Which one is more toxic? Findings from Jigsaw Rate Severity of Toxic Comments

Millon Madhur Das, Punyajoy Saha, Mithun Das
Indian Institute of Technology, Kharagpur, India
millonmadhurdas@kgpian.iitkgp.ac.in, {punyajoys, mithundas}@iitkgp.ac.in

Abstract

The proliferation of online hate speech has necessitated the creation of algorithms which can detect toxicity. Most of the past research focuses on this detection as a classification task, but assigning an absolute toxicity label is often tricky. Hence, few of the past works transform the same task into a regression. This paper shows the comparative evaluation of different transformers and traditional machine learning models on a recently released toxicity severity measurement dataset by Jigsaw. We further demonstrate the issues with the model predictions using explainability analysis.

Note: *This paper contains examples of toxic posts. But owing to the nature of work, we cannot avoid them.*

1 Introduction

In social media, toxic language denotes a text containing inappropriate language in a post or a comment. The presence of toxic language on social media hampers the fabric of communication in the social media posts; e.g., toxic posts targeting some community might silence members of the community (Das et al., 2020). Subsequently, social media platforms like Facebook (Facebook, 2022) and Twitter (Twitter, 2022) have laid down moderation guidelines. They also employ various automatic and manual detection techniques to detect such forms of language and apply appropriate moderation (Schroepfer, 2021). Henceforth, researchers have started looking into this direction (Das et al., 2021b; Banerjee et al., 2021; Das et al., 2021a). Most of the past research focused on developing a classification task which again varies based on the classification labels the researchers choose, i.e., abusive/non-abusive, hate speech/offensive/normal, troll/non-troll etc. (Nobata et al., 2016; Mathew et al., 2021; Saha et al., 2021; Das et al., 2022a,b) This variation in the classification labels makes transferring models across different datasets tricky. Secondly, assigning a label to a post in terms of toxicity labels is complicated as many of the posts can be subjective (Aroyo et al., 2019). Finally, a further challenge is that after encountering several highly toxic comments, an annotator might find subsequent moderately toxic comments as not toxic (Kurrek et al., 2020).

Research is currently trying to situate the toxicity detection tasks as regression tasks. In its simplest form, an annotator is provided two samples, and they have to decide which one is more toxic. Eventually, these annotated comparisons are converted to a scalar value which denotes the level of the toxicity of the post. Hada et al. (2021) uses best-worst scaling (Kiritchenko and Mohammad, 2017) to assign toxicity scores to a post based on the comparison annotated by annotators. Besides, another study (Kennedy et al., 2020) used Rasch measurement theory for converting the comparisons to scalar values.

In this shared task, Jigsaw released a new dataset for understanding the severity of toxic language. The organizers select a set of 14,000 datapoints. They used these datapoints to create multiple pairs, which were then annotated by some annotator. The annotators marked one of the comments as toxic based on their notion of toxicity. These comparisons were compared with the ones received from models, and average agreement was used as the final score.

Jigsaw is a unit within Google that explores threats to open societies, and builds technology that inspires scalable solutions. They forecast emerging threats like Disinformation, Censorship, Toxicity and Violent Extremism and explore how technology can protect individuals and societies.

In this paper, we focus on developing models for this task. Since the shared task did not provide any training dataset, we utilized different classification-based toxic language datasets and converted their labels to a scalar value based on various strategies.

Proceedings of the 29th International Conference on Computational Linguistic, pages 10–15
October 12–17, 2022.

Finally, we use simple models like TF-IDF to complex models like Transformers. We conclude the paper with a detailed error analysis to understand the behavior of the models.

2 Datasets

In this section, we illustrate the datasets used for this task. The first section 2.1 describes the task dataset, and the second section 2.2 exhibits the dataset used for training the models since we don't have any training dataset associated with this task.

2.1 Task dataset

In the task dataset [1], pairs of comments were presented to expert raters, who marked one of two comments more harmful – each according to their notion of toxicity. The final label for each pair is decided with a majority vote. The validation dataset contains $\sim$ 30k data points where each datapoint was a pair of toxic posts with the annotation mentioning which one is more toxic. However, this data cannot be used to train the models as they do not contain a toxicity score value for each comment. Apart from this we were provided with 5% of the test dataset for validating our models. The rest, 95%, is private and was used as hidden test data. Our results are discussed for the validation dataset and entire test dataset (150k posts).

2.2 External datasets

2.2.1 Ruddit

This dataset (Hada et al., 2021) contains English language Reddit comments that have fine-grained, real-valued scores between -1 (maximally supportive) and 1 (maximally offensive). The annotators were given a set of 4 comments and asked to arrange them in order of their toxicity/abusiveness. These were converted to scalar scores using best-worst scaling (Kiritchenko and Mohammad, 2017). We transformed these scores to a value between 0 and 1 to keep the distribution of values uniform to other datasets. This dataset contains $\sim$ 16k data points.

