IIITG-ADBU at SemEval-2020 Task 12: Comparison of BERT and BiLSTM in Detecting Offensive Language

Arup Baruah[1], Kaushik Amar Das[1], Ferdous Ahmed Barbhuiya[1], and Kuntal Dey[2]*
[1]IIIT Guwahati, India
[2]Accenture Technology Labs, Bangalore
arup.baruah@gmail.com, kaushikamardas@gmail.com,
ferdous@iiitg.ac.in, kuntal.dey@accenture.com

Abstract

Task 12 of SemEval 2020 consisted of 3 subtasks, namely offensive language identification (Subtask A), categorization of offense type (Subtask B), and offense target identification (Subtask C). This paper presents the results our classifiers obtained for the English language in the 3 subtasks. The classifiers used by us were BERT and BiLSTM. On the test set, our BERT classifier obtained a macro F1 score of 0.90707 for subtask A, and 0.65279 for subtask B. The BiLSTM classifier obtained a macro F1 score of 0.57565 for subtask C. The paper also performs an analysis of the errors made by our classifiers. We conjecture that the presence of a few misleading instances in the dataset is affecting the performance of the classifiers. Our analysis also discusses the need for temporal context and world knowledge to determine the offensiveness of a few comments.

1 Introduction

Detecting offensive language in social media is gaining a lot of importance. It is becoming commonplace to encounter offensive content in social media. This type of content if left unattended has the potential of creating a lot of damage to society. As such, research has been directed at developing automated systems for the detection and removal of offensive posts.

OffensEval 2020 is a shared task organized as part of the International Workshop on Semantic Evaluation 2020 (SemEval 2020) (Zampieri et al., 2020) to develop systems for detecting offensive content. OffensEval was also organized as part of SemEval 2019 (Zampieri et al., 2019). OffensEval 2020 had 3 subtasks: (1) Subtask A required participants to determine if a given tweet is offensive or not, (2) for offensive tweets, Subtask B required determining whether the offense is targeted or untargeted, and (3) if the tweet is a targeted offensive tweet, Subtask C required determining if it targets an individual, group, or other. This task was held for Arabic, Danish, English, Greek, and Turkish language.

We participated in all three subtasks for the English language. In this study, we used the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) and Bidirectional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) classifiers.

2 Related Work

The history of automatic offensive language detection goes back to Spertus (1997), where a rule based method was used to detect flames in comments received through the feedback forms of web pages. This topic has gained much research attention in recent times. Apart from offensive language, research has been performed to detect hate speech, abusive language, cyberbully, profanity, insults, and aggressiveness. Fortuna and Nunes (2018) defines these related terms.

Waseem and Hovy (2016) used a logistic regression classifier to detect racist and sexist tweets. Logistic regression and multi-layer perceptron classifiers were used in the study performed by Wulczyn et al. (2017). This study was performed to detect insults in Wikipedia comments.

*This work was done when the author was affiliated with IBM Research India, New Delhi

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1562–1568
Barcelona, Spain (Online), December 12, 2020.

Davidson et al. (2017) differentiated between hate speech and offensive language. It was found that differentiating between hate speech and offensive language is difficult. Almost 30% of the hate speech was misclassified as offensive language. Malmasi and Zampieri (2017) used an SVM classifier to perform this same 3-class classification and obtained an accuracy of 78%. Using ensembles and meta-classifiers, Malmasi and Zampieri (2018) was able to improve the accuracy to 79.8% on this same task.

3 Data

The Semi-Supervised Offensive Language Identification Dataset (SOLID) dataset was used in OffensEval 2020 for the English language task. This dataset is discussed in detail in Rosenthal et al. (2020). Table 1 to 3 below shows the distribution of different labels in the datasets for subtask A, B, and C respectively. The class for each instance in the dataset was predicted using several supervised models. The average confidence value predicted by these models along with the standard deviation were provided in the dataset. As can be seen from the table, the data set is imbalanced. For subtask A, only 15.94% of the instances were *offensive*; for subtask B, 33.32% of the instances were *untargeted*; and for subtask C, 80.73% of the instances were targeting *individual*.

