Vrije Universiteit
Subjectivity Mining

Final course paper

Which BERT can we trust?

Kamiel Gülpen (2718984)
Dante de Lang (2585032)

October 29, 2021

1 Introduction

With increasing digitization and online communication, offensive and hateful content has become a pressing problem within online communities and governmental organizations. In order to be able to address these problems regulatory measures are needed which are capable of handling such large amounts of content. Hate speech recognition has shown to be one of the more successful computational methods capable of monitoring online offensive content [10].

This paper aims to function as an additional guideline for model comparison. While there are already existing papers introducing both qualitative and quantitative comparison methods, see section 2, we add a similar approach based on the interpretability of models. For this comparison we implement three NLP (Natural Language Processing) classifiers and apply them on the same dataset. The results will then be evaluated in a quantitative and qualitative way. For the NLP classifiers we choose three different versions of BERT, which stands for Biderectional Encoder Representations from Transformers. The three implemented models are BERT-base-uncased, HateBERT and RoBERTa, these will be discussed more thorougly in section 4.1.

All three models will be fine-tuned and tested on the HateExplain dataset [6]. In order to asses the performance of the different BERT models we use the LIME (Local Interpretable Model-agnostic Explanations) method. The combination of both the information rich dataset and the LIME interpretability method enables us to do a more robust qualitative comparison between the BERT models. More information on the dataset and the LIME model will be provided in section 3 and 4.2.

2 Related work

In the past years, research on hate speech and automatic hate speech detection has increased. According to data gathered by Clarivate Analytics the amount of publications and citations with hate speech as a topic have increased continuously since 2017. An overview of this is given in Figure 2 in the appendix, see section A.

This paper aims to provide a method for comparing sentiment analyses models. For this method we rely on values extracted through LIME, an existing method useful for providing explanations of predictions of any classifier or regressor [7]. In the paper of Ribeiro et al. [7] they provide various examples for which LIME is able to explain the "reasoning" of a predicting model. Regarding the implementation of LIME on text classification models, the paper of Mathew et al. [6] functions as an interesting example. In their paper they use LIME in order to compare various

models using explainability measures based on the output of LIME. The information rich dataset provided by [6] will be used for this research in order to compare the LIME output of models with annotator labels.

The three models used in this research are BERT [3], RoBERTa [4] and HateBERT [1]. All three models are specifically build as text classifiers with HateBERT being finetuned on hate speech recognition. In order to compare these models using the output of LIME, we will use a Rank Biased Overlap (RBO) method [9]. With this method we are able to compare the ranked lists with hate labelled words.

3 Data

For this paper the Hatexplain dataset is used, which is first introduced by Mathew et al.,[6]. The dataset of Mathew et al., consists of data from two sources, namely Twitter[1] and Gab[2]. The Twitter data is obtained by collecting 1% tweets randomly in the time period Jan-2019 to Jun-2020. The Gab dataset is obtained from the paper of Mathew et al.,[5]. The final dataset consists of 9,055 twitter messages and 11,093 Gab messages resulting in a dataset of total 20,148 messages.

The text data is classified based on the classification scheme of Davidson et al.,[2]. Based on this scheme, the dataset has the following labels: normal, offensive, hate speech and undecided. A total of 253 Amazon Mechanical Turk (MTurk) workers were used in order to annotate the dataset. They first were asked to classify the text as normal, offensive or hate speech and when classified as offensive or hate speech they were also asked to label the targeted community. In addition, if an annotator believes the text to be hate speech or offensive, the annotator was asked to annotate sections of the text, either terms or phrases, that might be a plausible cause for the provided annotation, these annotated sections are called rationales.

While the original HateExplain data contains all information on annotations within a JSON format, it was needed to convert this first to VUA format. When formatting it was also needed to draw conclusions between the annotations. This was done through choosing the most occurring label of three. For the case that there were three different labels, the sentence was labelled as "undecided".

