TheNorth at SemEval-2020 Task 12: Hate Speech Detection using RoBERTa

Pedro Alonso **Rajkumar Saini** **György Kovács**
EISLAB Machine Learning
Luleå University of Technology
Luleå, Sweden
{pedro.alonso,rajkumar.saini,gyorgy.kovacs}@ltu.se

Abstract

Hate speech detection on social media platforms is crucial as it helps to avoid severe harm to marginalized people and groups. The application of Natural Language Processing (NLP) and Deep Learning has garnered encouraging results in the task of hate speech detection. The expression of hate, however, is varied and ever-evolving. Thus better detection systems need to adapt to this variance. Because of this, researchers keep on collecting data and regularly come up with hate speech detection competitions. In this paper, we discuss our entry to one such competition, namely the English version of sub-task A for the OffensEval competition. Our contribution can be perceived through our results, that was first an F1-score of 0.9087, and with further refinements described here climb up to 0.9166. It serves to give more support to our hypothesis that one of the variants of BERT, namely RoBERTa can successfully differentiate between offensive and non-offensive tweets, given the proper preprocessing steps.

1 Introduction

The present paper is a detailed description of the system[1] we use in the competition of detecting hate speech (Zampieri et al., 2020). The system proposed here was designed for the English language portion of sub-task A. We have decided to investigate a Deep Learning model, due to the considerable amount of data we had available at our disposal.

The macro F1-score of 0.9087 attained by our model in the competition puts us close to the top half of the competitors. While this can be discouraging, it should be noted that only one million samples were used to train our models due to the limitations of the available hardware capacity. Furthermore, even with the above-mentioned limitation, the difference between the classification score we attained, and the top team's classification score was only 0.02. This suggests that with some improvements (e.g. optimization of hyper-parameters and samples used), our system can be capable of State-of-the-art performance.

1.1 Related work

Delving into hate speech detection, we were able to gather previous research done on the topic, some by using Convolutional Neural Networks (CNNs) like (Zhang et al., 2018), (Gambäck and Sikdar, 2017) and (Pedro Alonso, 2019). As quite many research papers have dealt with hate speech with CNN, we decided to experiment with a different deep learning model that has shown excellent results in NLP tasks, namely BERT. To start, we found (Mozafari et al., 2019), which dealt with a task similar to ours, with good results. Therefore, we decided to try some of BERT's variants, namely DistilBERT (Sanh et al., 2019a) and RoBERTa (Liu et al., 2019a). After reading throughout the literature, we consider that the use of RoBERTa and DistilBERT, has not been researched enough, and we consider that our experiments serve to give some more visibility to the RoBERTa (Liu et al., 2019b) model.

[1] available at https://github.com/pedro-alonsod/OffenseEval

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2197–2202
Barcelona, Spain (Online), December 12, 2020.

2 Experimental data

Regarding the data used, for training, we used the training data published for the competition (Offenseval2020 (Zampieri et al., 2020) - described in Section 2.1), while for development and validation purposes, we used the OLID dataset (Zampieri et al., 2019) (described in Section 2.2). As we aimed to detect hate speech for English, we used the English language data from both datasets.

2.1 OffenseEval2020

The competition data available to us was OffensEval2020 (Zampieri et al., 2020), where task A (our focus) contained $9,089,140$ tweets. The task was annotated by two scores, namely mean and standard deviation. True labels were not provided for the training set of sub-task A, so part of the challenge was to select a threshold for the predictions attained and label them ourselves for the submission. The threshold found was then validated against the test set of 3887 tweets for which organisers provided the correct labels.

2.2 OLID

The OLID (Zampieri et al., 2019) data was also used for validation purposes, and setting the optimal threshold. The ways OLID (Zampieri et al., 2019), correspond to, if a post *”...do not contain offense or profane”*(Zampieri et al., 2019) language, that post is labeled as not-offensive if it violates any of rules in OLID such as, *”Posts containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct. These posts include insults, threats, and posts containing profane language or swear words”*(Zampieri et al., 2019), which are flagged as offensive. The OLID data comprises 13241 items or tweets, of which *4400 (roughly 35% percent)* fall in the offensive category. The rest are cataloged as non-offensive.