OFF	NOT	Total
1448861 (15.94%)	7640279 (84.06%)	9089140

Table 1: Subtask A Dataset

TIN	UNT	Total
126004 (66.68%)	62970 (33.32%)	188974

Table 2: Subtask B Dataset

IND	GRP	OTH	Total
152562 (80.73%)	24917 (13.19%)	11494 (6.08%)	188973

Table 3: Subtask C Dataset

We studied the instances having high standard deviation and some of these instances seemed to be annotated incorrectly. For example, *"Y'all mean as shit"* (standard deviation 0.402) and *"Sco pa tu man-shut-the-fuck-up"* (standard deviation 0.465) are two such instances. Annotators labelled these instances as *not offensive* for subtask A. The average confidence values assigned to these instances were 0.442 and 0.476 respectively. In our opinion, these instances have an offensive meaning and should have been annotated as *offensive*. Thus, in our experiments, we removed the instances having a high standard deviation from the dataset for subtask A. Based on manual inspection a threshold of 0.38 was decided and instances having a standard deviation greater than 0.38 were removed from the dataset for subtask A. This filtering process was performed only for subtask A and not for subtask B and C. It led to the removal of 389 instances from the dataset for subtask A.

In our experiments, for subtask A and B, we considered the instances having average confidence values less than 0.40 as *not offensive* and *targeted* respectively. The rest of the instances were considered *offensive* and *untargeted*. The threshold of 0.40 was selected based on manual inspection of the instances. For subtask C, the class having the highest average confidence value was considered as the class for the instance.

For our experiments, we split the dataset provided into train and development set. The development set was created by performing a stratified split. For subtask A and C, 20% of the dataset was used as the development set. For subtask B, 30% of the dataset was used as the development set.

4 Methodology

4.1 BiLSTM

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (RNN). It can handle long range dependencies by using input gate, output gate, and forget gate to tackle the vanishing gradient problem. Many NLP tasks benefit from processing the text from both the direction. Schuster and Paliwal (1997) first used a bi-directional RNN to classify phonemes where the text was processed from both positive and negative time directions. BiLSTM is the bi-directional version of the LSTM classifier.

In our study, we used a single BiLSTM layer with 100 units. A recurrent dropout of 0.1 was used for the layer and a dropout of 0.25 was applied to the output of this layer. This output after max pooling was provided as input to a dense layer with 100 units. The *ReLU* activation function was used for this dense layer. A dropout of 0.25 was applied to the output of the dense layer and fed to another dense layer of 1 unit to make the final classification. The *sigmoid* activation function was used for this layer. The *Adam* optimizer with a learning rate of 0.001 was used for training the classifier. The loss function used was *binary crossentropy*.

The BiLSTM classifier was trained using fastText word embeddings. The 300-dimensional fastText embeddings [1] trained on Wikipedia 2017, UMBC webbase corpus, and statmt.org news dataset were used in our study.

4.2 BERT

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 word embeddings produced by fastText is static in nature. Each word has a single embedding irrespective of the context in which the word appears. Static embeddings fail to handle polysemy. The embeddings produced by BERT are contextualized embeddings. The same word may have multiple embeddings depending on the context in which it appears.

In our study, we used the uncased large version of BERT [2] to generate an embedding for each comment. This version has 24 layers and 16 attention heads. It generates a 1024-dimensional vector for each word. We used the 1024-dimensional vector of the Extract layer as the representation of the comment. Our classification layer consisted of a single Dense layer. For tasks A and B, this layer used the *sigmoid* activation function. For task C, it used the *softmax* activation function. The classifier was trained using the *Adam* optimizer with a learning rate of 2e-5. For tasks A and B, the *binary crossentropy* loss function was used. For task C, the *sparse categorical crossentropy* loss function was used.