Because of the limited computational resources we used only 20% of the total dataset described above for our experiments. In this dataset a total of 782 sentences were annotated as normal, 583 as offensive, 558 as hate speech and 77 as undecided. Because we are only interested in hate speech we divide the dataset into hate speech

[1]https://twitter.com/

[2]https://gab.com/

and no hate speech. Our resulting dataset consists of 558 instances of hate speech and 1442 instances of no hate speech with the corresponding rationales.

4 Methods

4.1 Classifiers

For this study the three models, used to identify hate speech, are based on the original BERT model [3]. Without going into too much detail, it is necessary to get some basic understanding of the operations of this model. BERT uses a transformer which is a mechanism that can learn the relationship between words in a text. The BERT model differs from other transformer models in the sense that it does not read text sequentially but rather reads the text all at once. The BERT model is therefore a bidirectional transformer model, which has the advantage that the model can learn the context of the word based on the whole text. The BERT model is furthermore designed to pre-train deep bidirectional representations from unlabeled text and can then be fine tuned to a specific data-set to create state of the art results. In this study we choose three different pre-trained models to compare with each other, namely BERT-base-uncased (BERT), HateBERT and RoBERTa.

The BERT-base-uncased model, the basic model first introduced by Devin et al.,[3], is trained on a very large amount of data, namely on the whole domain of the English Wikipedia and on the Brown Corpus, which contain 2.5 billion and 800 million words respectively. The fact that it is pre-trained means that the model already 'understands' English text, and only has to be fine-tuned to perform a specific task. This makes for a better performing model, saves time and requires less training data. This model was chosen because of the large amount of data it was pretrained on, and to provide a BERT baseline.

The second model implemented in this paper is HateBERT. This model was first presented by Caselli et al. [1] and is a re-trained version of the BERT-base-uncased model. This re-training was done using the RAL-E dataset, which consists of 1.5 million Reddit messages coming from banned communities due to hosting or promoting offensive, abusive and/or harmful content. The extensive re-training resulted in a shifted version of BERT in terms of language variety and polarity. The authors achieved better results with the retrained model than with BERT-base-uncased on automatic detection of hate speech, making it a natural candidate model for the present study. Furthermore, comparing HateBERT and RoBERTa together against the baseline BERT model allows us to determine whether the improved performance of HateBERT is due to the retraining on just language variety or also

the polarity, meaning the presence of offensive and hateful language.

The third and last model implemented for this paper is the Robustly optimized BERT approach (RoBERTa), first presented by [4]. This model is in essence an improved recipe for training BERT models. This recipe contains larger batches and longer training sessions with longer sequences. The authors of [4] also removed the next sentence prediction objective present in the original BERT model, while also dynamically changing the masking pattern for the training data. Furthermore, besides increasing the training time, the amount of training data was also increased by almost 10 times with respect to BERT. This was done by adding three datasets besides the original BERT corpora, namely the CC-News dataset, containing 63 million news articles, the OpenWebText dataset, containing URL's shared on Reddit with at least 3 upvotes and the Stories dataset, containing a CommonCrawl dataset focused on story-like styled text.

4.1.1 Technical settings

The models where implemented using the PyTorch library[3] and a open source data and model platform called Huggingface[4]. All classifiers were fine-tuned on the datasets for only 2 epochs due to time and computational restrictions. The models were run using GPU's available through Google Colab's cloud computing environment. The specifications of this GPU are not always known since it is not possible to choose which GPU a user can connect to, however they often include Nvidia K80s, T4s, P4s and P100s. In all cases it was not possible to handle batch sizes larger than 8 since otherwise a GPU memory error will occur, ideally this batch size would have been larger, for example 16 or 32.

