Identifying and Reducing Gender Bias in Word-Level Language Models

Shikha Bordia[1]
sb6416@nyu.edu
[1]Courant Institute of Mathematical Sciences, New York University, 10 Washington Place, New York, NY 10003

Jason Cramer[2]
jtc440@nyu.edu
[2]Dept. of Electrical and Computer Engineering, New York University, 2 MetroTech Center, Brooklyn, NY 11201

Yu Wang[3]
wangyu@nyu.edu
[3]Music and Audio Research Laboratory, New York University, 35 W 4th Street, New York, NY 10012

Abstract

Many corpora exhibit problematic bias, which can be propagated or amplified by data-driven models trained on this data. Limiting the scope to gender bias, we use a news dataset and evaluate its gender bias by comparing co-occurrences with gendered words in a context window between genders. We then train word-level prediction models implemented with recurrent neural networks using an embedding encoder and measure the bias exhibited by the generated sentences. To reduce the bias, we propose a regularization term used in training the language model, which enforces the embeddings learned by the encoder project minimally onto an embedding subspace encoding gender. Finally, we measure the efficacy of our method in reducing gender bias. We find that the word-level language model does amplify bias, but surprisingly we find that bias regularization increases the bias score. However, we do find that word embeddings do move away from the gender subspace.

1 Introduction

Dealing with discriminatory bias in training data is a major issue concerning the mainstream implementation of machine learning. The bias in the data can be amplified by models and the resulting output consumed by the public can influence them, encourage and reinforce harmful stereotypes, or distort the truth. Automated systems that depend on these models can take problematic actions based on biased profiling of individuals.

A common example of bias is gender bias, which manifests itself in such ways as profiling the professional predilections and capabilities of a job applicant based on their gender as part of an automated recruitment resume screening process. Because of this growing concern, evaluation and mitigation of the bias in the data and the algorithms that use it has been a growing field of research in recent years.

One natural language understanding task vulnerable to gender bias is language modeling. The task of language modeling has a number of practical applications, such as word prediction used in text messaging applications. Predictions generated by these language models may contain gender bias that can have effects ranging to accidentally sending an insensitive text message to internally reinforcing discriminatory beliefs based on gender in a user. If possible, we would like to identify the bias in the data used to train these models and reduce its effect on model behavior.

Towards this pursuit, we aim to evaluate the effect of gender bias in word-level language models that are trained on text containing biased phrases. Our contributions in this work include 1. an analysis of the gender bias exhibited by a publicly available dataset, 2. an analysis of the effect of this bias on recurrent neural networks (RNNs) based word-level language models, 3. a method for reducing bias learned in these models, and 4. an analysis of the results of our method.

2 Related Work

A number of methods have been proposed for evaluating and addressing bias existing in datasets and the models that use them.

Recasens et al. studied the neutrality point of view (NPOV) edit tags in the Wikipedia history edits to understand linguistic realization of bias (Recasens et al., 2013). According to their study, bias can be broadly categorized into two classes, framing and epistemological. While the framing bias is more explicit, the epistemological bias is implicit and subtle. Framing bias occurs when subjective or one-sided words are used. Epistemological are entailed, asserted or hedged in the text.

It may be possible to capture both of these kinds of biases through the distributions of co-occurrences.

Srivastava et al. highlighted the concerns with bias that can be exhibited by consumer services driven by APIs implementing AI technologies (Srivastava and Rossi, 2018). They design an experimental framework for evaluating the bias of a service (or a composition of systems), in which they characterize the service's response to biased or unbiased input. In their case study, they evaluate gender bias in a machine translation setting, where an English sentence is translated to a series of intermediate languages before being translated to English.

Bolukbasi et al. in (Bolukbasi et al., 2016) investigate the gender bias present in popular word embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). They find that many words, like descriptors and professions, exhibit severe gender bias. They propose a method in which they construct a gender subspace using a set of binary gender pairs and remove the component of the word embeddings that project onto this subspace. A softer variation is also proposed that balances reconstruction of the original embeddings while minimizing the part of the embeddings that project onto the gender subspace.

Zhao et al. look at gender bias in the context of using structured prediction for visual object classification and semantic role labeling. They observe a gender bias in the training examples and their model amplifies it in the predictions. They measure the bias using the relative co-occurrence of the labels for each gender, and measure the amplification by seeing how the distributions change from the training set to the model output. They impose constraints on the optimization using Lagrangian relaxation to reduce bias amplification while incurring minimal degradation in their model's performance.

