Toxic Comment Classification with Unintended Bias

Levi Lian levilian@stanford.edu
Rui Liu rui3@stanford.edu
Sherry Li xjli1013@stanford.edu

Abstract

We investigate two methods for classifying toxic comments with unintended bias: an deep learning approach that relies on GloVe embeddings and Recurrent Neural Networks(RNN), and a contexual representational appraoch with Bidirectional Encoder Representations from Transformers(BERT). Using the Jigsaw dataset, we show that a vanilla BERT fine-tuned on 1.4M comments can produce better performance than an RNN with GloVe embeddings (after hyperparameter tuning). Detailed error analysis explains the efficacy of BERT in classifying toxic comments without introducing the same amount of unintended bias that RNN has.

1 Introduction

Toxic comment is verbal violence that can be considered rude, disrespectful, or even discriminatory. However, with the proliferation of social media and online communications, toxic comment has found its way to everywhere on the Internet, often resulting in ad hominem and cyberbullying. According to a 2014 Pew Research report, 73% of adult internet users have seen others being verbally abused online, and 40% have personally experienced it [Duggan, 2014]. The impact of toxic comments has negative spillover effects. A study by Wikipedia foundation shows that 54% of those who had experienced online harassment were less willing to participate thereafter [Wulczyn et al., 2017].

To tackle this severe problem, attempts have been made by developing crowdsourcing voting schemes or the capacity to denounce a comment [Wulczyn et al., 2017]. Nevertheless, those techniques do not perform well to predict a potential toxicity. To automatically identify toxic comment, Google and Jigsaw has recently launched a project called Perspective, which calls for implementing machine learning methods to automatically detect online insults, harassment, and abusive speech [Hosseini et al., 2017]. However, while accuracy score soars to above 90%, unintended bias seeps in. Comments that include certain identity types, such as "black" and "gay" are inaccurately flagged as toxic more often than those without.

Our project therefore intends to improve text classification models to identify toxic comments while accounting for bias. We used comments collected by Civil APIs, later annotated by Google Jigsaw, as our dataset.

Our main contribution lies in comparing the results of bidirectional LSTM with Gated Recurrent Unit(GRU) plus pre-trained word embeddings with those of Bert, and our detailed error analysis examines the reasons behind Bert's better performance especially when accounting for unintended biases. By analyzing false negatives and false positives of the ensemble we get insights about open challenges that all of the approaches share. The analysis points to common errors of all current approaches. We propose directions for future work based on these unsolved challenges.

2 Related Work

2.1 Task Definition

Toxic comment classification has been scoped in many different ways. One way is to simply deal with classifying comments as binary classification, toxic or non-toxic. The other approach is more nuanced; it focuses on subtypes of toxic comments, and classifies based on those subtypes. This is the case with Warner and Hirschberg [Warner and Hirschberg, 2012], who focus on classification of anti-Semitic posts from any other form of hate speech. Badjatiya et al. [Badjatiya et al., 2017] study the detection of racist and sexist tweets with deep neural net-

works. Gitari et al. [Gitari et al., 2015] further classify hateful comments into weak and strong hate, a strategy that the Jigsaw team used to produce "toxic" and "severe toxicity" labels.

The complexity with which one comment can be grouped into multiple subtypes and one data labeller can interpret the comments in many different ways has resulted in further study of toxic comment classification. In particular, Castelle [Castelle, 2018] studied the linguistic ideologies of abusive language, and showed that the primacy of context in the dynamic construction of meaning [Levinson, 1983] can provide additional context enrichment. For example. a single utterance usually has poor performance. However, the linguistic theories of pragmatics and metapragmatics showed improvement of macro F-1 accuracy score from 86% to 89%.

