Quantifying Gender Bias in Different Corpora

Marzieh Babaeianjelodar
Stephen Lorenz
Josh Gordon
Jeanna Matthews
babaeim,lorenzsj,jogordo,jnm@clarkson.edu
Clarkson University

Evan Freitag
EGF Statistical Consulting
evangfreitag@gmail.edu

ABSTRACT

Word embedding models have been shown to be effective in performing a wide variety of Natural Language Processing (NLP) tasks such as identifying audiences for web advertisements, parsing resumés to select promising job candidates, and translating documents from one language to another. However, it has been demonstrated that NLP systems learn gender bias from the corpora of documents on which they are trained. It is increasingly common for pre-trained models to be used as a starting point for building applications in a wide range of areas including critical decision making applications. It is also very easy to use a pre-trained model as the basis for a new application without careful consideration of the original nature of the training set. In this paper, we quantify the degree to which gender bias differs with the corpora used for training. We look especially at the impact of starting with a pre-trained model and fine-tuning with additional data. Specifically, we calculate a measure of direct gender bias on several pre-trained models including BERT's Wikipedia and Book corpus models as well as on several fine-tuned General Language Understanding Evaluation (GLUE) benchmarks. In addition, we evaluate the bias from several more extreme corpora including the Jigsaw identity toxic dataset that includes toxic speech biased against race, gender, religion, and disability and the RtGender dataset that includes speech specifically labelled by gender. Our results reveal that the direct gender bias of the Jigsaw toxic identity dataset is surprisingly close to that of the base pre-trained Google model, but the RtGender dataset has significantly higher direct gender bias than the base model. When the bias learned by an NLP system can vary significantly with the corpora used for training, it becomes important to consider and report these details, especially for use in critical decision-making applications.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Neural networks**.

KEYWORDS

gender bias, natural language processing, BERT, datasets

ACM Reference Format:
Marzieh Babaeianjelodar, Stephen Lorenz, Josh Gordon, Jeanna Matthews, and Evan Freitag. 2020. Quantifying Gender Bias in Different Corpora. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3366424.3383559

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3383559

1 INTRODUCTION

Machine learning models trained on big data are increasingly used in making important decisions about the lives of individuals in regulated areas such as hiring, housing and credit. The potential for these systems to learn gender bias from data fed to them in training has been demonstrated. For example, Amazon scraped an internally-developed recruiting engine when it downgraded graduates of women's colleges and resumes containing phrases such as "women's chess club captain" [2]. In the influential paper "Man is to computer programmer as woman is to homemaker? Debiasing Word Embeddings", Bolukbasi et al. demonstrated that word embedding models trained on a corpus of Google news associated men with the profession computer programmer and women with the profession homemaker [1]. This would clearly be problematic if used in parsing resumés of potential candidates for a computer programming job.

Despite these concerns, models pre-trained on "representative text" are regularly used and reused in a wide range of applications without a real appreciation of the nature of the actual training set. When building a system, developers look for ingredients that can be used to reduce development burden. The developers of the original system may recognize limitations based on their design, training data, or test coverage, but an appreciation of these limitations can easily be lost when a system is reused in a new and unanticipated context.

In this paper, we quantify the degree to which gender bias varies based on the corpora used as training set. We advocate for metrics of gender bias to be computed and reported, especially for pre-trained models that are frequently reused in new environments.

Specifically, we measure and analyze the gender bias in word embeddings using Bidirectional Encoder Representation of Transformers (BERT) and the direct gender bias calculation proposed by Bolukbasi et al. in their paper "Man is to computer programmer as woman is to homemaker? Debiasing Word Embeddings". We calculate a measure of direct gender bias on several pre-trained models including BERT's Wikipedia and Book corpus models as well as on several fine-tuned General Language Understanding Evaluation (GLUE) benchmarks. In addition, we evaluate the bias from several more extreme corpora including the Jigsaw identity toxic dataset that includes toxic speech biased against race, gender, religion, and disability and the RtGender dataset that includes speech specifically labelled by gender. Our results reveal that the direct gender bias of the Jigsaw toxic identity dataset is very close

to the direct gender bias of the base pre-trained Google model, but the RtGender dataset has significantly higher direct gender bias than the base model. When the bias learned by an NLP system can vary significantly with the corpora used for training, it becomes important to consider and report these details, especially in critical decision-making applications.

