/Offensive/Profane	Total
English	Trial	208 (41.19%)	297 (58.81%)	505
English	Train	3591 (61.36%)	2261 (38.64%)	5852
English	Test	Not Known	Not Known	1153
Hindi	Trial	64 (47.06%)	72 (52.94%)	136
Hindi	Train	2196 (47.07%)	2469 (52.93%)	4665
Hindi	Test	Not Known	Not Known	1318

4.2 Word and Sentence Embeddings

In our study, we used Embeddings from Language Models (ELMo), Glove, and fastText embeddings. The Glove and fastText embeddings were used to train our BiLSTM model. ELMo was used to train a simple neural network classifier. The 200 dimensional pre-trained Glove embeddings for Twitter dataset was used. The Glove embeddings were used only for the English language models. The fastText embeddings were used to train models for both English and Hindi. The 300 dimensional pre-trained fastText embeddings for English and Hindi were used.

For ELMo embeddings, we fine-tuned the ELMo module provided by TensorFlow Hub. This module returns the ELMo embeddings for each word of the sentence, as well as the vector for the complete sentence. We used the 1024 dimensional vector of the sentence to train a neural network classifier.

4.3 Models

We used the Logistic Regression (LR), Support Vector Machine (SVM), Bidirectional Long Short-Term Memory (BiLSTM), an ELMo based Neural Network (NN) and an ensemble of the ELMo based NN and character-based LR classifiers. All the classifiers used are described below:

Logistic Regression: The LR classifier was used for both the English and Hindi dataset. L2 regularization was used for the classifier. The hyperparameter C was set to 1.2. The classifier was trained using the TF-IDF features of word n-grams (1,3), character n-grams (1,6), and combination of word n-grams (1,3) and character n-grams (1,6).

Support Vector Machine: The SVM classifier was used for both English and Hindi dataset. The 'linear' kernel was used for the classifier. L2 regularization was used and the hyperparameter C was set to 1.0. The classifier was trained using the same TF-IDF features as mentioned above for the LR classifier.

Bi-directional Long Short-Term Memory: The BiLSTM model used in this study is based on the architecture from Baruah et al. [2]. The architecture

of the model is shown in Fig. 1. It consisted of a BiLSTM layer and two Dense layers. The BiLSTM layer has 100 units and used a recurrent dropout of 0.10. A dropout of 0.25 was applied to the output of this layer. Global max pooling was applied on the output of the BiLSTM layer. The Dense layer that followed had 100 units and it used the ReLU activation function. A dropout of 0.25 was applied to the output of this layer also. The final Dense layer had 1 unit and the sigmoid activation function was used for this layer. The Adam optimizer and the binary cross-entropy loss function was used for training.

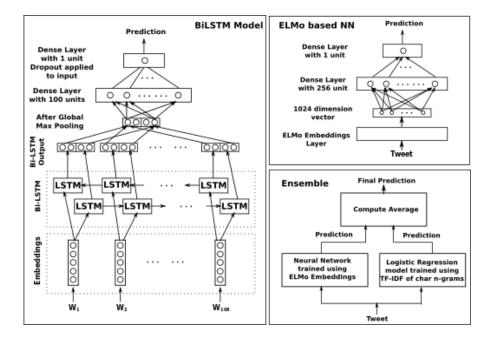


Fig. 1. BiLSTM model

The model has been trained using 200 dimensional Glove embeddings, 300 dimensional English fastText embeddings, and 300 dimensional Hindi fastText embeddings.

ELMo based Neural Network: The architecture of the ELMo based neural network is shown in Fig. 1. It consisted of an ELMo embedding layer and two Dense layers. The first Dense layer had 256 units and used the ReLU activation function. The next Dense layer had 1 unit and used the sigmoid activation function. The 1024 dimensional tweet vector obtained from the ELMo embedding layer is used to train the network.

Ensemble: The architecture of the Ensemble model used is shown in Fig. 1. It is the ensemble of the ELMo based NN classifier and the character n-gram based LR classifier. The prediction from the two classifiers were averaged to obtain the final prediction.

5 Results

As mentioned in section 3, training of the models for English was performed after combining the trial and train dataset. The models for Hindi were trained using the train dataset only. For validation, a stratified split of the dataset was performed. 20% of the dataset was reserved as the validation dataset and the remaining 80% was used for training the models. Table 2 and Table 3 presents the results obtained by our models on the English and Hindi validation dataset respectively.