2.2 Toxic Comments Classification

Toxic comment identification is a supervised classification task and has seen approaches in both traditional feature engineering and deep neural networks. Danescu-Niculescu-Mizil et al. build a hand-picked politeness classifier that matches human-level performance (84% versus 87% of human performance) [Danescu-Niculescu-Mizil et al., 2013]. It uses human-annotated data from Wikipedia talk pages and requests on question-answer forum StackExchange. The linguistic features are picked so that they represent human consensus on how much they weigh on the scale of politeness. These features include indirection, deference, impersonalization and modality, which are mentioned in previous sociolinguistic theories.

While in the first case manually selected features are combined into input vectors and directly used for classification, neural network approaches are supposed to automatically learn abstract features above these input features. Neural network approaches appear to be more effective for learning (Zhang and Luo, 2018), while feature-based approaches preserve some sort of explainability.

The state-of-the-art in this field in 2017 was bidirectional recurrent neural networks with attention [Pavlopoulos et al., 2017] and the use of pre-trained word embeddings [Badjatiya et al., 2017].

2.3 Contextual Word Embeddings

Recently, the introduction of Embeddings from Language Models (ELMo) [Peters et al., 2018] and Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] has brought the field of Natural Language Processing its moment of "ImageNet". These techniques generate embeddings based on the context in which a word appears. This has led to large improvements when applying to many supervised NLP tasks such as question answering [Rajpurkar et al., 2018] and classification [Chatterjee et al., 2019].

Peters et al. proposed ELMo, a character based pretrained model., that can handle out of vocabulary words because of the characterization. It looks at the entire sentence before assigning each word in an embedding. It uses a bi-drectional LSTM on an unsupervised task to create language modeling. The learnt representations are, however, words.

BERT, similarly, can account for context just like ELMo [Devlin et al., 2018]. However, it uses Transformers, which deal with long-term dependancies better than LSTMs. BERT represents input as subwords and learns embeddings for subwords. Representing input as subwords as opposed to words has become the latest trend because it strikes a balance between character based and word based representations - the most important benefit being avoidance of OOV (out of vocabulary) cases which the other two models (Glove, Word2vec) suffer from.

Because these two methods are all pre-trained on two unsupervised tasks, masked language modeling and next sentence prediction. They can be fine-tuned to perform on downstream specific tasks, similar to GloVe [Pennington et al., 2014].

2.4 Unintended Bias

Dixon et al. illustrate a new approach to measure and mitigate unintended bias in machine learning classifiers, with Wikipedia toxic comment classifier as an example [Dixon et al., 2018]. It makes a distinction of unfairness and unintended bias, defined as unequal model performance for comments containing particular identity terms such as gay or straight. The authors then considered the representation of training data according to different features such as comment length and frequency of identity terms in toxic comments and overall. They found that unintended bias came from under-representation of the training data, and they mitigated the effect by adding data from a different domain with an unsupervised strategy.

They then identified three evaluation metrics for

bias. One is AUC for different identity groups, which shows model quality. Second is false positive rates and false negative rates, which is a well-accepted definition of fairness. Third is a newly proposed method that combines the two called pinned AUC. It turned out that this strategy lowered the differences of false positive rates, false negative rates, and pinned AUC across identity groups (a 30% reduction), demonstrating improvement without suffering from quality depreciation (0.959 versus 0.960 of the original model). We will use the same evaluation metric in this paper.

3 Hypotheses

Our hypothesis is that using contextual representations can account for unintended bias that other methods, deep neural networks and traditional machine learning, suffer from. That is, by using BERT, we will achieve better performance than if we use LSTM/GRU with GloVe or Fasttext. The input to our algorithm is the text of a comment with relevant, human-annotated features that indicate its toxicity subtype attributes (e.g., insult, threat, etc.) and identity attributes (e.g., female, homosexual, white, etc.). We then use an RNN with GloVe as pretrained embeddings to output a predicted toxicity rating between 0 and 1. To evaluate this hypothesis, we use the following setup: We compare two set of methods explored in the literature review, SVM and RNN coupled with GloVe embeddings, with BERT as vanilla embeddings For evaluation, test set examples with target ¿= 0.5 will be considered to be in the positive class (toxic). We use the Area under the Receiver Operating Curve (ROC AUC) to measure classifier performance, combined with the additional bias AUC to evaluate the performance after accounting for identity types. In the following sections we will explain in more detail our data, methodology and evaluation metric.