This paper is organized as follows: in Section 2, we explain the background by introducing word embeddings, BERT, GLUE, the Jigsaw toxic dataset obtained from Kaggle and the RtGender dataset obtained from Stanford University. In Section 3, we describe in detail how we trained the fine-tuned models we used, as well as how we extracted the vectors and calculated a direct measure of gender bias. In Section 4, we provide an evaluation of our results after fine-tuning on different datasets. We also describe the impact of data set size on accuracy and runtime. We compare our accuracy results to previously published work and discuss the importance of considering both accuracy and bias when using NLP systems.

2 BACKGROUND

In this paper, we use BERT (Bidirectional Encoder Representation from Transformers) [3] to calculate a measure of direct gender bias on word embeddings produced from a variety of training corpora. We follow the methodology proposed in Bolukbasi et al.'s [1] "Man is to computer programmer as woman is to homemaker? Debiasing Word Embeddings" to compute this gender bias metric. In this section, we provide some background on word embedding in general, Word2Vec (the word embedding model used in Bolukbasi et al.), and BERT (the word embedding model we used in this paper). We also described the pre-trained model we used and the various datasests we used for fine-tuning.

2.1 Word2Vec and BERT

Word embeddings are a representation of words in a vector space, where position is based on how they are used in combination with other words. Each word w is represented as a word vector $\vec{w} \in \mathbb{R}^d$ and distances between these vectors encode relationships between words. Trained models can be either context-free or contextual. In the context-free (or context-independent) models, each word has a single unique vector that does not vary with the other words found around it in a document. However, the same word can have very different meanings in different contexts (e.g. "my grandfather" and "grandfather clause"). Contextual models are able to represent words with multiple word vectors depending on the context of surrounding words.

Word embeddings enable a wide variety of queries. In addition to predicting the next word in a phrase or sentence, they can answer word analogy questions such as "the word man is to the word woman as the word king is to what word? Answer: queen". Word embeddings have been used in a wide variety of natural language processing applications from web search to resumé parsing [7].

Word2Vec, a word embedding system developed by Mikolov et al. in 2013 [6], is a shallow neural network with one hidden layer which takes text from the input layer and produces vector representations of the words as output. It is context-free and unidirectional.

Bolukbasi et al. [1] used Word2Vec to compute a measure of gender bias in a corpora of input documents. To do so they defined a set of gendered word pairs such as ('he","she") and used the difference between these word pairs as a measure of gender bias. We discuss this more in Section 3. The Word pairs they used for defining gender are available from (https://github.com/tolga-b/debiaswe/blob/master/data/definitional_pairs.json) [1].

BERT is based on a deeper multi-layered neural network than Word2Vec and is both contextual and bi-directional. It is therefore able to produce a richer model than Word2Vec.

2.2 Datasets

To draw conclusions and extract relationships using word embeddings, we must first train with a corpus of text. Large datasets provide more opportunity for learning, but are also computationally expensive to process. In addition, to support specific NLP tasks, it is often necessary to label the dataset (supervised training). In many cases, this must be done manually. For example, to train a classifier for distinguishing grammatically correct text from incorrect text, a person might need to manually label training samples as grammatically correct or incorrect. Labeling corpora can be time consuming and can require domain specific expertise such as professional linguistic knowledge.

As a result, it is increasingly common for researchers and developers to start with pre-trained models developed by others. One can begin with a pre-trained model and then perform fine-tuning with different datasets for specific task types. This is called transfer learning and is also used for ML tasks in different domains beyond natural language processing including computer vision. This enables researchers to deploy NLP-based systems much more quickly. However, pre-trained models can also be a source of unrecognized and unmeasured bias. Those assembling the pre-trained model may have a sense of its limitations that is under-appreciated by those reusing the models in new contexts.

In our experiments, we begin with a pre-trained BERT model that was trained using Wikipedia and a Book corpus. We fine-tune this model using a variety of datasets including the General Language Understanding Evaluation (GLUE) benchmarks and several more extreme corpora. We use the Jigsaw identity toxic dataset from Kaggle that includes toxic speech biased against race, gender, religion, and disability and the RtGender dataset from Stanford that includes speech labelled by gender.

The GLUE benchmark contains multiple tasks to evaluate the performance of Natural Language Understanding (NLU) models and covers a varied range of dataset sizes, and text genres [12]. These tasks include question answering, sentiment analysis, and textual entailment. BERT has been shown to offer high performance and high average accuracy on these GLUE tasks [3].

