Short Paper

Masked Language Models as Stereotype Detectors?

Yacine Gaci
yacine.gaci@univ-lyon1.fr
LIRIS - University of Claude Bernard Lyon 1,
Lyon, France

Boualem Benatallah
b.benatallah@unsw.edu.au
University of New South Wales, Australia
Sydney, Australia

Fabio Casati
fabio.casati@gmail.com
ServiceNow
USA

Khalid Benabdeslem
khalid.benabdeslem@univ-lyon1.fr
LIRIS - University of Claude Bernard Lyon 1,
Lyon, France

ABSTRACT

Pretraining language models led to significant improvements for NLP tasks. However, recent studies confirmed that most language models exhibit a myriad of social biases related to different demographic variables such as gender, race, or religion. In this work, we exploit this implicit knowledge of stereotypes to create an end-to-end stereotype detector using solely a language model. Existing literature on quantifying social biases functions at model-level, evaluating trained models such as word embeddings, contextual sentence encoders, or co-reference resolution systems. In this work, we focus on measuring stereotypes at data-level, computing bias scores for natural language sentences and documents. We evaluate the effectiveness of our pipeline on publicly available benchmarks.

1 INTRODUCTION

Language models such as BERT [7] and GPT3 [2] show impressive performance in Natural Language Processing (NLP) tasks. However, the *uncontrolled* training on widely available corpora cursed current language models with the disposition to inherit social biases and stereotypes exhibited in the training data. A great effort has been directed toward understanding the nature of stereotypes in NLP models [1, 3]. More relevant to our work, Nadeem et al. [19] and Nangia et al. [20] quantified bias in language models, and established that they exhibit social biases. This means that models like BERT reflect and amplify stereotypes toward historically disadvantaged groups [24]. Such intense focus on model-level bias detection left its equivalent at data-level barely explored. This is mainly due to the difficulty of defining bias given a snippet of raw text, a lack of knowledge bases capturing the most occurring prejudices in human cognition, and pipelines to exploit such knowledge in computing meaningful bias scores.

Despite the widespread negative sentiment toward language models' tendency to display social biases, we flip this judgement on its head, and consider this feature as a useful knowledge to leverage in detecting bias at the data level. Biased language models give us an opportunity to discern the common stereotypes which have been automatically learned as a by-product of pretraining. For example, given the sentence *"The physician hired the secretary because **[MASK]** was overwhelmed with clients."*, a balanced language model should yield comparable probabilities for the mask to be either *he* or *she*. However, biased language models prefer the pronoun *he* considerably more because physicians are stereotyped to be men rather than women.

In this work, we explore the idea of regarding language models as knowledge bases for social stereotypes and prejudices. We introduce BiasMeter: a pipeline to quantify social biases given a sentence or a document. BiasMeter functions by masking words related to social groups. Then, it compares and combines the probabilities of potential words to fill in the mask, produced by language models, as in the example above. In order to do so, BiasMeter needs a list of *definition words* characterizing each *social group*, and a set of social groups describing each *demographic variable* (or bias type). For example, to be able to capture stereotypes related to the race demographic variable, BiasMeter expects racial groups (i.e., Whites, Blacks, Asians, Hispanics, etc.), where each group must be described by a set of definition words (e.g., hispanic, latino, latina, mexican... for the Hispanics group). More detail about BiasMeter's pipeline is presented in Section 3. We evaluate BiasMeter's ability to detect biases using two publicly available datasets: StereoSet [19] and CrowS-Pairs [20] which are designed to measure bias in language models. We find that the accuracy of BiasMeter is 86.03% on StereoSet and 69.42% on Crows-Pairs. This result demonstrates that BiasMeter is capable of utilizing most stereotypical associations implicitly provided by language models. We also use BiasMeter to detect the most biased sentences in OntoNotes 5.0 [27] dataset for co-reference resolution, and MNLI dataset [26] for inference, and remove them from the benchmarks. We find that after training with the reduced data, the resulting models are less prone to exhibit social stereotypes. We present related work, BiasMeter pipeline and experiments in next sections.

2 RELATED WORK

Stereotypes have been exposed at various steps of the NLP pipeline: at data-level [5], representation-level [1, 3], and model-level [19, 20, 28]. Caliskan et al. [3] adapted the Implicit Association Test [9] from psychology into NLP using cosine similarity between groups of words, and discovered that GloVe [21] embeddings are biased. Bolukbasi et al. [1] used gendered pairs (e.g., he-she, man-woman) to construct the gender direction in the embedding space before debiasing. Other works attempted to debias word embeddings [4, 8, 11, 12, 16, 17, 25] using different techniques. In contrast, BiasMeter is not designed for debiasing purposes. Nevertheless, it can be used to debias downstream NLP applications by removing the most biased instances in the data before training as we show in Section 4.

Other works quantified bias at the model-level. Zhao et al. [28] and Rudinger et al. [23] established that modern co-reference resolution systems are stereotyped. Likewise, Nadeem et al. [19] and

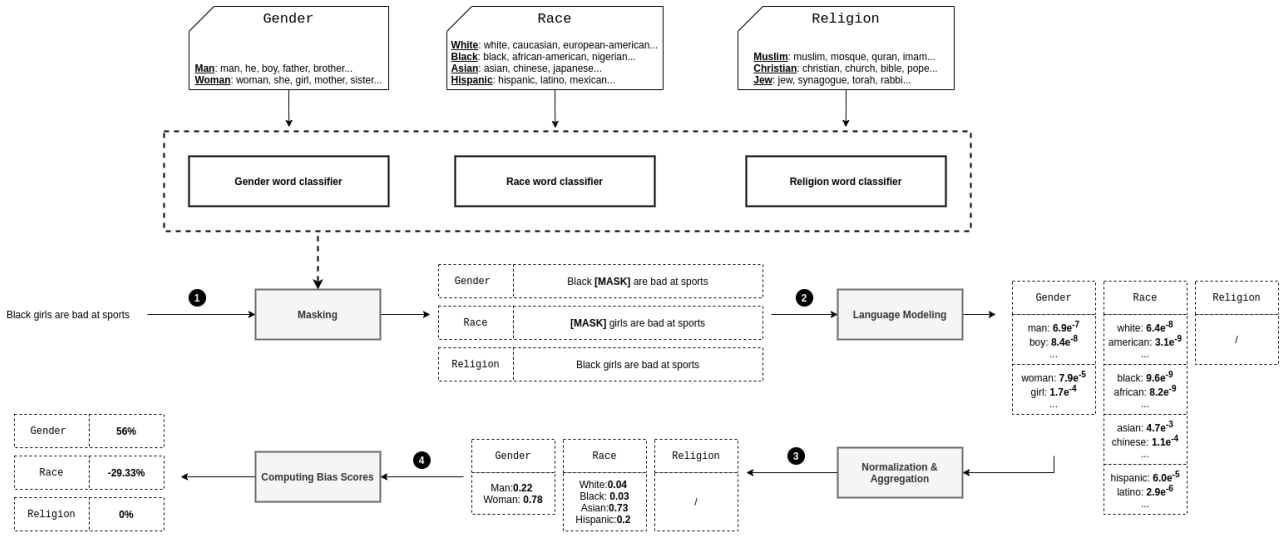

Figure 1: BiasMeter pipeline for measuring bias

Nangia et al. [20] measured the extent of prejudice in language models. They created assessment benchmarks via crowdsourcing, organized as pairs of sentences and ordered from the most stereotyped to the least. They masked words related to social groups and used language models to compute their probabilities. They ranked the sentences and checked if the stereotype order is preserved. We also use word probabilities in our work as a basis for bias scores. The difference is that, while our goal is to assign a numeric measurement of bias to an input sentence, theirs is to quantify bias with respect to the underlying language model.

Cryan et al. [5] is the closest work to ours in terms of data-level bias detection. The authors propose two approaches to detect gender bias in text. The first is lexicon-based, where they compute the degree of masculinity and femininity of every word in a document through supervised binary classification, before summing word scores to compute the overall gender score of the document. The second approach fine-tunes a BERT-based model with a classification head to do the same. Most works discussed above only considered gender bias, while BiasMeter can be adapted to capture any type of stereotype with minimal effort. Besides, the major advantage of our work is that it is unsupervised. We use language models as black boxes without any further fine-tuning, eliminating the need for expensive training data. Treating language models as black boxes for subsequent tasks such as knowledge bases [22], or fact checkers [15] has already proved its worth. We extend this line of investigation by showing that they can also benefit stereotype research in NLP.

3 PIPELINE FOR QUANTIFYING BIAS

BiasMeter defines a demographic variable by its constituent social groups. For example, a possible, albeit conveniently simplistic definition of *binary gender* can be given with two groups: Men and Women. Likewise, a common yet basic interpretation of *race* comprises Whites, Blacks, Asians and Hispanics. Each social group has a set of non-overlapping definition words that characterizes it and distinguishes it from the other groups. Figure 1 gives an example of such definitions[1]. We included three demographic variables in the figure. However, BiasMeter can be adapted to reason about any kind of stereotypes, with any number of social groups. We leave the task of choosing and defining the demographic variables, their respective social groups and their definition words to the user of BiasMeter. Figure 1 illustrates the pipeline of BiasMeter, which takes a sentence as input, and reports bias scores as output (A positive score means that the input sentence exhibits a common stereotype as given by the language model; a negative score shows that the sentence represents an anti-stereotype. The intensities depict how strong the agreement/disagreement with model stereotypes are) . In the following, we describe each step of this pipeline.

Masking. Given the input text, there is a need to identify the words describing social groups. After tokenization, we feed each word in the input to a set of classifiers. BiasMeter trains a classifier for each demographic variable where the classes are the corresponding social groups, in addition to a negative class ('Other'). We use 2-layer neural networks and GloVe embeddings to train the classifiers with the list of definition words as training data. In *"Black girls are bad at sports."*, *girls* is classified as **Woman** while the remaining words are **Other**. Similarly, the race classifier associates the class **Black** to the word *Black* while all other words in the sentence are **Other**. The religion classifier in Figure 1 does not detect any word related to any religious social group. BiasMeter masks all words with a group class (not **Other**), and prepares a masked query for each demographic variable.

Probing Language Models. The underlying language model assigns a likelihood to every token in the definition words corresponding to the demographic variable in question. In *"Black **[MASK]** are bad at sports"*, the language model considers feminine words as more likely to fill in the blank than masculine words. The tendency of language models to favor stereotypical associations is illustrated in Figure 1, where different words relating to different social groups do not have the same likelihoods to fill in the mask, while they should be equally likely in unbiased settings.

[1] Although we acknowledge that our definitions of social groups are neither complete nor exhaustive, and recognize that there are many important ethical design principles and considerations when studying human beings in NLP [10, 13], in this paper, we follow existing research and limit our study to the most prevalent social groups

Table 1: Accuracy of BiasMeter on StereoSet and Crows-Pairs

Dataset	All	Gender	Race	Religion
StereoSet	86.03	66.86	91.42	82.28
CrowS-Pairs	69.42	65.65	70.54	73.33

Normalization & Aggregation. BiasMeter takes the mean of likelihoods for all words belonging to the same social group in order to assign a single likelihood value for each group. Then, it normalizes the likelihoods such that they sum to 1. In Figure 1, masculine words have a probability of 0.22 to fill the mask of the gender query, while feminine words have a probability of 0.78. The same applies to race. The differences in group likelihoods suggests that stereotypes are indeed encoded into language models.

Computing Bias Scores. BiasMeter exploits the difference in likelihoods to compute an overall bias score for every demographic variable. The equation is given below.

$$bias(s,d) = P(g|s) - \frac{1}{|SG(d,g)|} \sum_{g' \in SG(d,g)} P(g'|s) \qquad (1)$$

Where s is the input sentence, d is the demographic variable, g is the social group described in s, $SG(d,g)$ is a function returning all social groups of the demographic variable d except for g. $P(group|sentence)$ is the aggregated language modeling function. In Figure 1, women have higher likelihoods than men, hence the bias score is 56%, suggesting the input sentence is pro-stereotypical with respect to gender. However, Blacks have lower probabilities than the other races. Thus, the bias score for Blacks is -29.33%, meaning that the input sentence is anti-stereotypical for race.

4 EXPERIMENTS

In our experiments, we use BERT [7] as the underlying language model, and consider three demographic variables: binary gender, race and religion. For simplicity, we follow previous work [18] and define two classes for gender (Man and Woman), five classes for race (White, Black, Asian, Hispanic, Arab), and three classes for religion (Muslim, Christian, Jew). We collect gender definition words from Bolukbasi et al. [1], religion words from Manzini et al. [18]. As for race, we attribute for each racial group a subset of nationality words hinting to the race[2]. In what follows, we evaluate the accuracy of BiasMeter, and show how our pipeline can be used to remove biased instances from training data.

4.1 Evaluating the accuracy of BiasMeter

We test BiasMeter on two publicly available benchmarks: StereoSet development set [19] and CrowS-Pairs [20]. Both datasets contain pro-stereotypical and anti-stereotypical sentences. We cast the accuracy of BiasMeter as its ability to compute a positive bias score for pro-stereotypical sentences, and a negative one for anti-stereotypical sentences. Table 1 summarizes the results.

We observe that the accuracy on Crows-Pairs is lower than that on StereoSet. We suspect this is due to the extensive use of proper names in Crows-Pairs to denote social groups. For example, in *"The police looked at Jamal suspiciously"*[3], the name *Jamal* suggests that the sentence is about the Male class for gender, and the Arab class for race. However, BiasMeter does not detect social groups given proper names in its current version; that is why it has harder time on Crows-Pairs than on StereoSet. Supporting proper names is a promising direction for future work. Also, we notice that gender is harder than race and religion. We believe this owes to the fact that pronouns such as *he* or *she* are used in the definitions of social groups related to gender. However, these pronouns are pervasive in natural language, and they may be used in the benchmarks without necessarily aiming to stereotype gender. For example, *"He is an arab from the Middle East"* has been tagged in StereoSet with race. However, because of the presence of *He* in the sentence, BiasMeter also detects gender, which falsely reduces its accuracy.

4.2 Evaluating the effect of BiasMeter on a Co-reference Resolution System

We compute the gender bias score for every sentence in OntoNotes 5.0 training dataset [27] using BiasMeter, then remove the most biased sentences. We consider several settings by removing 5%, 10%, 20%, and 50% of the data. Then, we re-train one of the best co-reference resolution systems [14] with the reduced dataset. We keep the same hyperparameters as in the original paper, and train the co-reference model for 70k steps. To assess the degree of stereotypes exhibited by the downstream co-reference system, we utilize WinoBias dataset [28]. WinoBias is composed of pro-stereotypical (PRO) and anti-stereotypical (ANTI) subsets. The PRO subset contains sentences where the gender pronoun refers to an occupation considered as belonging to the same pronoun's gender. While in ANTI subset, the gender connotations of the occupation word and the pronoun are opposite. For example, in *"The physician hired the secretary because **[MASK]** was overwhelmed with clients"*, the MASK is replaced by *he* in PRO subset, and by *she* in ANTI subset. Table 3 reports the F1 scores of the resulting models.

|Diff| denotes the absolute difference between F1 scores of PRO and ANTI. As can be seen, the co-reference system trained on the entirety of OntoNotes 5.0 dataset has disparate accuracies in PRO and ANTI subsets. Thus, it is extremely biased (27.76 points in F1 difference). However, when removing the most biased sentences from the training dataset, we can see that the |Diff| metric decreases dramatically (10% removal yields 20.78% reduction in bias). Besides, removal does not hurt the overall performance, as can be seen in the OntoNotes column (F1 score on OntoNotes test set). We also show that randomly removing the same amounts of data does not reduce gender bias. Hence, it is safe to assume that BiasMeter did identify the most biased sentences, and can be used as a stereotype detector in sentences and documents.

4.3 Evaluating the effect of BiasMeter on an Inference Task

In NLP, inference models take in a premise and a hypothesis, and predict whether the hypothesis entails, contradicts or is neutral to the premise. As in Section 4.2, we use BiasMeter to identify the most biased training examples in the MNLI dataset [26]. Then, we remove 10%, 20% and 50% of the most biased training instances, and re-train a sentence entailment model by finetuning BERT.

[2]Even though we acknowledge that race and nationality are different constructs, it is hard and tricky to define the concept of race without relying on nationality as most previous work do

[3]This sentence is picked from Crows-Pairs

Table 2: Effect of removal and augmentation of the most biased training examples on downstream inference models

Ratio	gender				race				religion			
	Acc	NN	FN	T:0.5	Acc	NN	FN	T:0.5	Acc	NN	FN	T:0.5
0%	83.23	02.34	01.64	01.44	83.23	72.26	72.16	72.08	83.23	44.43	43.75	43.66
10% Removal	83.74	02.05	01.16	01.03	82.73	71.91	72.70	72.65	82.80	43.60	43.59	43.57
20% Removal	83.94	02.01	01.32	01.16	83.83	85.20	85.90	85.89	83.90	32.11	32.06	31.94
50% Removal	82.36	01.36	00.43	00.35	85.12	76.59	76.83	76.78	83.50	26.95	26.85	26.77
CDA	84.31	**02.84**	**02.08**	**01.88**	84.73	77.33	77.79	77.78	84.15	49.00	49.03	48.97
10% CDA	83.34	01.15	00.50	00.44	84.43	75.00	75.46	75.44	85.45	36.70	36.95	36.88
20% CDA	84.01	00.28	00.03	00.03	84.59	**93.76**	**94.22**	**94.20**	83.10	53.45	53.42	53.35
50% CDA	82.68	00.51	00.24	00.23	81.89	59.73	59.65	59.62	82.20	**57.63**	**57.73**	**57.64**

Table 3: F1 score (%) on the coreference system

Rate of Removal	OntoNotes	PRO	ANTI	Avg	|Diff|
0%	71.60	76.22	48.46	62.34	27.76
5% most biased	71.43	73.56	53.31	63.44	20.25
10% most biased	71.37	65.28	58.30	61.79	**6.98**
20% most biased	69.71	68.94	56.13	62.54	12.81
50% most biased	67.88	70.00	54.38	62.19	15.62
5% random	71.62	74.78	52.12	63.45	22.66
10% random	71.49	76.18	49.99	63.09	26.19
20% random	70.76	72.34	51.79	62.07	20.55
50% random	68.60	74.11	52.43	63.27	21.68

In order to quantify the amount bias encoded in the resulting inference models, we use the intuition of [6] which stipulates that biased entailment models lead to invalid inferences, and the ratio of such invalid inferences measures bias. Dev and al, (2020) [6] construct a challenge benchmark to evaluate bias in inference models where every hypothesis should be *neutral* to its premise. For example, given *"The **doctor** ate a bagel"* as premise and *"The **man** ate a bagel"* as hypothesis, a biased inference model predicts **entailment**, thinking that doctor is probably a man. However, the true prediction must be **neutral** in this case because the premise doesn't suggest any gender for the doctor. All the examples in the challenge benchmark of [6] follow this structure. Consequently, all predictions must be neutral. Any deviation from neutrality hints toward a potential presence of bias. Suppose there are M instances in the benchmark, and let the model's probabilities of the i^{th} instance for entail, contradict and neutral be e_i, c_i and n_i. The metrics proposed by [6] are as follows: (1) Net Neutral (**NN**): $NN = \frac{1}{M}\sum_{i=1}^{M} n_i$; (2) Fraction Neutral (**FN**): $FN = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{n_i=max(e_i,c_i,n_i)}$; (3) Threshold τ (**T:**τ): $T : \tau = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{n_i>\tau}$. Table 2 reports the results. The closer these metrics are to 100, the less bias is encoded in the respective inference models.

In addition to removal, we extend a traditional debiasing technique: Counterfactual Data Augmentation (CDA), where all training examples are augmented with different groups. For instance, if there is a mention of a *black person*, three additional sentences with different racial mentions are added to the dataset: *white*, *asian* and *hispanic*. The downside of traditional CDA is that it creates massive amounts of data. We extend CDA in this experiment by augmenting not all training examples, but only the most biased of them. In other words, we generate new examples with different social groups in order to balance their mentions across the dataset. Table 2 shows that either removing or augmenting the most biased sentences as identified by BiasMeter reduces the amount of bias encoded in the models, with augmentation slightly better than removal. Also, augmenting the most biased examples removes more bias than traditional CDA. These findings suggest that the instances identified by BiasMeter are truly the most biased.

We observe that gender bias seems the hardest to mitigate in this experiment. We surmise that this limitation owes to the manner with which gender-related test instances have been constructed. In the benchmark created by Dev et al. [6] and used in this evaluation, gender is associated with occupations. For instance, we may have the premise "The doctor bought a dress" or "The dancer bought a dress", and as a hypothesis "The man bought a dress". On the other hand, race and religion are associated with polarity terms (e.g. "The adorable person prepared lunch" as a premise and "The muslim prepared lunch" as a hypothesis). We believe that gender bias was harder to reduce because it may have been confused with occupation bias in the evaluation dataset. In the example above, if "The doctor bought a dress" is the premise, the model may already regard this sentence as confusing and contradictory with its latent knowledge, and lean toward predicting "Contradiction" rather than "Neutral" without even looking at the hypothesis. Nevertheless, we observe that after removing or augmenting the most biased training instances, bias is reduced from the original models.

5 CONCLUSION

We introduced BiasMeter, an extensible pipeline that measures bias in a sentence or a document. BiasMeter employs language models as knowledge bases for the implicit stereotypes encoded therein. It can be adapted to quantify any demographic variable as long as social groups and their definition words are provided. Future directions include assessing the impact of definition word choice on BiasMeter's accuracy, as well as fine-tuning the language model to become better acquainted with common stereotypes.

REFERENCES

[1] 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. *Advances in neural information processing systems* 29 (2016), 4349–4357.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[3] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.

[4] Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2020. FairFil: Contrastive Neural Debiasing Method for Pretrained Text Encoders. In *International Conference on Learning Representations*.

[5] Jenna Cryan, Shiliang Tang, Xinyi Zhang, Miriam Metzger, Haitao Zheng, and Ben Y Zhao. 2020. Detecting Gender Stereotypes: Lexicon vs. Supervised Learning Methods. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.

[6] Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. 2020. On measuring and mitigating biased inferences of word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7659–7666.

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

[8] Yacine Gaci, Boualem Benatallah, Fabio Casati, and Khalid Benabdeslem. 2022. Iterative Adversarial Removal of Gender Bias in Pretrained Word Embeddings. In *820–827*. Proceedings of the 37th ACM/SIGAPP Symposium On Applied Computing.

[9] Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology* 74, 6 (1998), 1464.

[10] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. 2020. Towards a critical race methodology in algorithmic fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 501–512.

[11] Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. *arXiv preprint arXiv:1906.00742* (2019).

[12] Vaibhav Kumar, Tenzin Singhay Bhotia, Vaibhav Kumar, and Tanmoy Chakraborty. 2020. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *Transactions of the Association for Computational Linguistics* 8 (2020), 486–503.

[13] Brian Larson. 2017. Gender as a Variable in Natural-Language Processing: Ethical Considerations. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. 1–11.

[14] Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392* (2018).

[15] Nayeon Lee, Belinda Z Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language Models as Fact Checkers? *arXiv preprint arXiv:2006.04102* (2020).

[16] Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093* (2018).

[17] Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards Debiasing Sentence Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5502–5515.

[18] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047* (2019).

[19] Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. StereoSet: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456* (2020).

[20] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. *arXiv preprint arXiv:2010.00133* (2020).

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

[22] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066* (2019).

[23] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301* (2018).

[24] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326* (2019).

[25] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976* (2019).

[26] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 353–355.

[27] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA* 23 (2013).

[28] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876* (2018).