3 System overview

In this section, we give an overview of the full processing chain created to produce the results that had been submitted for the competition. First, we describe the text preprocessing steps we took to clean up the tweets and make them more usable to the whole system. Then, the focus will be on the selection of the model from the two models that were decided to test (DistilBERT (Sanh et al., 2019a) and RoBERTa (Liu et al., 2019a)), as well as, the reasons that lead us to select the final model (in this case RoBERTa (Liu et al., 2019a)). We will also discuss the hyper-parameters that gave us our (previously indicated) best score. Lastly, we discuss the algorithm we used once the output of our model was collected to transform it from a regression task to a classification one.

3.1 Text preprocessing

Text preprocessing was done on the tweets provided and trained the deep learning model with them. The tweets were tokenized, i.e., divided into words. Our preprocessing replaced or removed some of the tokens. With regular expression search, the $@ - words$ (e.g. $@StephenKing$) were replaced with $@USER$, and URLs (if any) were replaced with ”URL.” We noticed that the emoji count (number of emojis in a tweet) and the tweet scores were not correlated significantly. The facial emotion emojis also had almost

Table 1: Original tweets and their preprocessed counterpart

Original	**Preproccessed**
Somebody was abit excited by the first of his birthday celebrations 🎁🎂	Somebody was abit excited by the first of his birthday celebrations
@USER Lowkey my jam 😂🔥 but I’m on his head for this XXL shirt he wearing	@USER Lowkey my jam but I’m on his head for this XXL shirt he wearing
@USER His ass need to stay up 😂😂	@USER His ass need to stay up
@USER his bouffant tail is amazing 😄	@USER his bouffant tail is amazing

no correlation. Therefore, emojis were removed from the tweets. Hashtags (#) and emoticons were also removed from the tweets. In the end, the processed tweets were generated with only one whitespace between the words/tokens in each tweet (see Table 1 for examples of text preprocessing).

3.2 Models: RoBERTa & DistilBERT

The system we use to get our results is one of the many variants of BERT (Devlin et al., 2019), which is an instance of the transformers architectures of Deep Learning. The transformers are detailed in the original paper (Vaswani et al., 2017), and are a type of network that can deal with sequence-to-sequences transformation without the need for a recurrent neural network intervention. The main point raised in the transformers' paper (Vaswani et al., 2017) is that an attention-mechanism is the only thing that networks need to perform successful tasks while doing away with the idea that a recurrent neural network is also needed. One of the leading implementations of the transformers architecture and one of the most successful in NLP tasks is BERT (Devlin et al., 2019).

The success which BERT had on NLP tasks gave rise to several clones with slight modifications, namely DistilBERT (Sanh et al., 2019a), Roberta (Liu et al., 2019a), ElMo (Peters et al., 2018), to name a few. While these variants do offer some modifications, which for some tasks matter, they all share the common base, which is BERT and transformers at the end. For our experiments, we tried two transformer models, namely DistilBERT (Sanh et al., 2019a) and RoBERTa (Liu et al., 2019a). The results shown in table 2 represent our first experiments, which helped us to choose RoBERTa (Liu et al., 2019b) as our selected model. They were compiled using DistilBERT(Sanh et al., 2019b) for three and twenty epochs and (Liu et al., 2019b) for three, using early stopping in both cases. The threshold was selected as the one that got the best accuracy over the OLID (Zampieri et al., 2019) data sets.

Table 2: Model decisions analysis

Model	Training Epochs	Threshold	Results (macro-F_1 score) OLID train	OLID test
Distillbert	3	0.46	0.7917	0.6002
Distillbert	20	0.41	0.7720	0.5958
Roberta	3	0.42	**0.8043**	**0.6085**

4 Experimental setup

For training the model, the data was OffenseEval2020 data (Zampieri et al., 2020) raw was put to the preprocessing stage described previously. After we had the clean data, we took the first 1 million samples from the training data, and partitioned this one million tweets into a dev set of 3000 tweets (constructed from the last 3000 tweets), and the rest was used for training. The resulting training set was fed as input to the RoBERTa (Liu et al., 2019a) model. The hyper-parameters were mostly left as default, though we modified the following: "early_stopping_patience" this was set to three and "learning_rate" was set to $1e - 5$ and "evaluate_during_training_steps", the number of training steps before evaluating, was set to 2000. With these hyper-parameters, we trained two versions of DistilBert (one for 3 epochs, and another one for 20 epochs), as well as RoBERTa, for 3 epochs. Here, we used only the mean (and not the standard deviation) score as target for regression.