Approach	Features	Acc	Prec	Rec	F1
LR	Char n-grams (1 to 6)	62.50	62.08	62.53	61.95
LR	Word n-grams (1 to 3)	63.60	62.63	62.94	62.69
LR	Char & Word n-grams	64.07	63.17	63.53	63.23
SVM	Char n-grams (1 to 6)	65.96	64.37	62.46	62.57
SVM	Word n-grams (1 to 3)	66.19	64.73	62.47	62.53
SVM	Char & Word n-grams	64.86	63.10	62.34	62.50
BiLSTM	pre-trained fastText	67.69	66.87	63.56	63.59
BiLSTM	pre-trained Glove	64.31	63.12	63.31	63.19
NN	fine-tuned ELMo	65.8	64.99	60.83	60.26
Ensemble	ELMo & Char n-grams	65.49	63.89	63.47	63.61

Table 2. Results of our models for English

As can be seen from Table 2 that for English, the BiLSTM model trained on pre-trained fastText embeddings performed the best on all the metrics considered. It obtained a macro F1 score of 63.59. The second best F1 score of 63.61 was obtained using ensemble of ELMo based NN and the character n-gram based LR model. By itself, the ELMo based NN classifier performed the worst among all the models with an F1 score of 60.26. However, it had the second-best precision score of 64.99. Among the LR models, the one trained using both character and word n-grams preformed the best with an F1 score of 63.23. The performance of all the SVM models were almost identical.

From Table 3, it can be seen that for Hindi, the SVM model trained on character n-grams performed the best on all the metrics considered. The model obtained an F1 score of 82.73. Word n-gram based models (both LR and SVM) did not perform well for the Hindi dataset. The BiLSTM model trained using fastText Hindi embeddings performed the worst with an F1 score of only 54.15. The reason for this poor performance could be that the dataset was a code-mixed dataset and it had English words also. Whereas the embeddings used was for Hindi only.

A. Baruah et al.

SVM

BiLSTM

Table 4 shows the confusion matrix for the LR and SVM models for English. As can be seen that, among the LR models, the word n-gram based LR models were better in predicting the non-hate category, while the character n-gram based model was better in predicting the hate category. Among the SVM models, both character and word n-gram based models performed equally well in predicting both the categories. Compared to LR models, the SVM models were better in predicting the non-hate category while the LR models were better in predicting the hate category.

For Hindi, as can be seen from Table 5, both character-based LR and SVM models performed equally well in predicting the non-hate category. The characterbased SVM models were slightly better in predicting the hate category. Both word-based LR and SVM models performed poorly in predicting the non-hate category.

From Table 6, it can be seen that the ELMo based NN model was the best in predicting the non-hate category among all the models. However, it was poor in predicting the hate category. For this reason, it was paired with the characterbased LR model in our ensemble model. The fastText based BiLSTM model was the second best in predicting the non-hate group. Compared to the ELMo based NN model, its performance in predicting the hate category was much better.

Based on these results obtained on the validation dataset, we selected the following models for submission: fastText based BiLSTM (English Run 1), our ensemble model (English Run 2), character and word n-gram based LR (English Run 3), character n-gram based SVM (Hindi Run 1), character n-gram based LR (Hindi Run 2), and character and word n-gram based SVM (Hindi Run 3).

The official results for our models are listed in Table 7 and Table 8. As we made an error in submitting the results for run 3 of the English language, the results for this run are missing. As can be seen from the tables for English, our best performing model on the test dataset was the fastText based BiLSTM model. It obtained a macro F1 score of 74.62. This model obtained the 8th position among 79 submissions for English. For Hindi, our best performing models were the character-based LR and SVM models with F1 score of 81.05 and 80.98 respectively. These two models obtained the official ranking of 4th and 5th position respectively among 37 submissions made for the Hindi. Table 9 shows the confusion matrix of our models for the official run.

Approach	Features	Acc	Prec	Rec	F1
LR	Char n-grams (1 to 6)	81.67	81.85	81.91	81.67
LR	Word n-grams (1 to 3)	77.49	77.57	77.65	77.48
LR	Char & Word n-grams	81.14	81.18	81.29	81.13
SVM	Char n-grams (1 to 6)	82.74	82.77	82.88	82.73
SVM	Word n-grams (1 to 3)	77.38	77.33	77.40	77.34

81.56

63.13

81.54

62.22

81.65

56.65

Char & Word n-grams

pre-trained fastText

81.54

Table 3. Results of our models for Hindi

Automated Hate Speech and Offensive Content Detection

Table 4. Confusion Matrix of LR and SVM models for the English Dataset

	L	R	LR		LR		SVM		SVM		SVM	
	Ch	ıar	Word		Char &		Char		Word		Char &	
	n-grams n-grams		ams	Word n-grams		n-grams		n-grams		Word n-grams		
	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF
NOT	474	286	504	256	504	256	611	149	620	140	572	188
HOF	191	321	207	305	201	311	284	228	290	222	259	253

Table 5. Confusion Matrix of LR and SVM models for the Hindi Dataset

	L	\mathbf{R}	L	LR		$_{ m LR}$		SVM		M	SVM		
	Ch	ıar	Wo	Word		Char &		Char		Word		Char &	
	n-grams n-grams			Word n-grams		n-grams		n-grams		Word n-grams			
	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	
NOT	377	62	353	86	368	71	374	65	341	98	365	74	
HOF	109	385	124	370	105	389	96	398	113	381	98	396	