4 Data

We use the data set supplied by the 2019 Jigsaw Unintended Bias in Toxicity Classification Competition[1] for our final project. We split the data into training ($1,335,618$ observations), validation ($445,205$ observations) and ($96,846$ observations). The training and test sets combined include $1,780,823$ annotated user comments collected from Wikipedia talk pages and is the largest publicly available for the task. These comments were annotated by human raters with the six labels severe_toxicity, insult, threat, obscene, identity_attack, and target. Comments can be associated with multiple classes at once, which frames the task as a multi-label classification problem. Jigsaw has not published official definitions for the six classes. However, they do state that they defined a toxic comment as a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion. Moreover, a subset of comments were labelled with the following identity attributes: male, female, transgender, other gender, heterosexual, homosexual gay or lesbian, bisexual, other sexual orientation, christian, jewish, muslim, hindu, buddhist, atheist, other religion, black, white, asian, latino, other race or ethnicity, physical disability, intellectual or learning disability, psychiatric or mental illness, other disability. Figure 1 shows a screenshot of the dateset.

These identity attributes will be the additional features passed in along with the comment text and toxicity subgroups to our model. They will be critical to validate our hypothesis of whether adding these bias features will not reduce overall performance. The dataset features an imbalanced class

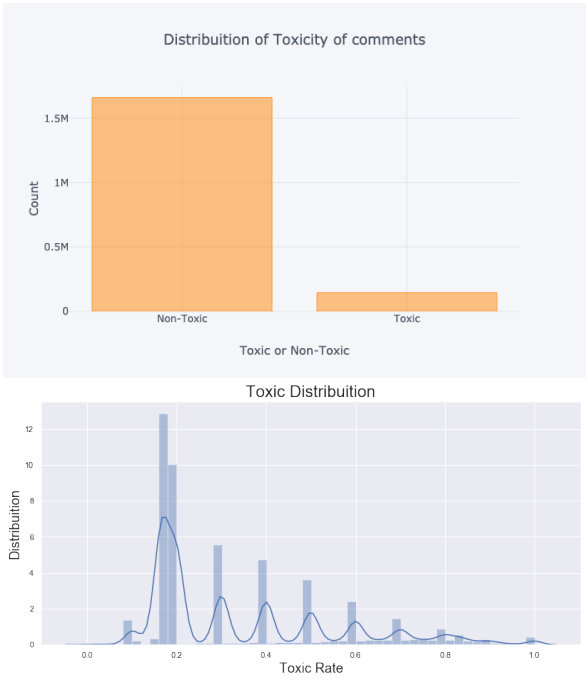

Figure 1: Distribution of of toxic vs non-toxic comments in the entire dataset; Distribution of comments according to their toxicity score

[1] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data

distribution, shown below in 1. $1,265,140$ samples fall under the majority class matching none of the six categories, i.e., target being zero, whereas $515,683$ samples belong to at least one of the other classes. While the insult class includes 24.1% of the samples, only 1.9% are labeled as sexual_explicit, marking the smallest class. According to the Jigsaw official page, comments were collected from the Civil Comments platform shut down, which included all the comments and were later annotated by crowd-sourcing.

Class	**# of occurrences**
total	1804874
target	1660540
severe toxicity	104636
obscene	147175
identity attack	218552
insult	454539
threat	106929
asian	10975
atheist	2253
bisexual	3330
black	19563
buddhist	1366
christian	61360
female	73690
heterosexual	3453
hindu	1557
homosexual gay or lesbian	15307
intellectual or learning disability	2648
jewish	10905
latino	6936
male	80179
muslim	26650
other disability	3545
other gender	2723
other race or ethnicity	18867
other religion	16732
other sexual orientation	4508
physical disability	3227
psychiatric or mental illness	10665
transgender	6120
white	29948

Table 1: Class distribution of Jigsaw dataset. The distribution shows a strong class imbalance

The label "target" is calculated as a fraction of human raters who believed the attribute applied to the given comment. With the test set that only includes the comment texts, we should output the "target" value as a decimal from 0 to 1 for each individual comment test with a fractional value. One example of a comment is "i'm a white woman in my late 60's and believe me, they are not too crazy about me either!!", and values for the toxicity and identity labels for this comment are 0s except that the "female" label is valued at 1.0.