4.2 LIME

Ribeiro, Singh and Guestrin first introduced LIME in their paper "Why should I trust you?" [7]. LIME stands for Local Interpretable Model-agnostic Explanations and is an unique explanation technique that is capable of explaining any classifier's predictions in an understandable and faithful manner. This is done by developing an interpretable model locally around the predictions made by the classifiers [7]. The LIME model can be seen as a agnostic model as it is applicable on any machine learning model. For a text classifier, the LIME model takes as input the concerning string and a the model that needs to be tested. This model should be able to take

[3]https://pytorch.org/

[4]https://huggingface.co/

a list of strings as input and a probability as output. The LIME model works by putting various strings into the model and saving the corresponding probabilities. It will then alternate this string a certain amount of times, based on the sample size, by removing different words and saving the probabilities. When finished, LIME will return the most interesting words of a sentence based on those probabilities. In other words LIME modifies a single data sample by tweaking the feature values while observing the impact on the output.

4.2.1 Technical settings

In order to be able to analyze a large part of the text, we chose to look at the 20 most important features. We furthermore choose a sample size of 100 as this is a computationally heavy computation and we did not find any great differences in ranking when increasing the sample size.

4.3 Evaluation-metrics

4.3.1 Classifiers

In order to assess the performance of the various transformer models, evaluation metrics are needed. To get a visual overview of performance, a confusion matrix can be helpful. Within this matrix the true labels are set up against the predicted labels. For a binary classification task this matrix will therefore show 4 values which directly gives insight in what type of errors are made and therefore also how well a classifier performs per class. Especially with skewed data a confusion matrix can be very helpful. An example for this would be an email spam-filter, when such a filter has a 1:100 chance of meeting a spam mail, it could still get a 99% accuracy when labelling all emails as safe. However, this doesn't mean the filter performs well. This can also be said about hate-speech recognition. Mostly since some datasets are skewed, it is necessary to get a clear image of how well and for which classes a classifier performs correctly.

An alteration of this matrix used in this paper is normalizing the counted values over the true labels. Therefore, giving better insight in its performance per class. The values from the confusion matrix can also be used to calculate additional performance metrics. Most common metrics are Precision, Recall and the F_1-score. As will become clear in the next sections, our evaluation metric of choice is the macro F_1 score, which is a mean of label-wise F_1 scores.

4.3.2 Ranking Biased Overlap (RBO)

In order to be able to compare the LIME values with HateExplain, we introduce an existing method in order to analyze explainability. This method is based on the Rank-Biased Overlap (RBO)[5]. This RBO method is able to compare lists of different sizes containing different values. The output range of this RBO algorithm is a value between 0 and 1, a 1 represents full similarity between rankings and a 0 means they are totally different. Since we only have data from annotators for hateful/offensive texts, these text were also only selected from the predicted data using their ids. As input for this RBO algorithm we selected from both datasets the same amount of words which depended on the amount of annotations present in the HateExplain dataset.

5 Results

In this section the results of the experiments will be represented. First the predictive accuracy of the three models described in section 4.1 will be discussed and interpreted. Secondly, a error analysis will be conducted on the predictions of the classifiers and finally, the interpretability and explainability of the models will be treated.

5.1 Predictive accuracy

Table 1 shows a summary of the average macro results of the BERT, HateBERT and RoBERTa models on the Hatexplain dataset. The table shows that RoBERTa has the highest F_1 score of 0.87 followed by HateBERT with a score of 0.85 and BERT with a score of 0.84. This means that the model with the most pre-trained data points has the most accurate predictions on the Hatexplain dataset.

Model	Recall	Precision	F_1
BERT	0.84	0.84	0.84
RoBERTa	0.87	0.87	**0.87**
HateBERT	084	0.85	0.85

Table 1: Macro average scores of BERT models on Hatexplain dataset.

[5]https://github.com/changyaochen/rbo

5.2 Interpretability and explainability

For all three models the similarities per tweet were calculated using the RBO algorithm. When averaging over all tweets per model, we get an average indication of the similarity between the annotations stated in the HateExplain paper [6] and the interpreted "annotations" extracted by LIME. Looking at the RBO averages the BERT model turned out to receive the lowest with a value of 0.678, while the RoBERTa model was able to get a similarity value of 0.697, lastly the HateBERT model received the highest similarity score of 0.704.

	RoBERTa			HateBERT			BERT		
Rank	Word	Count	Avg. Sent.	Word	Count	Avg. Sent.	Word	Count	Avg. Sent.
0	muzzrat	1	0.779101	muzzrat	1	0.774937	muzzrat	1	0.798452
1	kike	30	0.689932	nigger	41	0.667333	raghead	1	0.789936
2	muzzies	4	0.666688	muzzies	4	0.652040	mussie	1	0.783010
3	nigger	41	0.662049	sandniggers	1	0.645593	kike	30	0.692875
4	muzzie	3	0.584462	raghead	1	0.605838	muzzies	4	0.668411
5	niggers	17	0.502742	kike	30	0.601421	nigger	41	0.654846
6	muzzy	1	0.465004	muzzie	3	0.599220	muzrat	3	0.569488
7	sandniggers	1	0.454777	muzzy	1	0.580785	mudshark	3	0.556818
8	moslem	9	0.414581	niggers	17	0.487563	muzzie	3	0.550262
9	muzrat	3	0.381952	mussie	1	0.445098	niggers	17	0.503434

Table 2: This table shows the top 10 of hate classified words per BERT model. For every word the average sentiment value and count is presented.

	RoBERTa			HateBERT			BERT		
Rank	Word	Count	Avg. Sent.	Word	Count	Avg. Sent.	Word	Count	Avg. Sent.
0	joaquin	1	-0.205752	pink	1	-0.267446	gender	1	-0.191443
1	amber	1	-0.148634	waist	1	-0.256427	goal	1	-0.178050
2	waist	1	-0.140107	terrorist	1	-0.195071	anonymous	1	-0.173180
3	muhammad	1	-0.124602	debt	1	-0.169730	vegas	1	-0.168834
4	disingenuous	1	-0.116302	amber	1	-0.144746	dieting	1	-0.161721

Table 3: This table shows the top 5 of no-hate classified words per BERT model. For every word the average sentiment value and count is presented.

When taking a closer look at the LIME values per word, we are able to create a list showing the average sentiment value per word with the amount of occurences. In Table 2 the top 10 words classified as hate with the highest sentiment values are shown, in Table 3 the top 5 words are shown for no-hate classified words. As shown in Table 2 we see that the words in the top-10's are very similar between the different models. Foremost, it is interesting to see that almost all words can be categorized as racist or religious swear words. Such a theme or similarity between lists is not the case when looking at the no-hate labelled top 5 in Table 3.

5.3 Error analysis

An error analysis is conducted to get a deeper understanding of the performance of the three models on the hatexplain dataset. The confusion matrices of the models will first be described followed by a analysis on Type I and Type II errors.

Figure 1 shows the error matrices of the BERT models on the Hatexplain dataset. One can observe that all three models have a relatively high percentage of correct hate predictions as well as noHate predictions, namely are 79% - 84% and 90% - 91% respectively. It is interesting to see that the models perform almost equally good when predicting noHate messages but perform relatively different when predicting hate speech messages. Furthermore it can be observed that all models do better on predicting noHate messages than Hate messages, which is in line with the findings of Davidson et al.,[2].

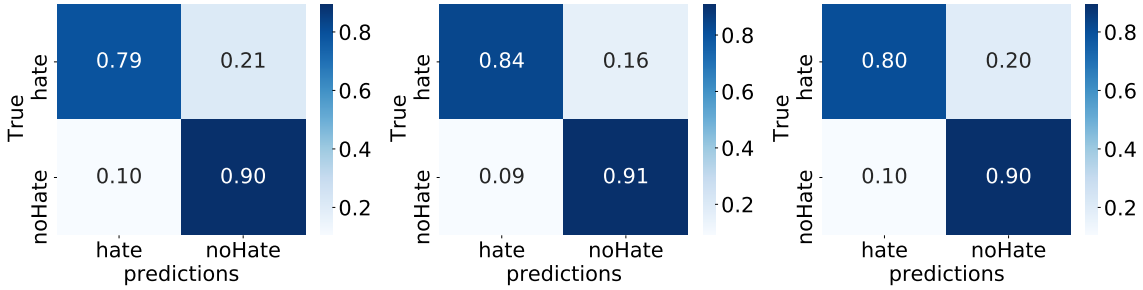

Figure 1: Error matrices of the predictions of BERT (left), RoBERTa (middle) and HateBERT (right)

Type I error: A Type I error is an error where a text is annotated as hate speech but is not predicted as such. The error I values for all three models are relatively high, ranging from 16% to 21% of the messages labeled as hate speech. The highest error I value is obtained by the BERT model (21%). After looking through all the errors we see a interesting trend, namely that a majority of all the type I errors do NOT contain racial, cultural or religious words namely 18/26. Another possible reason for the misclassificaton can be the absence of swearwords as 13/26 of the messages did not contain swearwords.

We see a similar trend in the other models, the type I errors obtained by the HateBERT model (20%) are mostly caused by non-racial, non-cultural or non-religious messages namely 17/26, but only 4/26 contained no swear words which differs from the BERT model. The RoBERTa model (16%) also showed non-racial, non-cultural or non-religious messages in the majority of the errors, namely 13/20 while only 7/20 did not contain curse words.

Through this error I analysis we hypothesize that the majority of the predictions resulting in a type I error are caused by the absence of hate towards races, cultures or religions. Table 2 shows that this hypothesis is feasible as the most important features for hate classification are racial, cultural or religious. It is furthermore interesting to see that the majority of the obtained error I messages by the models overlapped, namely 18, which is 90% overlap for the RoBERTa model and around 70% for the HateBERT and BERT model. A example of a overlapping message without hate towards race, culture or religion is: "that bullshit you know nothing you are a fucking retard".

Type II error: A type II error is an error where the text is annotated as no hate speech but is labeled as if it is. Figure 1 shows that BERT and HateBERT have the same percentage of predictions resulting in a type II error, namely 10% while RoBERTa obtained a similar type II error score of 9%. When looking at the errors obtained by the BERT model we see the same trend as observed with the type I error, namely the majority of the sentences contained racial, cultural or religious words (22/29) or swearwords (18/29). An example of a misclassified sentence is: "this whole album is ¡number¿ ¡number¿ very underrated not joking he even says nigger hard r in one of the songs hee hee".

We see a similar pattern when looking at the type II error obtained by HateBERT and RoBERTa, which show that for HateBERT and RoBERTa respectively 23/29 and 21/25 words contained racial, cultural or religious words and 14/29 and 11/25 contained swearwords. It is furthermore interesting to see that the sentences overlap of the error II is much less than that of the error I, namely: 16, which is 64% for RoBERTa, and 55% for BERT and HateBERT.

6 Discussion

Three BERT models, BERT, HateBERT and RoBERTa trained and tested on the Hatexplain dataset. Their performance is furthermore analysed for interpretability and explainability with the LIME model. The RoBERTa model outperformed the HateBERT model and BERT model on predictive accuracy. The HateBERT performed second best on this metric and the standard BERT model performed the worst, even though the F_1 scores on the different models all lie within a few percentage points of each other. It was expected that HateBERT would outperform the other two models in predictive accuracy as it was retrained on the type of language thought to be most similar to the language used in the dataset. One possible explaination for RoBERTa to outperform HateBERT is that it was pretrained on the largest amount of data of the three models. This suggests that the models perfor-

mance can be better explained by the quantity of pre-trained data than the quality of the data.

The HateBERT model outperforms the others based on interpretability and explainability. RoBERTa performed second best while BERT obtained the lowest score, though all obtained scores lay close to each other. This similarity in results can be explained by the top 10 important features for prediction of the models. We see that all three models consider the similar words as good predictors of hate speech. A large proportion of the features occurring in the top ten of one model also seem to appear in the top ten of the other models. The biggest difference, however, is the ranking of these features, which can explain the difference in interpretability/explainability score. It is furthermore interesting to note that for all three models there is a high probability that they will classify a sentence as hate speech when a cultural, religious or racial word appears in it. This is in line with the findings of Mathew et al.,[6] and Davidson et al.,[2].