2.2.2 Jigsaw Toxic Comment Dataset(JTC)

This dataset contains a large number of Wikipedia comments labeled by human raters for toxic behavior. The types of toxicity are toxic, severe toxic, obscene, threat, insult, and identity hate. Each comment can have any one or more of these labels. It contains $\sim$230k data points. This dataset is a part of the Toxic Comment Classification Challenge hosted on Kaggle [2]. We converted the labels into a single score. The different toxicity categories were given different weights, and the final toxicity score was the sum of weights for each example. Our final weighing scheme was, severe toxic:12, identity hate:9, threat:8, insult:6, obscene:5, toxic:4

2.2.3 Jigsaw Unintended Bias Dataset

This dataset is part of a Kaggle Competition, Jigsaw Unintended Bias in Toxicity Classification [3]. Each comment has a toxicity label that lies between 0 and 1. It has $\sim$ 2 million samples. This attribute (and all others) are fractional values representing the fraction of human raters who believed the attribute applied to the given comment. For evaluation, test set examples with a target >= 0.5 will be considered to be in a positive class (toxic).

The data also has several toxicity sub-type attributes like severe toxicity, obscene, threat, insult, identity attack, and sexually explicit. We have used mapping similar to that used for the Jigsaw Toxic Comment dataset for assigning the toxicity score.

2.2.4 Davidson

The dataset is sourced from (Davidson et al., 2017). The data is compiled using a hate speech lexicon, and all the instances are from Twitter. A minimum of 3 coders labeled tweets into classes Hate speech, Offensive, and Neither. The final sample consisted of $\sim$ 24,000 examples, and only about 5% fell into the Hate Speech class. We map the toxicity score using the formula - (3$*$(# hate speech annotations)+2$*$(# offensive annotations)+(# neither annotations))/No.of labelers. We then normalise this value between 0 and 1.

2.2.5 Founta

Similar to the previous dataset, (Founta et al., 2018) analyzed comments from Twitter and published a dataset with $\sim$ 80k examples. It has three labels (0, 1, 2) with an increasing level of toxicity. We scaled it between 0 and 1 by normalizing it.

3 Methodology

We preprocessed the datasets using standard techniques like stemming, lemmatization, removing contractions, and hyperlinks. For the toxic severity rating, we first tried traditional techniques

[1] https://www.kaggle.com/c/jigsaw-toxic-severity-rating

[2] https://tinyurl.com/2p85bsnj

[3] https://tinyurl.com/9cbyp3ry

like TF-IDF (Rajaraman and Ullman, 2011) and doc2vec (Le and Mikolov, 2014) based regressors to set the baseline. We further add other deep learning setups based on Transformers (Vaswani et al., 2017) to check if the scores improve further.

3.1 Baselines

Initially, we used TF-IDF and Doc2Vec as feature extractors. **TFIDF** is a method to find the importance of a word to a document in a text corpus (Rajaraman and Ullman, 2011). Doc2Vec is an unsupervised method to represent a document as a vector. To train using these features, we use ridge regression, which enhances linear regression by adding L2 regularization.

We used a hyperparameter optimization framework, Optuna, to automate the hyperparameter search for TFIDF. We found the Tfidf vectorizer to work best with the 'charwb' analyzer, n-gram range (3,5) & vocabulary of $\sim$ 30k most frequent words. The ridge regressor had a regularization strength of ~ 1.

Doc2Vec was trained with a feature vector of size 300, learning rate α of 0.025. Both distributed memory and distributed bag of words methods were tested. As the performance was unsatisfactory, we did not conduct hyperparameter tuning for doc2vec.

3.2 Transformers

We take a pre-trained transformers model (Vaswani et al., 2017) that outputs a 768-dimensional vector representation of an input sentence. As this output cannot be directly used as a score for toxicity, we added a single linear layer on top of the encoder to get a single value for toxicity. As we feed input data, the entire pre-trained transformers model and the additional untrained regression layer is trained on our specific task. We focused on tuning hyperparameters manually instead of using any hyperparameter search library due to resource constraints. All the transformers were trained for three epochs with a batch size of 16.

In the following section, we discuss the specifics of the pre-trained models used in detail.

3.2.1 bert-base-multilingual-cased (M-BERT)

This language representation model is a modification of BERT, introduced by (Devlin et al., 2018). It was pretrained on a large corpus of multilingual data from Wikipedia with the objective of Masked language modeling(MLM) in a self-supervised setting. In the masked language model pre-training, the model learns using predicting some of the mask tokens in the text, and it should also be noted that this model is case sensitive.

3.2.2 bert-base-uncased (BERT)

Similar to the above model, this was also pretrained using MLM objective, except this model was trained only on English text corpus, specifically on the BookCorpus, and is not case sensitive.

3.2.3 Hate-speech-CNERG/dehatebert-mono-English(dehateBERT)

(Aluru et al., 2020) benchmarked hate speech classification models for 9 different languages and 16 datasets. All their models are based on the multilingual BERT model. We used their model that was finetuned on an English text corpus.

3.2.4 cardiffnlp/twitter-roberta-base-hate(HRoBERTa)

This model is derived from the RoBERTa-base model (Liu et al., 2019) trained on $\sim$ 58M tweets and finetuned on for hate speech detection with the TweetEval benchmark (Barbieri et al., 2020). Unlike the previous two models, this is an end-to-end regression model, meaning given a sentence, it directly outputs a number between 0 and 1.

3.2.5 GroNLP/hateBERT

This is a re-trained BERT model for abusive language detection in English by (Caselli et al., 2020). It was trained using MLM objective on RAL-E, a large-scale dataset of Reddit comments in English.

3.2.6 sentence-transformer/mpnet-base-v2(mpnet)

This is a sentence embedding model introduced by (Reimers and Gurevych, 2019) trained using a self-supervised contrastive learning objective. It is trained on 1 billion sentence pairs and is based on the pretrained Masked and Permuted Network introduced by (Song et al., 2020). It solves the problems of MLM in BERT and PLM (permuted language modeling) in XLNet and achieves better accuracy.

3.3 Ensembles

Finally, we experimented with ensembles of the models described in the previous sections. To do the ensembling, we predict the scores for a typical post using various models and then combine the

Less Toxic Sentence	More Toxic Sentence
real encyclopedia So f**k you f**king all f*g f**ks	vandalism will you please stop vandalizing my talk page ? it very irritating and borderlining outright harassment
Hey f*ck you c*nt bag ! you dont own wikpedia and please report my i want to close my account any way you creep	Please stop your disruptive editing If your vandalism continues you will be blocked from editing Wikipedia andemu
i said P**S OFF YOU GEEKY LITTLE S**T	And yes people should recognize that but they usually do not One of the first objections you hear directed at someone who says he favors free markets is invariably There no such thing as a free market Ridiculously trivial
F**k up.	Wikinfo, and purge devils from here. Muzemike damned for being enemy of God: I was banned for this: So be damned, MuZemike, daemon in human skin.
Why don't you go s*rew Why don't you go s*rew instead of harassing me?	I also think ... For example i can say that Muslims go to mosques, pray, beat, up their wives, blow, up, etc, what do atheists do?.

Table 1: Samples mislabeled by human labeler (top 3) and model misclassifications (bottom 2). The highlighted text denotes how words affect the model predictions. Darker highlight denotes that the model is paying more attention to that words.

scores using a weighted average. The weights are decided based on the performance of the validation dataset. We used the weights as a variable using the Limited-memory BFGS (LM-BFGS) method, which is an optimization function in the family of quasi-Newton methods that approximates the Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) using a limited amount of computer memory. It is a popular algorithm for parameter estimation in machine learning. The algorithm's target problem is to minimize $f(x)$ over unconstrained values of the real-vector x where f is a differentiable scalar function.

Dataset	Models	Val. Acc.	Test Acc.
Ruddit	TF-IDF	57.54	69.38
	M-BERT	59.83	74.71
	BERT	60.71	78.41
	HRoBERTa (A)	**61.06**	**79.47**
	hateBERT	60.69	78.46
	dehateBERT	58.52	71.28
JTC	TF-IDF	61.01	78.57
	doc2vec	59.87	68.80
	M-BERT (B)	61.31	79.17
	BERT (C)	61.32	78.79
	HRoBERTa (D)	**61.53**	**80.16**
	hateBERT (E)	61.25	78.90
	dehateBERT	59.81	74.95
Founta	TF-IDF	64.58	72.66
	BERT	51.50	75.67
Toxic Unintended	TF-IDF	62.64	72.47
	BERT	59.92	77.70
Davidson	TF-IDF	62.64	72.47
	BERT	52.38	76.64
A+B+C+D+E		**76**	**80.74**

Table 2: Performance on Jigsaw Rate Severity of Toxic Comment Dataset for the validation and entire test dataset.

4 Results and Inference

In this section, we present a detailed analysis of the performance of our models.

4.1 Comparative study of performance

Table 2 shows the performance of our model on the validation dataset and total test dataset. As expected, the transformer-based approaches outperform the traditional approaches like TF-IDF/doc2vec. We found that HRoBERTa model performed the best among the transformers models. It is interesting to note that BERT & M-BERT give comparable results to language models already pretrained for detecting toxicity(hateBERT & dehateBERT). Experiments on the transformed Founta, Davidson, and Toxic unintended did not give good scores; hence we did not perform further experiments on them.

Our team secured a rank of 145 out of 2301 in the Kaggle Jigsaw Rate Severity of Toxic Comments Competition with an accuracy of 79.84% in the private leaderboard. However, one of our ensembles which was not part of our final submission, performed even better. We achieved an accuracy of 80.74% in the final standings (Table 2). It is also worth mentioning that our approach was quite similar to the winning approach(accuracy of 81.39%), except they used Genetic Algorithm (Xu) to find weights for their ensemble. Our method using an ensemble of 5 models performs half a percent worse than their 15 ensemble model.

4.2 LIME

We also conducted local interpretable model-agnostic explanations extensively on our best model (HRoBERTa) to identify potential issues with model predictions on the validation dataset. The validation set contains pairs of sentences labeled as less toxic and more toxic.

We ranked the model predictions and checked the top 100 wrong predictions manually. The top 100 wrong predictions were found by ranking the difference between the score assigned to less toxic to more toxic sentence. For most of the cases, it was not the model but the human annotator who was at fault. There were several cases where we found difficult to select the more toxic comment. We found 68 samples where the annotator was wrong, 3 samples where our model was wrong and found 29 samples to be equally toxic. We add some of the samples from each category in Table 1.

The top 100 worst predictions were selected on the following basis. At first, for each sample we compared the scores generated by our model. The samples where the more toxic sentence had a lower score than less toxic sentence(similarly, less toxic with higher score than more toxic sentence) were marked as incorrectly classified samples. For all the incorrect classifications, the difference between the scores generated for less toxic and more toxic comment was computed. This list was sorted in descending order according to the difference. The top 100 samples were selected for LIME analysis. Hence, the samples where the model is more confident about the prediction yet wrong are selected. We believe that this method captures the worst errors of the model.

5 Conclusion

We present a detailed analysis of both the traditional and modern machine learning algorithms for toxicity detection. Instead of a binary classification, a relatively new notion of toxic speech rating is explored. The existing toxicity classification datasets are modified to train the models to output a toxicity score in a continuous range. We test our models on a new dataset proposed by Jigsaw. Additionally we present the LIME analysis to understand the model predictions.

References

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Lora Aroyo, Lucas Dixon, Nithum Thain, Olivia Redfield, and Rachel Rosen. 2019. Crowdsourcing subjective tasks: the case study of understanding toxicity in online discussions. In *Companion proceedings of the 2019 world wide web conference*, pages 1100–1105.

Somnath Banerjee, Maulindu Sarkar, Nancy Agrawal, Punyajoy Saha, and Mithun Das. 2021. Exploring transformer based models to identify hate speech and offensive content in english and indo-aryan languages. *arXiv preprint arXiv:2111.13974*.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022a. Data bootstrapping approaches to improve low resource abusive language detection for indic languages. pages 32–42.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022b. hate-alert@ dravidianlangtech-acl2022: Ensembling multi-modalities for tamil trollmeme classification. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 51–57.

Mithun Das, Somnath Banerjee, and Punyajoy Saha. 2021a. Abusive and threatening language detection in urdu using boosting based and bert based models: A comparative approach. *arXiv preprint arXiv:2111.14830*.

Mithun Das, Binny Mathew, Punyajoy Saha, Pawan Goyal, and Animesh Mukherjee. 2020. Hate speech in online social media. *ACM SIGWEB Newsletter*, (Autumn):1–8.

Mithun Das, Punyajoy Saha, Ritam Dutt, Pawan Goyal, Animesh Mukherjee, and Binny Mathew. 2021b. You too brutus! trapping hateful users in social media: Challenges, solutions & insights. In *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, pages 79–89.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Facebook. 2022. Facebook community standards. (Accessed on 04/15/2022).

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.

Rishav Hada, Sohi Sudhir, Pushkar Mishra, Helen Yannakoudakis, Saif M. Mohammad, and Ekaterina Shutova. 2021. Ruddit: Norms of offensiveness for English Reddit comments. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2700–2717, Online. Association for Computational Linguistics.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Jana Kurrek, Haji Mohammad Saleem, and Derek Ruths. 2020. Towards a comprehensive taxonomy and large-scale annotated corpus for online slur usage. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 138–149, Online. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Data Mining*, page 1–17. Cambridge University Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Punyajoy Saha, Binny Mathew, Kiran Garimella, and Animesh Mukherjee. 2021. "short is the road that leads from fear to hate": Fear speech in indian whatsapp groups. In *Proceedings of the Web Conference 2021*, pages 1110–1121.

Mike Schroepfer. 2021. Update on our progress on ai and hate speech detection. (Accessed on 04/16/2022).

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Twitter. 2022. Twitter's policy on hateful conduct | twitter help. (Accessed on 04/15/2022).

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*, 30.

Guanshuo Xu. Jigsaw rate severity of toxic comments winner's approach. (Accessed on 04/16/2022).