We also looked at the correlations among all the identity and label features to get a sense of their inter-dependency.

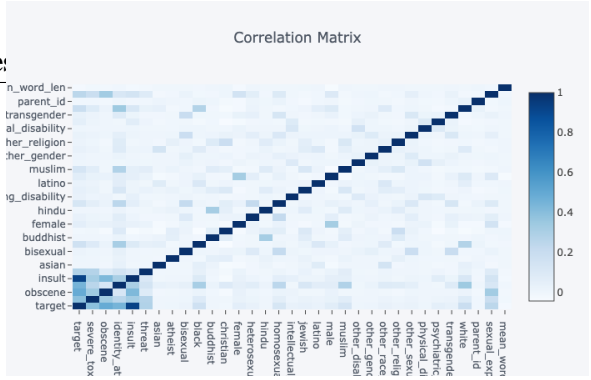

Figure 2: Correlations among all the identity features and label features. The graph shows that the comment being an insult is highly correlated with being a target. This fits with our linguistic common knowledge. Insulting someone seems a good criterion to be removed.

Below are the steps we did on our text for preprocessing:

- Removed stop words.
- Isolated punctuation and contractions.
- Replaced multiple spaces ( ) or . to a single one.
- Removed or isolated emojis and symbols.

5 Metrics for Evaluation

For our evaluation, we implement the original Area Under the Receiver Operating Characteristic Curve (ROC-AUC) and a variation of it for the full test set to measure accuracy.

Since the original ROC-AUC is straightforward, we will only discuss the variation here. ROC-AUC measures the probability that a randomly chosen negative example will receive a lower score than a randomly chosen positive example, i.e. that the two will be correctly ordered. ROC-AUC is preferable in our case because it is threshold agnostic as opposed to macro average F-1 score. We follow the Kaggle official metric and [Borkan

et al., 2019] to incorporate three additional bias AUCs. These three submetrics will be combined to form a generalized mean of bias AUC, taking into account how our model performs against unintended bias. According to Borkan et al. [Borkan et al., 2019], unintended bias includes:

(1) being unable to distinguish toxic and non-toxic comments that mention the specific identity (Subgroup AUC),

(2) having high false positive rates for comments that mention the identity but turn out to be non-toxic (The Background Positive Subgroup Negative (BPSN) AUC),

(3) and having high true negative rates for comments that mention the identity but are indeed toxic (Background Negative Subgroup Positive (BNSP) AUC).

The aforementioned bias AUCs can be combined per-identity into one overall measure, as shown below[2]

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

where:

Mp = the pth power-mean function

ms = the bias metric m calculated for subgroup s

N = number of identity subgroups

$$\text{score} = w_0 AUC_{\text{overall}} + \sum_{a=1}^{A} w_a M_p(m_{s,a}) \qquad (2)$$

where:

A = number of submetrics (i.e., 3)

$m_{s,a}$ = bias metric for identity subgroup s using submetric a

w_a = a weighting for the relative importance of each submetric; all four w values set to 0.25

6 Models

We used TF-IDF with Logistic Regression and SVM with NB features(NBSVM) for our baseline. Then we used one layer of LSTM and GRU with GloVe embeddings for our RNN model. We finally tested the performance of BERT.