The particular task of language modeling has been given a lot of attention over the years. Simple N-gram Markov models have been a classic approach to word-level language modeling (Jurafsky and Martin, 2008). More recent approaches to language models use recurrent neural networks (RNNs) with specific optimization algorithms (Sutskever et al., 2011). RNNs are generally powerful for modeling sequences; their high-dimensional hidden state with nonlinear dynamics enable it to remember and process past information. Due to its unstable gradient, special optimization techniques such as gradient clipping or Hessian-Free optimization are required to train RNNs. Another approach is to use Long Short-Term Memory (LSTM) (Sundermeyer et al., 2012), which has a modified network architecture to avoid vanishing or exploding gradient problem.

3 Methods

We first examine the bias existing in the dataset through qualitative and quantitative analysis of trained embeddings and co-occurrence patterns. We then train a LSTM word-level language model on the Daily Mail dataset and measure the bias of the generated outputs. We then apply a regularization procedure, which encourage the encoder embeddings learned by the model to depend minimally on gender for words we expect to be gender neutral. Finally, we assess the efficacy of the proposed method in reducing bias. The code implementing our methods can be found in our GitHub repository[1].

3.1 Dataset

The data that we use comes from a dataset of articles from the Daily Mail newspaper.

3.1.1 Daily Mail Summarization Dataset

Daily Mail is considered to be a biased [2] and sensational [3] news source. This dataset was released as part of a summarization dataset (Hermann et al., 2015), and contains 219,506 articles. We subsample the sentences by a factor of 100 in order to make the dataset more manageable for experiments.

3.2 Text Preprocessing

For text preprocessing, we make use of `spaCy` (Honnibal et al., 2018). We use the sentence tokenizer and word tokenizer to split each document into sentences and each sentence into tokens. We then clean up the tokens by 1) transliterating the unicode characters to ASCII using `unidecode` [4], 2) converting to lowercase, 3) replacing digits with a placeholder string `<NUM>`, 4) removing special characters except for `<>$.-'`, and 5) removing the following leading or trailing characters: `.-'` .

[1] https://github.com/jtcramer/language-model-bias
[2] https://www.allsides.com/news-source/daily-mail
[3] https://arstechnica.com/information-technology/2017/02/wikipedia-bans-daily-mail
[4] https://pypi.org/project/Unidecode

At the end of each sentence, we append a <eos> tag to denote the end of the sentence. We also remove the highlights included for the summarization task since they are not full English sentences and are not relevant to our task.

3.3 Word-Level Language Model

For the language model, we use the multi-layer LSTM word-level language model implemented with PyTorch (Paszke et al., 2017). The model consists of an encoder word embedding layer, arbitrary number of LSTM layers, and a fully connected decoding layer. The model had originally been tested by training on the Wikitext-2 dataset (Merity et al., 2016). With the default hyperparameters: word embedding dimension of 200, 2-layers LSTM, 200 hidden units per layer, initial learning rate of 20, batch size of 20, sequence length of 35, 0.2 dropout, and trained on 6 epochs, the reported perplexity is 117.61.

3.4 Quantifying Biases

Machine learning techniques capture patterns in data to make coherent predictions, which can unintentionally capture bias as well. For simple data, this bias can be caused simply by class imbalance, which is relatively easy to quantify and fix. For text and image data, the complexity in the nature of data increases and it becomes difficult to quantify. Nonetheless, defining relevant metrics are crucial in assessing the bias exhibited in a dataset or in a model's behavior.

3.4.1 Bias Score

The bias for a given word, towards either gender, we define as:

$$b(o) = \frac{c(o, g_f) - c(o, g_m)}{c(o, g_f) + c(o, g_m)}$$

where $c(o, g)$ is the number of co-occurrences of o and g in the corpus

We count a co-occurrence with g if the word co-occurs with a word we associate with g. For example, for $g =$ "female" such words would include "she", "her", and "woman". We take the context of co-occurrences to be a window of M words. M is varied and chosen based on empirical trials that gives a better accuracy. Note that the bias score varies from -1 to 1, where negative values indicate a bias towards males and positive values indicate a bias towards females.