Task	**System**	**Precision (Macro)**	**Recall (Macro)**	**F1 (macro)**	**F1 (weighted)**	**Accuracy**
Subtask A	BiLSTM	**0.9495**	0.9319	**0.9403**	**0.9583**	**0.9587**
Subtask A	BERT	0.9388	**0.9400**	0.9394	0.9572	0.9572
Subtask B	BiLSTM	0.8460	0.8260	0.8345	0.8549	0.8570
Subtask B	BERT	**0.8464**	**0.8554**	**0.8644**	**0.8811**	**0.8827**
Subtask C	BiLSTM	0.7978	**0.8391**	**0.8170**	**0.9262**	0.9242
Subtask C	BERT	**0.8207**	0.7886	0.8030	0.9241	**0.9258**

Table 4: Dev Set Results

5 Results

Table 4 shows the results obtained by our BiLSTM and BERT classifiers on the development set. As mentioned is Section 3, the development set was created by performing a stratified split on the dataset

[1] https://fasttext.cc/docs/en/english-vectors.html

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

	BiLSTM		BERT	
	Pred NOT	Pred OFF	Pred NOT	Pred OFF
NOT	1375962	26188	1362433	39717
OFF	48860	366741	38127	377474

Table 5: Confusion Matrix for Subtask A Dev Set

	BiLSTM		BERT	
	Pred TIN	Pred UNT	Pred TIN	Pred UNT
TIN	34739	3063	35436	2366
UNT	5045	13846	4282	14609

Table 6: Confusion Matrix for Subtask B Dev Set

	BiLSTM			BERT		
	Pred GRP	Pred IND	Pred OTH	Pred GRP	Pred IND	Pred OTH
True GRP	4358	381	244	4165	593	225
True IND	716	28981	816	572	29535	406
True OTH	230	476	1593	259	748	1292

Table 7: Confusion Matrix for Subtask C Dev Set

provided for the shared task. The official metric for evaluation was the macro F1 score. As can be seen from the table, the BiLSTM classifier performed better than the BERT classifier for subtask A and C. BERT performed better than BiLSTM in subtask B.

Tables 5 to 7 show the confusion matrices obtained by our classifiers for subtask A, B, and C respectively. In subtask A, BiLSTM performed better that BERT in predicting the majority class (NOT). BERT, however, performed better in predicting the minority class (OFF). In subtask B, BERT performed better than BiLSTM in predicting both the classes. In subtask C, BiLSTM performed better than BERT in predicting the minority classes (GRP and OTH). This was the reason for its superior performance in the subtask. BERT, however, performed better in predicting the majority class.

In this task, only the final predictions submitted were used to rank the systems. Our group submitted the BERT predictions for subtask A and B, and the BiLSTM predictions for subtask C. As the models took a long time for training, the models obtained during the validation stage was used to make the submission. This approach has a disadvantage that the final model submitted did not see a part of the dataset. Table 8 shows the performance of our systems in the test set. As can be seen from the table, our BERT classifier obtained a macro F1 score of 0.90707 and 0.65279 for subtask A and B respectively. Our BiLSTM classifier obtained a score of 0.57565 for subtask C. The scores of the best systems for the three subtasks were 0.92226, 0.74618, and 0.71450 respectively.

6 Error Analysis

This section discusses some of the errors made by our classifiers.

6.1 Misleading Instances

On analysis of the errors made by our classifiers on the development set, we found that many of the errors occurred when words such as *sick*, *boring*, etc. were present in the comment. Some of the comments that were misclassified are listed in Table 9. These comments were labelled as *not offensive* in the data set. Our classifiers misclassified then as *offensive*.