[2]https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview/evaluation

6.1 Baselines

For the baselines, we use word TF-IDF as frequency-based text representations and apply them to traditional machine learning methods. Later, they will be compared with the use of word embeddings being applied to deep learning methods. TF-IDF weight is a common feature used in natural language understanding. The weight measures how important a vocabulary is to a document in a collection or corpus. Since different types of toxic comments are closely related to toxic vocabulary, TF-IDF is a good baseline for features. We compare two machine learning models that take TF-IDF as input features and apply NBSVM. This baseline was identified by Wang and Manning [Wang and Manning, 2012] as a simple yet powerful method that exceeds the state-of-the-art published results. We deem it fitting for our use as a strong baseline.

The Logistic Regression (LR) and SVM algorithms are widely used for classification tasks. Research from Waseem and Hovy [Waseem and Hovy, 2016] shows that word n-grams is simple but effective for toxicity classification. For this reason we investigate the use of word n-grams for LR and SVM models as a baseline.

6.2 Recurrent Neural Networks

According to Chen [Chen et al., 2016], Recurrent Neural Networks (RNNs) break down a comment as a sequence of words and can therefore control long-text learning. We will use two different RNN approaches: An LSTM (Long-Short-Term-Memory Network) and a bidirectional LSTM. Our LSTM model takes a sequence of words as input (which is why we eliminated comments with more than 50 words because even LSTM will fail to capture the temporal relationships). The embedding layer will then transform words to vector representations, and the LSTM layer will process the sequence of word embeddings. Finally, a dense layer will output the result of this multi-class classification.

Bidirectional LSTM, on the other hand, can compensate certain errors on long range dependencies because it handles both forward and backward connections. It will use two LSTM layers to process the input sequence in opposite directions, i.e., forward and backward order of words. The output will be averaged to account for both directions.

We also conducted an extensive parameter-search to find the best dropout rate, and learning rates and number of epochs. We will present the details below.

6.3 Transfer Learning with Bert

BERT built upon a collection of contextual representations such as ELMo, ULMFit, Semi-supervised Sequence Learning and Generative Pre-Training. However, BERT is both deeply bidirectional (much deeper than bidirectional LSTM) and unsupervised. It is pretrained with only a plain text corpus, Wikipedia. To implement Transfer Learning with Bert, this project uses BERT open-sourced by Google [3]. We apply the BERT-LARGE (Uncased (Whole Word Masking): 24-layer, 1024-hidden, 16-heads, 340M parameters) onto our training model. This BERT then learns the specific context of each word in our training dataset, i.e., fine-tuning. Finally, we test the model on our test set based on the aforementioned evaluation metric.

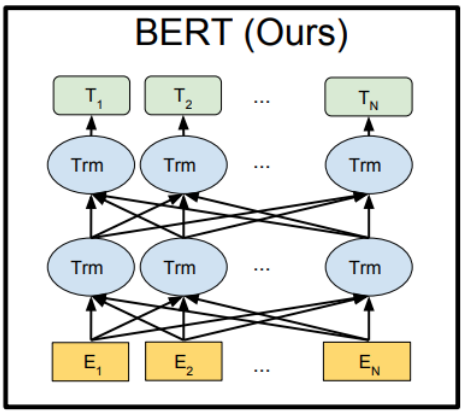

Figure 3: Pre-trained BERT model implementation. The figure shows that BERT is bidirectional from layer to layer, producing the final contextualized representation by taking into account the meaning of whole sentence, rather than the previous and the next word.

7 Experiments

We conducted three sets of experiments on our RNN model to search for the best hyperparameters. Each set of experiments was conducted on our original dataset. The data split follows the aforementioned criteria. In all cases, the raw comment text was tokenized into words and punctuation marks, which also follows our preprocessing. We also mapped tokens to GloVe embedding. Exceptions were mapped to a randomly initialized UNK vector, similar to the CS224U assignment code.

Our best performance for LSTM+GRU model (with GloVe embeddings) comes with a learning rate of 0.0005, dropout rate of 0.1 and 7 epochs.