The GLUE datasets have been labeled to support specific tasks. In a two-class model, a binary labeling scheme is used (yes/no, 0/1). In a multi-class labeling scheme, labels take on three or more domain specific possibilities. All of the GLUE tasks use binary labeling except for the Multi-Genre Natural Language Inference (MNLI) corpus. MNLI used three classes.

Each GLUE task, including MNLI, is described in more detail below and summarized in Table 1. The Jigsaw Toxicity and RtGender

Table 1: GLUE Datasets

Corpus	Task	Type	Domain
WNLI	Natural Language Inference	Two-class	Fiction Books
CoLA	English Acceptability	Two-class	Books and journal articles on linguistic theory
SST-2	Sentiment Prediction	Two-class	Movie reviews and human annotations of their sentiment
MRPC	Paraphrase	Two-class	Online news sources
QQP	Paraphrase	Two-class	Social QA questions on Quora
MNLI	Natural Language Inference	Three-class	Transcribed speech, fiction, and government reports
RTE	Natural Language Inference	Two-class	Online news sources
QNLI	Natural Language Inference	Two-class	Wikipedia

Table 2: Jigsaw Toxicity and RtGender Datasets

Corpus	Task	Type	Domain
Toxic	Identity based Biases	Two-class	Wikipedia
RtGender	Gender based Biases	Two-class	Facebook, TED, Fitocracy, and Reddit

datasets are also described in more detail below and summarized in Table 2.

- **CoLA**
  The Corpus of Linguistic Acceptability (CoLA) [13] is a single-sentence task that uses 22 books and journal articles on linguistic theory. Text is labeled as grammatically correct or incorrect.

- **SST-2**
  SST-2 is the Stanford Sentiment Treebank [10] consisting of movie reviews with human sentiment annotations. SST-2 is a binary classification task which can predict the sentiment of a given sentence using binary positive/negative sentiment labels.

- **MRPC**
  The Microsoft Research Paraphrase Corpus (MRPC) contains text selected from online news sources. Sentence pairs labeled whether they are semantically equivalent or not.

- **QQP**
  The Quora Question Pairs (QQP) dataset consists of a set of question pairs obtained from the Quora website. Pair of questions are labeled as semantically equivalent or not.

- **WNLI**
  The Winograd Schema Challenge (WNLI) [5] is a reading comprehension task with text extracted from fiction books. In this task, the system must identify the entity to which a pronoun refers. As an example [4], looking at the sentence "The city councilmen refused the demonstrators a permit because they [feared/advocated] violence". If "feared" is chosen then it could be interpreted that "they" refers to the city council on the other hand if "advocated" is chosen then "they" could be referred to "demonstrators". The binary classification is to choose whether the first or second word is correct, given a version of the sentence.

- **MNLI**
  The Multi-Genre Natural Language Inference (MNLI) corpus uses crowd-sourced sentence pairs for a variety of sources (e.g. transcribed speech, fiction and government reports). Within the text, premise sentences and hypothesis sentences are identified. These sentences are then labeled with whether the premise affirms the hypothesis, contradicts it, or if it is neutral. Unlike the others, this is a multi-class (three-class) classification task.

- **RTE**
  Recognizing Textual Entailment (RTE) is a binary dataset that looks for textual entailment and is collected from news and Wikipedia text. In textual entailment, two text fragments are identified and then the pair is classified by whether the truth of one text fragment follows from the other. The task is to determine whether or not a hypothesis follows from a given sentence.

- **QNLI**
  The Stanford Question Answering dataset (QNLI) [9] contains question-paragraph pairs extracted from Wikipedia. The task is to classify whether the answer to the question is found in the sentences of the paragraph or not.

- **Toxic**
  The Toxic corpus obtained from Kaggle[1] contains a set of negative online behaviors, such as toxic comments that are biased against race, religion, disability, gender. This corpus is not a part of the GLUE benchmark. The task in this dataset is to tell if a given piece of text is toxic or not.