To quantify the bias on an entire corpus, we can compute the average bias scores over words in the corpus:

$$\bar{b} = \frac{1}{O_b} \sum_{o \in O_b} b(o)$$

where O_b is the set of tokens that co-occur with at least one gender word; that is:

$$O_b = \{o \in O | (c(o, g_f) > 0) \cup (c(o, g_m) > 0)\}$$

This is done since the bias score is only well-defined for words that co-occur with a gender word.

3.4.2 Bias Amplification

Let $\bar{b}`(o, g)$ correspond to bias computed on the generated output of the model and let $\bar{b}^*$ correspond to the average bias computed in the training dataset. Assuming that the test set is similarly distributed to the training set, if $\bar{b}^*$ is greater than $\bar{b}`$ we say that bias has been amplified.

The bias amplification is defined as:

$$\delta_b(o) = |b`(o) - b^*(o)|$$

We can define the bias amplification of one corpus with respect to another by averaging over all words:

$$\bar{\delta_b} = \frac{1}{O_b} \sum_{o \in O_b} \delta_b(o)$$

3.5 Model De-biasing

Taking a similar approach to that in (Bolukbasi et al., 2016), we consider a *gender subspace* present in the learned embedding matrix in our model. Let $\mathbf{w} \in S_W$ be a word embedding corresponding to a row in the word embedding matrix W. Let

$$D_i, \ldots, D_n \subset S_W$$

be the *defining sets* that contain difference between gender-opposing words, e.g. female and male. We consider the matrix C which is defined as a stack of difference vectors between the pairs in the defining sets. If $\{\mathbf{u}_i, \mathbf{v}_i\} = D_i$ then we have

$$C = \begin{bmatrix} (\frac{\mathbf{u}_1 - \mathbf{v}_1}{2})^\intercal \\ \vdots \\ (\frac{\mathbf{u}_n - \mathbf{v}_n}{2})^\intercal \end{bmatrix} = U \Sigma V^\intercal$$

The difference between the pairs captures components corresponding to gender difference, hopefully discarding irrelevant information like age (in a pair "girl" and "boy") or relationship (in a pair "wife" and "husband"). We then perform singular value decomposition on C, obtaining $U\Sigma V^\intercal$. The gender subspace B is then defined as the first k columns (where k is chosen to capture 50% of the variation) of the right singular matrix V. That is,

$$B = V_{:,1:k}$$

.

Let N be the matrix consisting of the embeddings for which we would like the corresponding words to exhibit unbiased behavior. If we want the embeddings in N to have minimal bias, then its projection onto the gender subspace B should be small in terms its the squared Frobenius norm. Therefore, to reduce the bias learned by the embedding layer in the model, we can add the following bias regularization term to the training loss:

$$\mathcal{L}_B = \lambda \|NB\|_F^2$$

where λ controls the importance of minimizing bias in the embedding matrix W (from which N and B are derived) relative to the other components of the model loss.

4 Experiments

4.1 Design

First, we measure the bias of the Daily Mail dataset using our bias measure. We then train our LSTM language models on the dataset with and without bias regularization, picking the model that achieves the best loss on a development set. The models that performed best used an embedding size (with tied weights to the decoder hidden layer) of 500, with a learning rate of 20. We train for up to 100 epochs, using early stopping with a patience of 5. After training our models, we generate 4000 example documents of 500 words each. We compute our bias metrics and compare them to the training corpus and see if our regularization term reduced the bias compared to both the standard model and the dataset.

4.2 Dataset Bias

The bias score we obtained for the Daily Mail dataset was -0.2163, which indicates that there is an overall bias towards males in the dataset.

4.3 Results

Regarding the language modeling task itself, our baseline model achieved a perplexity of 257 on the Daily Mail training set of vocabulary size of approximately 40,000 on our test set. First, we note that the language modeling task amplified bias in the dataset, as we see a considerable bias amplification when in the baseline model. As mentioned, this is to be expected in machine learning models trained on biased data.

It is interesting to note that bias regularization actually decreased perplexity. This is surprising, as decreasing the variation due to gender which one would expect the perplexity to increase. However, we do observe that as the perplexity increases, the bias difference increases. This unfortunately means that our proposed bias regularization method increases the bias score rather than decreasing it.