As the final step of the process, predicted scores had to be transformed into class labels. To do so, we set a threshold for each model on the OLID train data. This threshold was selected for each model based on the best result attainable on the training set of the OLID database. This threshold and the corresponding result for each model are listed in Table 2. Then for final submission, we selected the model that attained the best classification score on OLID test using the same threshold. As shown in 2, the RoBERTamodel attained the best classification score on the OLID train set, but its performance proved to be the best on the test set of OLID as well.

5 Results

When evaluated by the organizers, our submission attained a classification score of 0.9087 on the test set of OffensEval. To examine the results further, we have trained two more RoBERTa models, in this case, using 3000 instances for development, one for 1 epoch, and the other for 0.18 epoch (here, early stopping ended the training process when only 18% of training instances had been used). If we apply the same method on these models as before, the thresholds we arrive at are 0.45 and 0.44 Respectively. Using this threshold results in the scores shown in Table 4. As shown in Table 4, with an increased development set, we were able to attain the same results by either only training 1 epochs, or even using less training data. Furthermore, results show that by using a more sophisticated method for finding the optimal threshold, the resulting scores with the same models could be 0.9164 and 0.9166 respectively.

Table 3: Model hyper-parameters.

Model Name	Epochs	Learning Rate	Threshold
RoBERTa	0.18	1e-5	0.45
RoBERTa	1	1e-5	0.44

Table 4: Model latest results

Model	Threshold	Results (macro-F_1 score)		
		OLID train	OLID test	OffensEval test
RoBERTa	0.45	0.7960	0.8012	**0.9116**
RoBERTa	0.44	**0.8124**	**0.8026**	0.9085

6 Quantitative analysis

Here we describe what hyper-parameter changes (we only varied two) or modeling (also includes the two) decisions we think affected our results. As presented in Table 2, the current results obtained by a change in hyper-parameters are better from the ones reported. To get these results, we used a more conservative train/dev split, in this case, 1.04 million training samples, 3000 *dev*, and a combination of 0.18 and 1 epochs in training.

6.1 Error analysis

Here are some of the misclassified tweets (as OFF) and our intuition as to why these errors were made.

A33 - I see the news but I refuse to believe it and will continue my day wilfully ignorant. Glancing at this tweet, our impression is that ”wilfully ignorant” (or just ”ignorant”) may be offensive in most context, but in this case (since the person is saying it about himself/herself), it is not. However, the model may not be sensitive to such nuances.

A43 - It really sucks I want to stream so bad but after an hour of talking my throat says that’s it for today. Here we believe the model takes the word **sucks** as being offensive in this context: this is the type of error that is worth being pushed down to content moderators.

A253 - @USER He deserve the worst of this all. This last example, which was also misclassified, we think our system is being harsh in its evaluation. As the tweet does carry anger towards a person and may sometimes be censored, but in our view, it is not offensive.

We show confusion matrices obtained (Table 5), where it shows that our system is punishing the NOT tweet harder where there may be no need to do so. One can also see from Table 5 that when setting the threshold to the value used in our earlier model, we can attain a high f-score without any false negatives while having a relatively small amount of false positives.

Table 5: Confusion matrices with two different thresholds on the Offenseva2020 Test set

		Predicted	
	th=0.420	NOT	OFF
Actual	NOT	2500	307
	OFF	0	1080

		Predicted	
	th=0.572	NOT	OFF
Actual	NOT	2554	253
	OFF	23	1057

7 Conclusion

To conclude this paper, our system shows a good performance while not being overly complicated; we show that it is possible to tackle the task of hate speech detection using one of the variants of BERT (Devlin et al., 2018), in this case, RoBERTa (Liu et al., 2019a). We are obtaining an F1-Score of 0.9087 (final result), which can be further pushed down to content moderators. For future work, we will continue focusing on the model proposed here with slight modifications to hyper-parameters and text preprocessing. We think that getting a 0.92 is quite possible with further optimization of hyper-parameters, and by increasing the size of the development set. We also plan to experience with ensembles of models trained on different portions of the train set.

Acknowledgements

Part of this work has been funded by the Vinnova project ”Language models for Swedish authorities” (ref. number: 2019-02996)

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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, June. Association for Computational Linguistics.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media.

György Kovács Pedro Alonso, Rajkumar Saini. 2019. TheNorth at HASOC 2019 Hate Speech Detection in Social Media Data. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, December.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019a. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019b. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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 1415–1420, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 745–760, Cham. Springer International Publishing.