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

- **RtGender**
  RtGender [11] is a labeled multi-genre dataset that studies responses to online speech when the gender of the speaker is known. Specifically, online posts are labeled with the gender of the author and comments/responses to that post are collected. The binary classification task is for the system to predict the gender of the author from the comments. This dataset was developed by Stanford University and allows the study of gender bias in language and social media [11]. RtGender consists of 25M comments from different sources as shown in Table 2. The RtGender dataset includes responses to Facebook posts from U.S. House, Senate, public figures such as TV hosts and athletes. It also includes responses to TED speakers, fitness progress posts from Fitocracy, and Reddit comments.

3 METHODOLOGY

In this section, we describe our methodology for using BERT and fine-tuning on different datasets. We also describe how we compute the direct bias metric of gender bias.

3.1 BERT's Pre-trained Model

In order to train an effective model, one must have a large corpus consisting of millions or billions of labeled data points. BERT offers the ability to use pre-trained models that can be fine-tuned with additional smaller corpora. The BERT Github page from Google (https://github.com/google-research/bert#bert) offers a variety of BERT models pre-trained on large datasets. The pre-trained models use the supervised training method which means it is trained on labeled data. The pre-trained models vary in layer size (L), hidden layer size (H),the number of self-attention heads (A), and being cased or uncased (i.e. whether or not they pay attention to capital letters when training). We chose to use the uncased_L-12_H-768_A-12 model trained on Wikipedia with 2500 M and BookCorpus 800 M words respectively as our baseline model. We note that Bolukbasi et al. also used an uncased model, but one using Word2Vec and trained on a corpus of Google News [1].

3.2 Fine-Tuning

Beginning with a pre-trained model for BERT, we then perform a fine-tuning step. Specifically, we fine-tuned BERT with each of the datasets we described in Section 2. For each of these datasets, we produce a different fine-tuned model from which we can extract a set of resulting word vectors to be used in calculating direct gender bias.

We perform this fine-tuning on an Nvidia GeForce GTX 1070 graphics card using the TensorFlow library. We use a batch size of 16 for 3 epochs similar to the process followed in Devlin et al. [3]. We ran this process 20 times for each dataset. There is a degree of non-determinism in each run due to the random initialization of weights in the models and drop out in the internal nodes of the neural network.

3.3 Computing Direct Gender Bias

In this section, we describe how we compute a direct bias metric of gender bias. As in Bolukbasi et al., we compute a gender direction $g \in \mathbb{R}^d$ based on a definitional set of 10 gendered word pairs [1]. Specifically, we use ten pairs of words (she-he, her-his, woman-man, mary-john, herself-himself, daughter-son, mother-father, gal-guy, girl-boy, female-male). These gendered word pairs are used to define a gender subspace.

We then evaluate the position of a set of 320 profession words (such as accountant, doctor, inventor, etc.) relative to the gender subspace that we calculated. These words are intended to be words that should not, but might in practice, imply a specific gender. The full list of profession words is given in Appendix A.

We use the same set of profession words as in Bolukbasi et al. [1]. The list was generated by asking crowd workers on Amazon Mechanical Turk to propose profession/occupation words that might reflect gender stereotypes. The intuition is that to avoid gender-based discrimination in hiring, these profession words should be gender neutral (e.g. words like mechanic or doctor should not be gendered). However, these words may reflect gender stereotypes in the way they are used in a corpus of human language. If so, an NLP system trained on that corpus of human language could learn these gender stereotypes. This would be a problem if we for example used that NLP system as the basis for hiring decisions.

To extract the vector(s) for each desired word (the definitional set of 10 gendered word pairs and the list of 320 profession words), we use the contents of the last four hidden layers in BERT's neural network model, denoted by -1, -2, -3, -4. We obtain a single vector representation by summing these four hidden layers into a single vector. In the event that a word is separated into word pieces (e.g. adjunct professor), each piece is treated as a token vector and then each token vector is further summed with the other token vectors to produce a single word vector.

We first calculate the center of each definitional pair. For example, to calculate the center of the pair she/he, we average the vector for "she" with the vector for "he". Then, we calculate the distance of each word in the definitional pair from the center by subtracting the center from each word in the pair (e.g. "she" - center). Finally, we apply Principal Component Analysis (PCA) to the matrix of these distances. PCA is an approach that compresses multiple dimensions in to fewer dimensions, in such a way that the information within the original data is not lost. Usually the number of reduced dimensions is 1-3 as it allows for easier visualization of a dataset. We call the first eigenvalue from the PCA matrix (larger than the rest), the g direction.