In fact, as we increase the bias regularization parameter λ, the perplexity decreases and the bias measure grows. We can see from Figure 1 and Figure 2 that not only does the bias regularization increase the average bias amplification, it actually shifts the mass of the distribution of bias scores to be more biased towards men. Additionally the distribution of bias amplification is skewed towards men. Counter to our expectations and intuitions, this method of bias regularization dramatically worsens the effect of gender bias in our model.

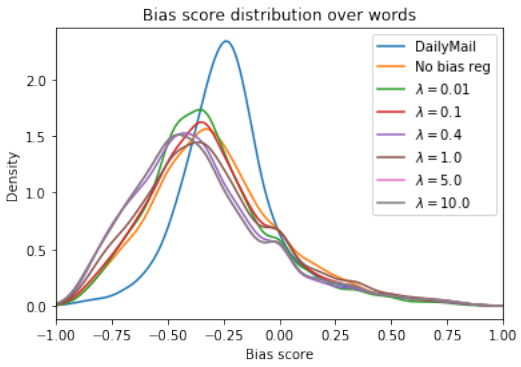

Figure 1: Distribution of bias scores across words in the dataset and generated text from the models.

A possible explanation to this phenomenon could be as follows. There is perplexity bias tradeoff of adding the bias regularization term. In order to achieve a better perplexity, there is compromise on bias. Intuitively, if we were to reduce bias, the perplexity is bound to increase. The fact

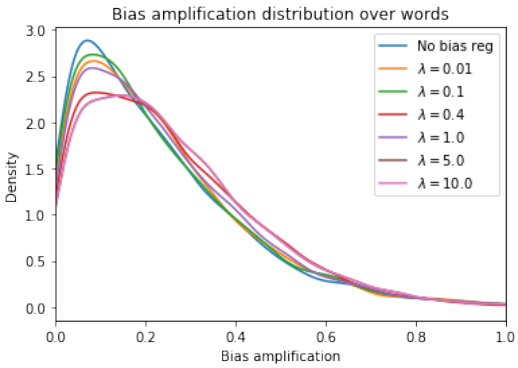

Figure 2: Distribution of bias amplification across words in generated text from the models. The distribution plot of bias amplification is highly positively skewed towards right. The skewness decreases as λ increases indicating a decrease in least amplified biased terms and an increase in more amplified bias terms.

λ	**Perplexity**	**Mean Bias Difference**	**Skew of Bias Difference**
0.0	257.00	0.2275	1.4665
0.01	255.33	0.2366	1.3873
0.10	258.46	0.2328	1.3903
0.40	244.28	0.2550	1.2193
1.00	238.91	0.2408	1.2646
5.00	238.91	0.2615	1.1762
10.00	238.96	0.2615	1.1762

Table 1: Test set results. Our implementations of language word model with different values of λ. (i) The mean bias difference increases with decrease in perplexity indicating bias-perplexity trade-off.(ii) The skew of bias difference decreases with increase in λ indicating decrease of less amplified terms near zero bias difference. In other words, bias amplification increases with λ

that male and female words will be predicted with equal probability. In reality, that is not the case.

If we look at the t-SNE projection of the learned embeddings with and without bias regularization in Figure 3-6, we can see the closest words to ”man” and ”woman” in the embedding space (with respect to cosine distance). Interestingly, when we use bias regularization, the words from the defining sets tend to be closest to ”man” and ”woman”. This seems to suggest that bias regularization makes binary gender words closer to each other than other words. This would also seem to suggest that other words move away from the direction of the gender subspace when bias regularization is applied. However, this is not reflected in the bias score. This could indicate that we need a better metric for measuring bias.

It is not clear to us why the model exhibited more bias after when using regularization term, despite some indication that other words are not projecting as much in the gender subspace. Further investigation is required to determine why this is the case.

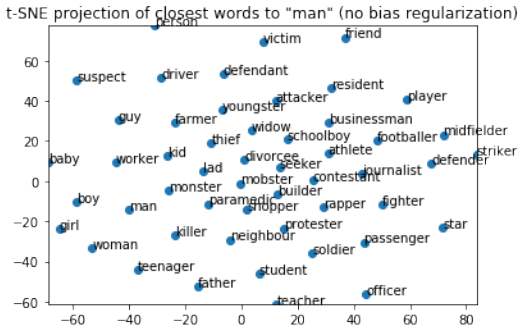

Figure 3: t-SNE projection of learned embeddings from the model without bias regularization. The words shown are ”man” and the 50 closest words (with respect to cosine distance).