Task	Submitted System	F1 (Macro)	Score of Best System	Rank
Subtask A	BERT	0.90707	0.92226	50 out of 86
Subtask B	BERT	0.65279	0.74618	10 out of 44
Subtask C	BiLSTM	0.57565	0.71450	24 out of 40

Table 8: Official Results on Test Set

Sl.No.	Text	Average Confidence	Our Label	Predicted Label
9.1	Is test cricket...dare I say it......**boring**?!	0.386	NOT	OFF
9.2	Life is getting **boring** I think it's about time I spice it up a bit and do crack	0.339	NOT	OFF
9.3	i have the **sickest** dance moves on the team	0.363	NOT	OFF
9.4	I may be **sick** and in bed but theres a new video going up at 10pm BST	0.382	NOT	OFF

Table 9: Comments having the words sick, boring, etc. and annotated as not offensive

Sl.No.	Text	Average Confidence	Our Label	Predicted Label
10.1	whole family is mocking me for reading a book about salt as if there's anything **boring** about the history of a vital commodity!!	0.444	OFF	OFF
10.2	being single is **boring**. i hate it here	0.615	OFF	OFF
10.3	I hate getting **sick** cause I barely eat and when I have to it's a struggle	0.608	OFF	OFF
10.4	I thought I had allergies but I think I'm just **sick** the moment my throat started to hurt	0.591	OFF	OFF

Table 10: Comments having the words sick, boring, etc. and annotated as offensive

On closer analysis of other comments containing the above mentioned words (*sick*, *boring*, etc.), we found that many comments were labelled as *offensive* in the data set although the comments seemed to be *not offensive*. Such type of comments is listed in Table 10. For these comments, the labels determined by our classifier matched with the label provided in the data set.

We, thus, conjecture that the presence of such misleading instances in data as the cause of wrong predictions for the instances mentioned in Table 9. However, such conjecture needs further experimentation.

6.2 Presence of acronyms

Sl.No.	Text	Average Confidence	Our Label	Predicted Label
11.1	I don't give people the time of my day to be mistreated **gtfo**	0.414	OFF	NOT
11.2	**Smh** dondria went out as a one hit wonder	0.424	OFF	NOT

Table 11: Comments containing acronyms such as gtfo, smh

It was observed that our classifier failed to classify some instances that contained acronyms such as *gtfo*, *smh*, etc. Some of such types of comments are listed in Table 12. These *offensive* comments were misclassified by our classifier as *not offensive*. Whether the presence of these acronyms are causing the classifier to mispredict needs further investigation.

6.3 Need for temporal context and world knowledge

The following comments require temporal context and world knowledge to enable automated systems to decide if the comments are offensive or not. In the absence of such data, our classifiers misclassified the comments as *not offensive*.

Sl.No.	Text	Average Confidence	Our Label	Predicted Label
12.1	Kanye West probably thinks he is a genius because he spends most his time with the Kardashians.	0.442	OFF	NOT
12.2	@USER this is a compliment I swear: you remind me of a blonde nick cave	0.408	OFF	NOT

Table 12: Comments that need temporal context and world knowledge

6.4 Selection of threshold value for labeling purpose

The errors made by our classifier for the tweets 11.1, 11.2, 12.1, and 12.2, indicates that the choice of the average confidence value of 0.40 (at the time of data preparation) as the threshold for determining the *not offensive* and *offensive* classes may not have been the correct choice.

7 Conclusion

In our study, we found that both BERT and BiLSTM performed equally well in subtask A and C during the validation stage. In subtask B, BERT had a gain of 3% in the F1 score compared to BiLSTM. In the official results, our classifiers obtained an F1 score of 0.907, 0.653, and 0.576 in subtask A, B, and C respectively. Error analysis showed the presence of a few misleading instances in the dataset. We conjecture these instances to be affecting the performance of the classifiers. The classifiers also failed to classify tweets that required temporal context and world knowledge to interpret them. Incorporating world knowledge to help detect offensive content is a possible direction for future work.

References

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

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*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

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

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November.

E. Spertus. 1997. Smokey: automatic recognition of hostile messages. In *AAAI 1997*, pages 1058–1065, Rhode Island, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan. N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Z. Waseem and D. Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *NAACL-HLT 2016*, pages 88–93, California.

E. Wulczyn, N. Thain, and L. Dixon. 2017. Ex Machina: Personal Attacks Seen at Scale. In *WWW 2017*, pages 1391–1399, Perth.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.