Drop. rate (%)	10	30	50
AUC	0.93197	0.921875	0.91920
Learning rate	0.0005	0.005	0.5
AUC	0.93352	0.93197	0.49843
Number of epochs	3	5	7
AUC	0.92884	0.93197	0.93401

Table 2: LSTM model performance with respect to dropout rate, learning rate and number of epochs

8 Results and Discussion

Methodology	**Score**
NBSVM	0.89930
Logistic Regression	0.90148
LSTM+GRU with GloVe	0.93401
BERT	0.94591

Table 3: Results of all methodologies implemented. BERT achieved the best performance based on our evaluation metric that accounts for unintended bias

The results above show that the vanilla BERT beat the performance of the best of LSTM with GloVe after our hyperparameter tuning. Based on our literature review, this result is both surprising and reasonable. In part, BERT has incorporated the missing pieces of ELMo with long-term dependencies. The main issue with comment classification is long-term dependencies and contextual representation, which is solved by BERT.

At the same time, the vanilla BERT actually learns the specific context of each sentence, thereby producing significantly better results than GloVe. Even after training on downstream dataset, GloVe does not contain the information-rich context that BERT has before fine-tuning. This can be best illustrated in the following example. In figure 4, GloVe has static representations of a word in different context whereas in figure 5, BERT produces different geometric structures.

Moreover, if we look at one specific example, we can calculate the cosine similarity between two contextual embeddings of a word. Take "*Maybe*

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

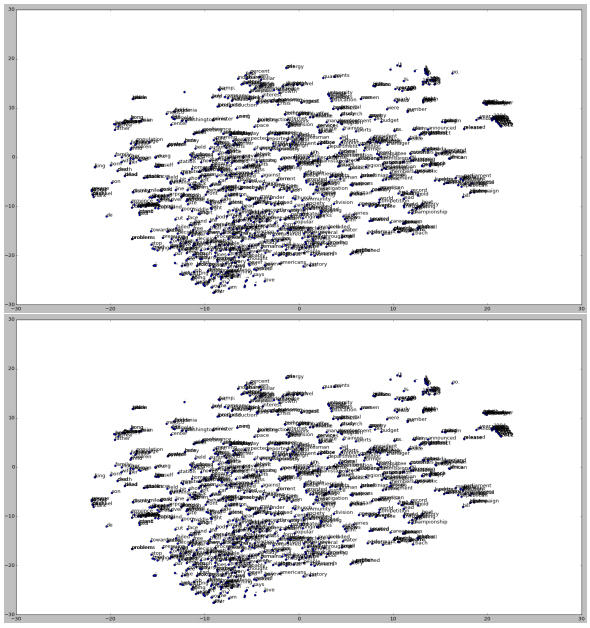

Figure 4: GloVe representations do not account for context. For a word that means differently in two different sentences, they would show the same embeddings

one day we will be screwed . Screw you, Donald Trump ." as an example. Note that we already isolated the stop word ".". In figure 6, the embeddings for the word "screwed" and "screw" are clearly different, and in effect their cosine similarity is 0.54, which shows that they almost take on opposite meaning and therefore toxicity in different contexts.

This explains why that when accounting for unintended biases, specifically those targeted at identity groups such as LGBTQ and ethnic groups, BERT is able to take meaning into context.

9 Error Analysis

We conducted a thorough error analysis to pinpoint mistakes made by the RNN classifier by were caught by BERT.

First we look at false positive examples[4]. The classifier mislabelled the following sentence as toxic (0.51362): "*Or drive the speed limit, or serve black people at your lunch counter, or not sell meth to school children, or wait for permission to immigrate.*" The score is so close to below 0.5 so it is a close catch. The sentence contains several words that potentially misleading to the classifier "black people", "meth", and "immigrate".

[4]The examples in this work may contain offensive language. They have been taken from actual web data and by no means reflect the authors opinion.

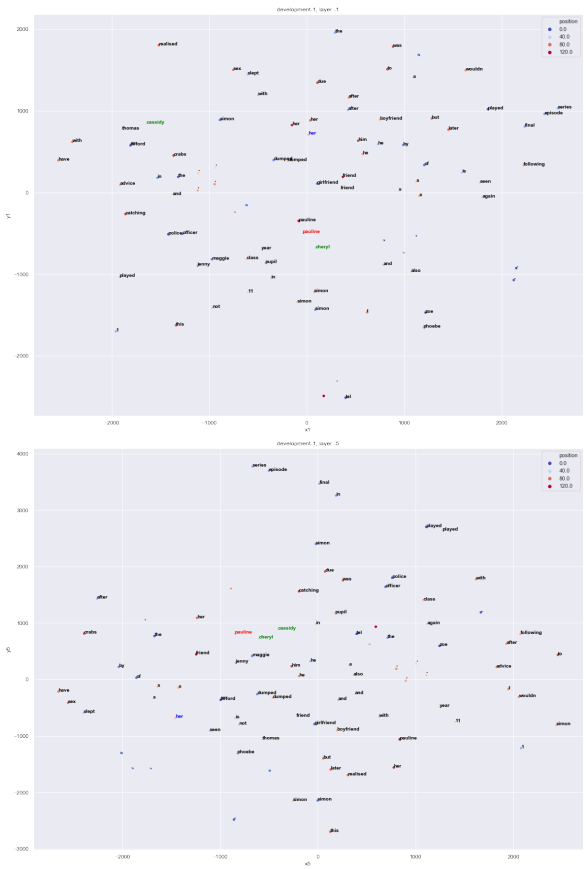

Figure 5: BERT representations, however, do account for context. For a word that means differently in two different sentences, they would show different embeddings

```
Token: 1 Maybe
tensor([ 0.7849,  0.5081,  0.1299,  ..., -0.4393, -0.2719,  1.0371])
Token: 2 one
tensor([-0.4292, -0.1515,  0.5473,  ..., -1.2619, -0.0470,  1.4617])
Token: 3 day
tensor([-0.6517,  0.5459,  0.7170,  ..., -0.8185,  0.0662, -0.8210])
Token: 4 we
tensor([ 0.6770,  0.4690,  0.8428,  ..., -0.0308,  0.9957, -0.4204])
Token: 5 will
tensor([ 0.3685,  0.2550,  0.6730,  ..., -0.5367, -0.0440, -0.3395])
Token: 6 be
tensor([ 0.7428,  0.3272,  0.3777,  ...,  0.3683, -0.0262,  0.2708])
Token: 7 screwed
tensor([ 1.2891, -0.2990,  0.0194,  ..., -0.1220, -0.2343, -0.7568])
Token: 8 .
tensor([ 0.7171,  0.3021, -0.1893,  ..., -0.0853, -0.3967,  0.2321])
Token: 9 Screw
tensor([ 1.2157,  0.6220,  0.6050,  ..., -0.1542, -0.3591,  0.1821])
Token: 10 you,
tensor([-0.3458, -0.0538,  0.8258,  ...,  0.4261, -0.1108,  0.0225])
Token: 11 Donald
tensor([-0.4262, -0.1441,  0.3454,  ..., -0.4003, -0.1221,  0.3416])
Token: 12 Trump
tensor([ 1.2573,  0.1022,  0.4784,  ..., -0.1434,  0.4780,  0.0834])
Token: 13 .
tensor([ 0.0267, -0.1510, -0.0639,  ..., -0.0559, -1.0765,  0.1154])
```

Figure 6: BERT representations, however, do account for context. For a word that means differently in two different sentences, they would show different embeddings

BERT clearly accounts for the contextual meanings in this case to not commit the same mistake. However, we think that more research into BERT's

interpretability, specifically with geometric representations, could further improve its performance. Otherwise we need to implement hand-picked features so that the classifier does not inadvertently discriminate against this innocuous sentence.