For further research it would be interesting to look at the paper of Sarkar et al. [8]. In their paper they introduced a new BERT model, fBERT, that is, similarly to HateBERT, retrained on a hate speech corpus. They compared its performance against BERT and HateBERT, using the Zampieri and Davidson datasets. fBERT seems to consistently outperformed BERT and HateBERT greatly in their experiments and could therefore potentially outperform HateBERT and even RoBERTa on the Hatexplain dataset as well.

7 Conclusion

With this paper we contributed to existing literature of comparative researches regarding predictive accuracy and interpretability of the BERT model. We looked at three different BERT models and analyzed them based on there predictive accuracy and interpretability while also conducting an error analysis. We observed that RoBERTa obtained the highest predictive accuracy score while HateBERT performed the best regarding interpretability. The errors made by the models where similar as they seemed to be biased towards cultural, religious and racial words. Future research should also include the fBERT model [8] in its experimental setup as it has been found to be even more effective at detecting hate speech than BERT and HateBERT and would therefore be a interesting candidate for an analysis on interpretability.

References

[1] T. Caselli, V. Basile, J. Mitrovic, and M. Granitzer. Hatebert: Retraining bert for abusive language detection in english. *ArXiv*, abs/2010.12472, 2021.

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, June 2019.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[5] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee. Spread of hate speech in online social media. In *Proceedings of the 10th ACM conference on web science*, pages 173–182, 2019.

[6] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*, 2020.

[7] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[8] D. Sarkar, M. Zampieri, T. Ranasinghe, and A. Ororbia. Fbert: A neural transformer for identifying offensive content. *arXiv preprint arXiv:2109.05074*, 2021.

[9] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

[10] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*, 2019.

A Appendix

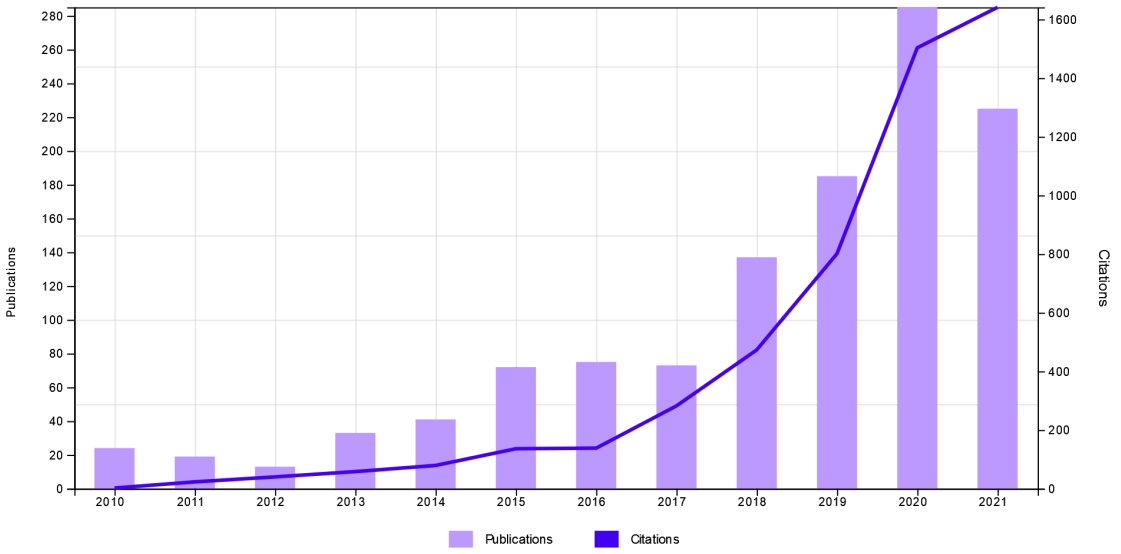

Figure 2: Data on citations and publications from 2010 till 2021 for scientific papers with hate speech as a topic. Graph was requested as part of a citation report on 26 October 2021 from the web of science interface provided by Clarivate Analytics.