We use the following formula for direct gender bias from Bolukbasi et al. [1]:

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(w, g)|^c \tag{1}$$

where N represents the list of profession words obtained from Bolukbasi et al. [1], g represents the gender direction calculated, w represents each job title word and c is a parameter to measure the strictness of the bias. The profession words which are the words that should be gender neutral are defined as $N \subset W$ such as flight-attendant, which should not be gender specific. N is equal to the number of profession words, in our case 320. To calculate the cosine

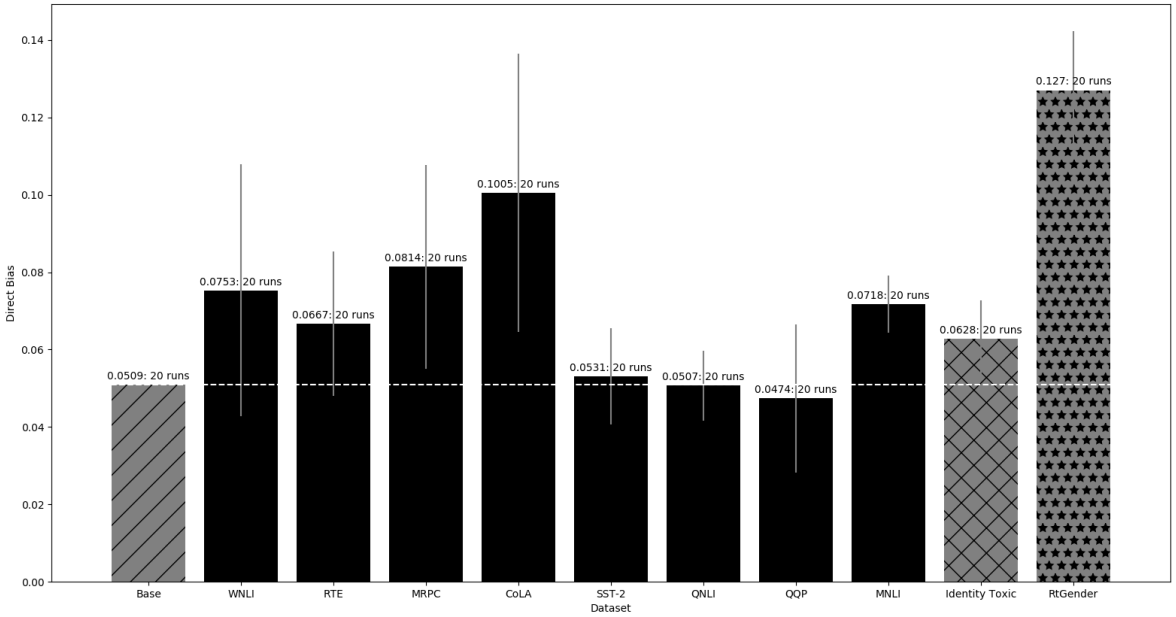

Figure 1: Direct Gender Bias for datasets trained on BERT. Black bars are GLUE datasets.

Table 3: Additional Metrics for the Pre-Trained BERT Model and Fine-Tuned Models

Trained Models	Direct Bias	Accuracy	Size (N) Lines	Size Bytes	Runtimes (m)
Pre-trained BERT Model	0.0509	—	—	—	—
WNLI	0.0753	52	636	152Kb	1h3m
RTE	0.0667	68.8	2491	1.8Mb	2h49m
MRPC	0.0802	86.1	3669	2.9Mb	3h55m
CoLA	0.0963	81.6	8551	1.5Mb	8h33m
SST-2	0.0572	92.5	67350	24Mb	64h23m
QNLI	0.0519	91.5	104744	28Mb	19h8m
QQP	0.0629	90.8	363871	160Mb	310h41m
MNLI	0.0718	83.8	392703	1.4Gb	373h34m
Jigsaw Toxicity	0.0628	95.3	3377	1.6Mb	3h40m
RtGender	0.1271	81	10000	49Mb	24h22m

Table 4: Comparison of Accuracy for BERT on GLUE Benchmarks

	RTE	MRPC	CoLA	SST-2	QNLI	QQP	MNLI
Original BERT paper	66.4	88.9	52.1	93.5	90.5	71.2	84.6/83.6
Our results	68.8	86.1	81.6	92.5	91.5	90.8	83.8

of vectors u and v, we use the following formula:

$$\cos(u, v) = \frac{u \cdot v}{||u|| \times ||v||} \tag{2}$$

In this paper, we chose to only explore setting $c = 1$, however c values and their effects are explained in more detail in Bolukbasi et al. [1].