Similarly, the following sentence was inaccurately flagged by our RNN model: "*If the shooter was a Muslim man, then Donald Trump would call him a terrorist and a ban on Muslim immigration. If the shooter was a Hispanic man, then Trump would call him a terrorist thug and a crackdown on illegal immigration. If the shooter was Asian, then he would be a criminal terrorist and a crackdown on Triads. If the shooter was black, then there would be a crackdown on gang members. If the shooter voted for Hillary Clinton and the Democrats, then he would be a political terrorist and a member of the Antifa.*" The appearance of too many politically charged terms might be the reason behind the error.

The language of sarcasm is also a heated topic of study in NLP. The following example shows that even BERT fails to capture the sarcastic tone in the word "ridiculous" and "juvenile" : "*The America you believe in is a ridiculous and juvenile fantasy. You should read up a little on the idea of the social contract.*"

Now we look at true negative examples. The following sentence that clearly contains fake news URLs and targeted hatred was not captured by the classifier: "*http://www.bullshitexposed.com/scandinavian-socialism-debunked/ https://fee.org/articles/the-myth-of-scandinavian-socialism/ Yes, Scandanavia has moved closer to capitalism. See the above links. Their current problem is immigration policy, allowing in vicious Muslim men who are eager to rape and pillage. Hee. Tough to have your entire world view wrecked.*". Partly, it might be the URLs to blame. Legitimate links should be included as part of the corpus and future training data. Moreover, NLP researchers might want to borrow email filtering methods to correctly flag the appearance of fake news web links.

Statistically, the false positive rate is much higher than the true negative rate. This is most easily reasoned by the data imbalance present in our training data. Far more data are in the non-toxic category, and a significant portion of data has a toxicity score below 0.2. This provides BERT with even more fine-tuning on learning the non-toxic sentence structure and word combinations.

In terms of future directions, we think it will be straightforward to use BERT as first layer embeddings that output to LSTM, and see if this combination boosts performance. It will also be tempting to build a hand-picked feature engineering method particularly for the task of comment classification, and compare and contrast the result with vanilla BERT and BERT-enhanced LSTM. Besides, future research could further pre-train BERT on similarly toxic datasets to learn even more detailed contexts and sentence structure of online comments.

10 Conclusion

This paper evaluated two methods for toxic comment classification problems. Our evaluation model in particular accounts for the unintended bias present in classifiers. We showed that a fine-tuned vanilla BERT outperforms the result of LSTM+GRU(with GloVe embeddings). We also analyzed mistakes made by our RNN model to shed light on BERT's better performance. We propose directions for future research based on these unresolved problems.

Acknowledgments

We would like to thank our project mentor, Cindy Wang, for her guidance us through each milestone of this project, and Christopher Potts, for his teaching of CS224U Natural Language Understanding.

11 Authorship Statement

Liu implemented pre-trained BERT language model, discussed evaluation metrics and datasets, Reviewed literature on text classification and toxic comment identifications. Li implemented logistic regression and NBSVM baseline, fine-tuned RNN model hyperparameters, performed data analysis, and reviewed literature on text classification methods. Lian implemented the RNN model, performed exploratory data analysis and error analysis, and wrote up this paper.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Compan-*

ion, pages 759–760. International World Wide Web Conferences Steering Committee, 2017.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *arXiv preprint arXiv:1903.04561*, 2019.

Michael Castelle. The linguistic ideologies of deep abusive language classification. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 160–170, 2018.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, 2019.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of ACL*, 2013.

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

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73. ACM, 2018.

M. Duggan. Online harassment. 2014.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4): 215–230, 2015.

H. Hosseini, S. Kannan, and B Zhang. Deceiving google's perspective api built for detecting toxic comments. arXiv preprint arXiv:1702.08138, 2017.

Stephen C Levinson. Pragmatics. cambridge textbooks in linguistics. *Cambridge/New York*, 1983.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26. Association for Computational Linguistics, 2012.

Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people. *Predictive Features for Hate Speech Detection on Twitter.. In HLT-NAACL*, 2016.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee, 2017.