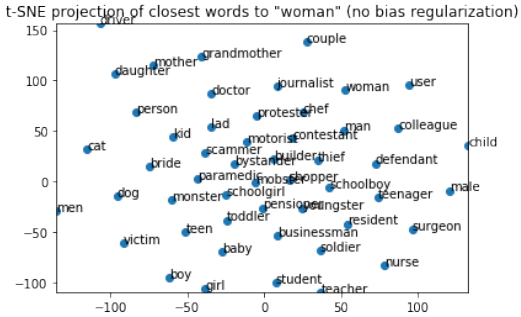

Figure 4: t-SNE projection of learned embeddings from the model without bias regularization. The words shown are ”woman” and the 50 closest words (with respect to cosine distance).

5 Conclusions and Future Work

In this work we wished to address the issue of gender bias in word-level language modeling. We evaluated the bias of the Daily Mail corpus and evaluated the effect of this bias on an RNNs based model. Unfortunately, our proposed bias regularization method increased our bias measure instead of decreasing it, though it seemed to move other words away from gender subspace. Future work is necessary in order to identify the cause of this phenomenon and to propose an improved method

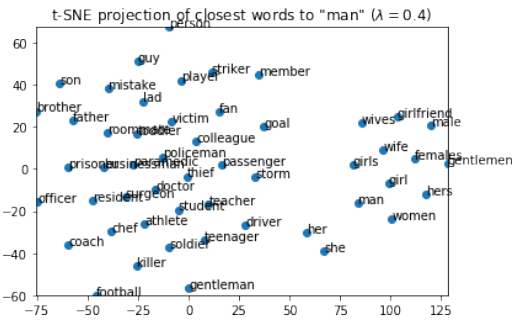

Figure 5: t-SNE projection of learned embeddings from the model with bias regularization ($\lambda = 0.4$). The words shown are "man" and the 50 closest words (with respect to cosine distance).

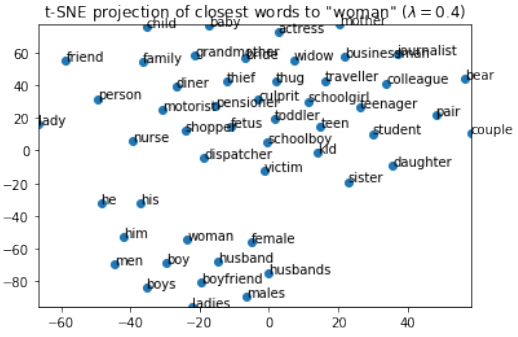

Figure 6: t-SNE projection of learned embeddings from the model with bias regularization ($\lambda = 0.4$). The words shown are "woman" and the 50 closest words (with respect to cosine distance).

that would be successful in reducing gender bias and a better metric for measuring gender bias. It would also be interesting to see how these methods could apply to other types of bias other than gender bias. Towards reducing the effect of bias affecting machine learning models, we believe that it is a worthwhile endeavor to pursue means of mitigating bias during training.

Collaboration Statement

Shikha performed word embedding analysis and debiasing, performed bias calculations, explored the data and results, and implemented the metrics. Jason performed text preprocessing, and implementation some metrics, bias regularization, and some training refinements. Yu set up model and training framework, trained models, and explored hyperparameters for training. All members contributed to writing paper, preparing the presentation, discussion of the debiasing methods, overall discussion, and training of models.

References

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. http://arxiv.org/abs/1506.03340.

Matthew Honnibal et al. 2018. Explosion/spacy: V2.0.11: Alpha vietnamese support, fixes to vectors, improved errors and more. https://doi.org/10.5281/zenodo.1212304.

Daniel S. Jurafsky and James H. Martin. 2008. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice-Hall, Inc., 2nd edition.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Adam Paszke et al. 2017. Automatic differentiation in pytorch .

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

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language.

Biplav Srivastava and Francesca Rossi. 2018. Towards composable bias rating of ai services.

Martin Sundermeyer, Ralf Schluter, and Hermann Ney. 2012. Lstm neural networks for language modeling.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks.