# COMPARISON OF CLASSIFICATION MODELS ON UNINTENDED BIAS IN TOXICITY CLASSIFICATION

Mijeong Ban[1] and Nathaniel Burgdorfer[2]

[1]Stevens Institute of Technology , mban1@stevens.edu
[2]Stevens Institute of Technology , nburgdor@stevens.edu

**ABSTRACT**

The goal of this work is to approach the problem of text classification in a comparative manner. To the best of our knowledge, there is a general consensus that one of the relative state-of-the-art methods in NLP for vectorization and word embedding is the BERT [2] pre-trained model and it's derivatives; however, there is no real consensus as to which learning method is best for the task of text classification, especially with regards to model bias. There are many models that can be used for the task of text classification, each with their benefits and drawbacks. We look to draw a comparison between different learning models on the basis of specifically measuring unintended model bias as well as prediction accuracy. We look to add this extra dimension to evaluation as it is an important part of ensuring models are as accurate and as appropriate as they should be, even when the data may be skewed or biased in one way or another.

## 1 Introduction

When discussing text classification and understanding, it sometimes may not be enough to just measure the accuracy of the model evaluated on a given data set. Most often in the process of data collection, there can exist some skew or some bias in the data being collected and in the layout or distribution of the collected data. When the goal of machine learning and classification tasks is most often to recover the true distribution for the data involved in the task, it is of utmost importance that the specific sampled data set that we use in this process does not affect the outcome of the model and lead to a distribution far from the true distribution. In this work, we look to analyze and compare two different classification network models for the task of text classification on the toxicity in online comments. For this task, it is important that we not only compare traditional classification accuracy, but also take into account a bias metric.

## 2 Background/Related work

In the context of toxicity in online comments, there can be significant bias towards subgroups mentioned in the comments. Models that deal with classifying this type of content often times incorrectly learn to associate certain subgroups with certain labels. Based on existent bias in the collected data, whether this be due to a high volume of certain classes in the data or due to repeated samples of certain pairings (e.g. comments including the subgroup 'white' having the label 'toxic'), a model may learn to just associate the existence of a subgroup in a comment with a label of 'toxic' or 'not toxic' instead of learning to classify the entire context of the comment. In this work, we look to compare the results of different machine learning models with a focused evaluation on error resulting from model bias as well as overall prediction accuracy. We will be referencing some current state-of-the-art architectures for text classification and semantic analysis involving CNNs [3][4] and RNNs [5][6]. We look to experimentally analyze how each architecture responds and learns from the data and to compare how each may avoid biased word association and instead focus on the context of comments.

## 3 Datasets

As the Civil Comments platform made their $\sim 2m$ public comments available to help researchers understand and improve civility in online conversations, Jigsaw helped to extend annotation of this data by asking human raters to rate the toxicity of each comment. The datasets contain train data, test data and other extra datasets for additional research. Train data includes the text of the individual comment in the *comment_text* column and a toxicity label in the *target* column. The *target* values are fractional values which represent the fraction of 10 human raters who believed the label fit the comment with $target >= 0.5$ being considered to be in the positive class (*very toxic / toxic*), and $target < 0.5$ being considered to be in the negative class (*hard to say / not toxic*). Test data has only the comment text values so that models can predict the target toxicity. The train data also has additional toxicity subtype attributes for research, such as *severe_toxicity, obscene, threat, insult, identity_attack, sexual_explicit*. Furthermore, a subset of comments have identity attributes, *male, female, homosexual_gay_or_lesbian, etc*, which shows what identities that were mentioned in the comment.

## 4 Method

We look to use the BERT model as the input embedding for different text classification models, specifically CNN and RNN model architectures, in an attempt to compare the performance between them. We will be using the Kaggle competition, 'Jigsaw Unintended Bias in Toxicity Classification', as the context for our investigation. We will construct several different CNN and RNN models, training each and evaluating them on their prediction accuracy and bias. The comparison of these models will include their respective time requirements (training time), complexity (model parameters), and model parameter values (learning rate, optimization, loss, etc). The comparison will be focused on their evaluation, detailed in section 5, on the data provided by the competition.

## 5 Evaluation

To evaluate two models, we will use the evaluation method that the competition provided, ROC-AUC [1]. This evaluation method is a newly developed metrics which contains three different submetrics(Subgroup AUC, BPSN and BNSP), so that we can properly evaluate the overall performance with respect to overall classification accuracy and unintended bias evaluation. The goal is to restrict the data samples during each sub-metric evaluation in order to extract different bias evaluations from the model. In the first sub-metric, we look at each specific subgroup and measure the accuracy score within the subgroup. This tells us how well the model performs at classifying comments within a subgroup. The BPSN (Background Positive Subgroup Negative) sub-metric focuses on negative (non-toxic) samples within the subgroup and positive (toxic) samples outside of the subgroup. This sub-metric is a measure of the false-positive rate of the model. The lower this score, the more biased the model is in classifying a comment containing a subgroup as toxic when it has a true value of non-toxic. Furthermore, the BNSP (Background Negative Subgroup Positive) sub-metric focuses on positive (toxic) samples withing the subgroup and negative (non-toxic) samples outside the subgroup. This sub-metric is a measure of the false-negative rate of the model. The lower the BNSP score, the more biased the model is in classifying comments containing a subgroup as non-toxic when it has a true value of toxic. The latter two sub-metrics also include the 'Background' data in the metric to incorporate the accuracy of the model's predictions of the incorrect label for the subgroup.

The overall score is calculated as follows:

$$M_p(m_s) = \left( \frac{1}{N} \sum_{s=1}^{N} m_s^p \right)^{\frac{1}{p}}$$

where,

$$\begin{aligned} M_p &= \text{the } p^{th} \text{ power-mean function} \\ m_s &= \text{the bias metric } m \text{ calculated for subgroup } s \\ N &= \text{the number of identity subgroups} \end{aligned}$$

The above equation is the per-identity bias AUC score. This paper combines this score with the overall AUC score to obtain the final evaluation score,

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a})$$

where,

$A =$ the number of sub-metrics (3)

$m_{s,a} =$ the bias metric for subgroup $s$ using sub-metric $a$

$w_a =$ a weighting for the relative importance of each metric; (all four are set to 0.25 for this evaluation)

## References


[1] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019. URL http://arxiv.org/abs/1903.04561.

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

[3] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. pages 562–570, July 2017. doi: 10.18653/v1/P17-1052. URL https://www.aclweb.org/anthology/P17-1052.

[4] Yoon Kim. Convolutional neural networks for sentence classification. pages 1746–1751, October 2014. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

[5] Baoxin Wang. Disconnected recurrent neural networks for text categorization. pages 2311–2320, July 2018. doi: 10.18653/v1/P18-1215. URL https://www.aclweb.org/anthology/P18-1215.

[6] Zeping Yu and Gongshen Liu. Sliced recurrent neural networks. *CoRR*, abs/1807.02291, 2018. URL http://arxiv.org/abs/1807.02291.