 ${\bf Table~6.}~{\bf Confusion~Matrix~of~BiLSTM,~ELMo~based~NN~and~Ensemble~for~English,~and~BiLSTM~for~Hindi} \\$

	BiLSTM		BiLS	BiLSTM		ELMo based		emble	BiLSTM	
	English		English		Neural		ELMo	NN &	Hindi	
	Glove		fast	fastText		Network		ar LR	fastText	
	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF
NOT	520	240	644	116	656	104	561	199	683	77
HOF	214	298	295	217	331	181	240	272	392	120

Table 7. Official results for English Subtask-A

Run	Model	Accuracy	Precision	Recall	Macro	Weighted	Position
					F1	F1	
1	BiLSTM (fastText)	80.00	74.00	76.00	74.62	80.64	8th
2	Ensemble (ELMo	77.00	72.00	77.00	73.21	78.43	15th
	NN + Char LR)						
	Best System	-	-	-	78.82	83.95	1st

Table 8. Official results for Hindi Subtask-A

Run	Model	Accuracy	Precision	Recall	Macro	Weighted	Position
					F1	F1	
1	SVM (char)	81.00	81.00	81.00	80.98	81.06	5th
2	LR (char)	81.00	81.00	81.00	81.05	81.13	4th
3	SVM (word+char)	80.00	80.00	80.00	79.85	79.93	14th
	Best System	-	-	-	81.49	82.02	1st

Table 9. Confusion Matrix from the official results

			English Run 2		Hindi Run 1		Hindi Run 2		Hindi Run 3		
	fastText based		Ensemble of		Char n-gram		Char n-gram		Char and Word		
	BiLSTM		ELM	ELMo NN &		based		based		n-gram based	
			Char LR		SVM		$_{ m LR}$		SVM		
	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	HOF	NOT	
HOF	190	98	221	67	499	106	497	108	496	109	
NOT	129	736	195	670	144	569	141	572	156	557	

6 Conclusion

Hate speech and offensive content in social media is potentially dangerous for the society. As part of the shared task HASOC, this study used LR, SVM, BiLSTM and NN models for automated detection of hate speech and offensive content. Features such as word and character n-grams, Glove, fastText and ELMo embeddings were used in the study. Our best models obtained F1 score of 74.62 and 81.05 for English and Hindi dataset respectively. In our study, we did not use features such as dependency relations, part-of-speech tags etc. Further experiments can be performed to check if these features improve the performance of the classifier.

References

- Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep Learning for Hate Speech Detection in Tweets. In: WWW 2017. pp. 759–760. Perth (2017)
- Baruah, A., Barbhuiya, F., Dey, K.: Bi-directional LSTM for Hate Speech Detection. In: 13th International Workshop on Semantic Evaluation. pp. 317–376. Minneapolis (2019)
- 3. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N.: Hate Speech Detection with Comment Embeddings. In: WWW 2015. pp. 29–30. Florence, Italy (2015)
- 4. Gamback, B., Sikdar, U.: Using Convolutional Neural Networks to Classify Hate-Speech. In: ALW1 at ACL 2017. pp. 85–90. Vancouver (2017)
- 5. Kamble, S., Joshi, A.: Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. In: 15th International Conference on Natural Language Processing. pp. 155–160. Punjab, India (2018)
- Mathur, P., Shah, R., Sawhney, R., Mahata, D.: Detecting offensive tweets in Hindi-English code-switched language. In: Sixth International Workshop on Natural Language Processing for Social Media. pp. 18–26. Melbourne (2018)
- Modha, S., Mandl, T., Majumder, P., Patel, D.: Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (2019)
- 8. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive Language Detection in Online User Content. In: WWW 2016. pp. 145–153. Montreal (2016)
- Risch, J., Krestel, R.: Delete or not Delete? Semi-Automatic Comment Moderation for the Newsroom. In: TRAC-1 at COLING 2018. pp. 166–176. Santa Fe, USA (2018)
- Santosh, T., Aravind, K.: Hate Speech Detection in Hindi-English Code-Mixed Social Media Text. In: ACM India Joint International Conference on Data Science and Management of Data. pp. 310–313. Kolkata, India (2019)
- 11. Serra, J., Leontiadis, I., Spathis, D., Stringhini, G., Blackburn, J., Vakali, A.: Class-based Prediction Errors to Detect Hate Speech with Out-of-vocabulary Words. In: ALW1 at ACL 2017. pp. 36–40. Vancouver (2017)
- Waseem, Z.: Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In: NLP+CSS at EMNLP 2016. pp. 138–142. Austin, USA (2016)