4 EVALUATION

Figure 1 presents our findings about how gender bias varies with the training data set used. The first bar, labeled Base, reflects the direct gender bias calculation from the pre-trained BERT model with no fine-tuning. Each of the 10 subsequent bars reflect the direct gender bias calculated from a fine-tuned model which started with the pre-trained BERT model and then added additional training with the specified corpus. We include a horizontal line for the direct

gender bias measurement for the pre-trained model. Bars for the 8 GLUE benchmark datasets are solid. Bars for the Base BERT model as well as the Identity Toxic and RTGender datasets are patterned. The GLUE benchmarks bars are ordered by the size of the dataset.

We see that fine-tuning with 8 of the 10 datasets result in a higher direct gender bias metric than the base BERT model. Not surprisingly, RtGender, the corpus that specifically focuses on gender-based language results in the largest increase in the direct gender bias metric. Interestingly, the Identity Toxic data set produced a smaller increase in gender bias than many of the GLUE datasets even though it is specifically described as containing toxic gender comments. With a manual look through this dataset (not a fun experience because of the offensive nature of the comments!), we found that the majority of the toxic comments in the dataset were related to racial identity or sexuality rather than gender.

This really underscores the importance of measuring and reporting the gender bias resulting from different data sets. Qualitative descriptions of a dataset can lead to misleading assumptions. Datasets that seem innocuous can actually lead to more gender bias in the resulting model than a dataset specifically described as containing toxic genderized speech. We highly recommend that bias metrics be calculated and reported for training sets. For pre-trained models designed to be re-used in new, unanticipated contexts, this is critical, especially if a model is going to be re-used in high-impact/regulated areas such as hiring, housing or credit.

Table 3 provides some additional context. In addition to the direct bias results shown in Figure 1, it reports accuracy, size and total runtime for all 20 runs of the fine-tuned models. Accuracy refers to success on the specific NLP task for which the fine-tuning corpus was designed. We report the size of the datasets in two ways. Size in lines reflects the number of lines of text in the training example file used for fine-tuning, whereas size in bytes is the number of bytes in the corpus itself. The runtimes reported are for 20 runs (not a single run) of fine-tuning executed on a Nvidia GeForce GTX 1070 graphics card using the TensorFlow library. Twenty runs on the different data sets vary from just over 1 hour for 152 Kb WNLI to almost 16 days for 1.4 GB MNLI. This does not include the much longer time that would be required to produce the pre-trained model itself and helps put in perspective why many researchers and developers start with pre-trained models. This trend toward transfer learning makes it increasingly important that we calculate and report various measures of bias such as this direct gender bias metric for the pre-trained models used widely in unanticipated contexts by groups who did not assemble the training set themselves.

Table 3 reports the accuracy on the specific NLP task for each dataset. For example, how in the case of SST-2 how accurately the resulting model can classify the sentiment of a given sentence as positive or negative. The average accuracy for most tasks in our experiments are between 80 and 95%. However, there are some interesting outliers. For example, WNLI has an accuracy of 52% which is not significantly better than randomness. This is consistent with the findings in the original BERT paper where Devlin et al. state that WNLI is problematic and even exclude it when calculating the average GLUE score in their paper [3]. RTE also has a relatively low accuracy of 68% and that is also consistent with the findings of the original BERT paper.

We do not necessarily expect high accuracy on all of these tasks. Nangia and Bowman compare the performance of the GLUE tasks with humans and show that although GLUE tasks have improved recently, there is a gap between human and machine performance in Nangia and Bowman (2019) [8]. Levesque et al. showed that in the WNLI task humans perform better than machines by more than 30 percent [5]. Accuracy on a particular task is not necessarily an indicator of quality of the word embedding used. The fact that we are seeing high accuracy on a wide range of meaningful NLP tasks does suggest the usefulness of the word embedding. It is however important to consider both bias and accuracy in any given task. Even if using gender would increase the accuracy of a prediction task overall, discrimination on the basis of protected attributes is still illegal. Individual candidates for a job must be considered on their individual merits.

Table 3 also emphasizes that as in real applications the datasets used for fine-tuning varied substantially in their size. The fine-tuning process contains some non-determinism due to random weight initialization and dropout, and due to this, smaller datasets can lead to more inconsistencies. We note that the standard deviation of the direct gender bias and accuracy both generally shrink with increased size of the training set. Devlin et al. observed a similar trend in the original BERT paper where they state that they ran the smaller datasets several times and chose the most accurate of the resulting models [3].

Table 4 contains a more detailed comparison of accuracy as reported in the original BERT paper [3] and our own experiments. It was a nice opportunity to perform some independent testing of published results, something that regrettably isn't done as often in computer science as in other sciences. In most cases our results show similar or higher accuracy for the GLUE datasets. We in fact observed dramatically higher accuracy for CoLA and QQP, adding additional evidence to BERT's claims of both good performance and high accuracy. We hypothesize that improvements in BERT since the publishing of the original paper in 2018 may be responsible for these increases in accuracy. Note: the original BERT paper reports two accuracy numbers for MNLI: both the match and mismatch accuracy. We are reporting only the match accuracy. WNLI is not listed in Table 4 because Devlin et al. excluded it when calculating the average GLUE score in their paper.

5 CONCLUSION

In this paper, we have demonstrated the degree to which the gender bias learned by NLP systems can vary with the training corpus. We quantified the gender bias from a wide variety of corpora including a pre-trained model used widely by others and fine-tuned models with a variety of corpora including some extreme datasets with toxic speech and gender-specific comments. We have extended the work in Bolukbasi et al. to both use BERT, a bi-directional contextual word embedding system, rather than the uni-directional, context-free Word2Vec and also to compare the results across multiple training corpora rather a single Google News corpus. Our results reveal that the direct gender bias of seemingly innocuous datasets can be even higher than a data set specifically described as containing toxic comments on race, religion, disability, and gender. We discuss the current trend toward transfer learning and how that makes it even more important to quantify and report metrics of bias in a training

set. When the bias learned by an NLP system can vary significantly with the corpora used for training and especially when pre-trained models are used in unanticipated contexts by groups unaware of limitations in the model, we should be computing and reporting metrics of bias learned from the training set especially for critical decision-making applications in areas such as hiring, housing and credit.

6 FUTURE WORK

We would like to directly compare the impact of Word2Vec vs. BERT on the exact same corpus. We would like to experiment with some changes in 10 word pairs used to define the gender subspace and the 320 profession words used. We are also interested in extending this work to debias word embeddings across these corpora and to compare the direct bias metrics before debiasing to those after debiasing.

7 ACKNOWLEDGMENTS

Thanks to Adam Tauman Kalai at Microsoft Research New England (Cambridge, MA) who helped us with the idea of investigating BERT. We thank our wider research group at Clarkson as well as the Clarkson CS 649 class in the Fall 2019 for their helpful comments and discussion, especially Hunter Bashaw and Izzi Grasso. We couldn't have done this work without help from the Clarkson Open Source Institute.

REFERENCES

[1] Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems* (2016), pp. 4349–4357.

[2] Dastin, J. Amazon scraps secret ai recruiting tool that showed bias against women. reuters business news.

[3] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Ernest Davis, L. M., and Ortiz, C. The winograd schema challenge.

[5] Levesque, H. J., Davis, E., and Morgenstern, L. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning* (2012), KR'12, AAAI Press, pp. 552–561.

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (2013), pp. 3111–3119.

[7] Nalisnick, E., Mitra, B., Craswell, N., and Caruana, R. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web* (2016), International World Wide Web Conferences Steering Committee, pp. 83–84.

[8] Nangia, N., and Bowman, S. R. Human vs. muppet: A conservative estimate of human performance on the glue benchmark. *arXiv preprint arXiv:1905.10425* (2019).

[9] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[10] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (2013), pp. 1631–1642.

[11] Voigt, R., Jurgens, D., Prabhakaran, V., Jurafsky, D., and Tsvetkov, Y. Rtgender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)* (2018).

[12] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[13] Warstadt, A., and Bowman, S. R. Grammatical analysis of pretrained sentence encoders with acceptability judgments. *arXiv preprint arXiv:1901.03438* (2019).

APPENDIX A

The set of 320 profession/occupation words used for measuring bias are listed below. We use the same set as Bolukbasi et al. in their "Man is to computer programmer as woman is to homemaker? Debiasing Word Embeddings" paper [1]. We used the list from https://github.com/tolga-b/debiaswe/blob/master/data/professions.json. The intention for this list is to represent words that should be gender neutral, specifically to avoid gender-based discrimination in hiring, profession words should be gender neutral. The list below is indeed dominated by gender neutral profession words, but we also note a few words that we would not have considered gender neutral including actress, waitress, ballerina, dad, nun and housewife. There are also some words that might not be considered professions/occupations but are gender neutral including acquaintance, drug addict, teenager and alter ego. We also note some words like confesses and soft spoken that while gender neutral seem fundamentally different than the other words in the list. We decided not to alter the list in this paper, but plan to experiment with a slightly modified list in the future. We also note that Bolukbasi et al. report using at list of 327 profession/occupation words, however the list below from Github has 320 words.

actor, accountant, acquaintance, actor, actress, adjunct_professor, administrator, adventurer, advocate, aide, alderman, alter_ego, ambassador, analyst, anthropologist, archaeologist, archbishop, architect, artist, artiste, assassin, assistant_professor, associate_dean, associate_professor, astronaut, astronomer, athlete, athletic_director, attorney, author, baker, ballerina, ballplayer, banker, barber, baron, barrister, bartender, biologist, bishop, bodyguard, bookkeeper, boss, boxer, broadcaster, broker, bureaucrat, businessman, businesswoman, butcher, butler, cab_driver, cabbie, cameraman, campaigner, captain, cardiologist, caretaker, carpenter, cartoonist, cellist, chancellor, chaplain, character, chef, chemist, choreographer, cinematographer, citizen, civil_servant, cleric, clerk, coach, collector, colonel, columnist, comedian, comic, commander, commentator, commissioner, composer, conductor, confesses, congressman, constable, consultant, cop, correspondent, councilman, councilor, counselor, critic, crooner, crusader, curator, custodian, dad, dancer, dean, dentist, deputy, dermatologist, detective, diplomat, director, disc_jockey, doctor, doctoral_student, drug_addict, drummer, economics_professor, economist, editor, educator, electrician, employee, entertainer, entrepreneur, environmentalist, envoy, epidemiologist, evangelist, farmer, fashion_designer, fighter_pilot, filmmaker, financier, firebrand, firefighter, fireman, fisherman, footballer, foreman, freelance_writer, gangster, gardener, geologist, goalkeeper, graphic_designer, guidance_counselor, guitarist, hairdresser, handyman, headmaster, historian, hitman, homemaker, hooker, housekeeper, housewife, illustrator, industrialist, infielder, inspector, instructor, interior_designer, inventor, investigator, investment_banker, janitor, jeweler, journalist, judge, jurist, laborer, landlord, lawmaker, lawyer, lecturer, legislator, librarian, lieutenant, lifeguard, lyricist, maestro, magician, magistrate, maid, major_leaguer, manager, marksman, marshal, mathematician, mechanic, mediator, medic, midfielder, minister, missionary, mobster, monk, musician, nanny, narrator, naturalist, negotiator, neurologist, neurosurgeon, novelist,

nun, nurse, observer, officer, organist, painter, paralegal, parishioner, parliamentarian, pastor, pathologist, patrolman, pediatrician, performer, pharmacist, philanthropist, philosopher, photographer, photojournalist, physician, physicist, pianist, planner, plastic_surgeon, playwright, plumber, poet, policeman, politician, pollster, preacher, president, priest, principal, prisoner, professor, professor_emeritus, programmer, promoter, proprietor, prosecutor, protagonist, protege, protester, provost, psychiatrist, psychologist, publicist, pundit, rabbi, radiologist, ranger, realtor, receptionist, registered_nurse, researcher, restaurateur, sailor, saint, salesman, saxophonist, scholar, scientist, screenwriter, sculptor, secretary, senator, sergeant, servant, serviceman, sheriff_deputy, shopkeeper, singer, singer_songwriter, skipper, socialite, sociologist, soft_spoken, soldier, solicitor, solicitor_general, soloist, sportsman, sportswriter, statesman, steward, stockbroker, strategist, student, stylist, substitute, superintendent, surgeon, surveyor, swimmer, taxi_driver, teacher, technician, teenager, therapist, trader, treasurer, trooper, trucker, trumpeter, tutor, tycoon, undersecretary, understudy, valedictorian, vice_chancellor, violinist, vocalist, waiter, waitress, warden, warrior, welder, worker, wrestler, writer