Fair Data Generation using Language Models with Hard Constraints

SK Mainul Islam
IIT Kharagpur, India
mainul.islam@iitkgp.ac.in

Abhinav Nagpal
Harman International
abhinav.nagpal@harman.com

Balaji Ganesan
IBM Research, India
bganesa1@in.ibm.com

Pranay Kumar Lohia
IBM Research, India
plohia07@in.ibm.com

Abstract

Natural language text generation has seen significant improvements with the advent of pre-trained language models. Using such language models to predict personal data entities, in place of redacted spans in text, could help generate synthetic datasets. In order to address privacy and ethical concerns with such datasets, we need to ensure that the masked entity predictions are also fair and controlled by application specific constraints. We introduce new ways to inject hard constraints and knowledge into the language models that address such concerns and also improve performance on this task.

1 Introduction

Deep Neural Networks models have become the state of the art in many fields with models being trained on different kinds of data including images where they became popular, to unstructured text especially natural language processing, to graphs, and to a limited extent in structured data.

To reap the benefits of these advancements, we need to be able to train models on real world datasets including customer data available with enterprises, and citizen data available with governments. This needs to be balanced with the ethical requirements in building fair models, and complying with regulations against the use of personal data for tasks not specifically authorised by users with informed consent. We use the term *personal data* to denote data of both humans and organizations.

The solution to this problem so far has predominantly been synthetic data generation like AMLSim (Suzumura and Kanezashi [2021]) and anonymization like MIMIC III (Johnson et al. [2016]). We believe synthetically generated data has limitations and often does not represent real world scenarios. To generate realistic datasets, it is productive to start with anonymized versions of real data where personal data entities are redacted and then replace the personal data entities with entities of the same entity types. We believe anonymization of personal data should be the default while training all machine learning models, even when the data is publicly available like in Wikipedia, news articles and social media.

We propose using language models to impute entities in place of redacted personal information in unstructured text datasets. Language models have been shown to perform well on some of the mask prediction (fill-in-the-blank) tasks. From subword, to word and now to sentences, latest language models have progressed. While there have been attempts to generate entities using language models, there is still lot of scope for improvement.

We do not just want the language models to generate entities, but generate them with several additional characteristics like context, fairness, and appropriate entity type. One way to achieve this is using

1st CtrlGen: Controllable Generative Modeling in Language and Vision Workshop at NeurIPS 2021.

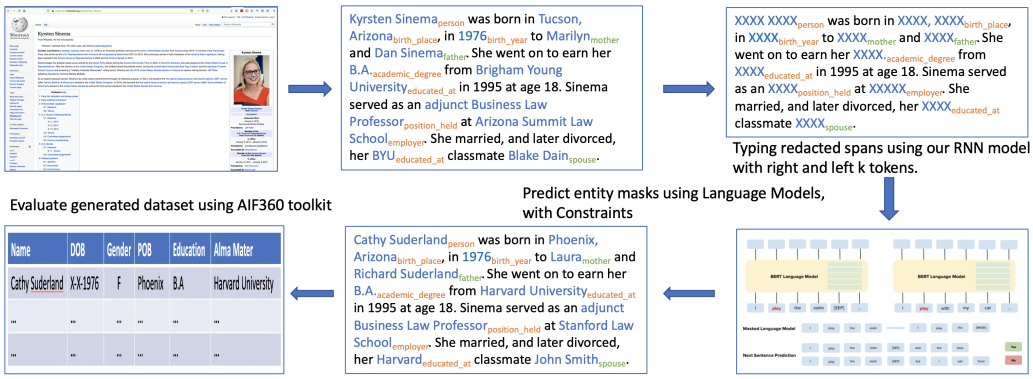


Figure 1: We generate datasets from redacted text, using language models with constraints

constraints. Introducing constraints on the mask prediction task or in general controlled language generation has been an active area of research. Sha [2020] introduced a gradient and lexical approach to introduce constraints. Khalifa et al. [2020] proposed a distributional approach to controlled text generation. However external constraints like fairness, commonsense and temporality have proven to be harder.

One approach to generate fair data from unstructured text, and then to populate structured or graph data, would be to simply discard generated tokens and entities which skew the distribution. However, this method has limitations in domains where the language models have not been trained well and requires several counters to be maintained outside of the data generation solution. There have been few attempts to make the language models aware of entities Shahbazi et al. [2019] and to guide counterfactual generation to desired distributions.

In this work, we introduce new ways to constrain the output of a language model for the mask prediction task, so that we can ensure diversity in the generated entities by default, and also to introduce application specific controls to make the predictions fairer. As in the desiderata mentioned earlier, we optimize the mask prediction for 3 factors namely entity type, diversity, and context.

We then introduce a way to incorporate knowledge embeddings that complement language embeddings. As the name suggests, knowledge embeddings fill the knowledge gaps in the language models and also make it easier to incorporate knowledge that would otherwise require substantial amount of training. For example, <person, alma mater, institution> could be a relationship that we wish to introduce in our model. This can be accomplished far more easily by incorporating knowledge embeddings whereas language models will need several instances of this relationship to learn the same.

2 Related Work

POINTER Huang et al. [2020] proposed conditional text generation by replacing nouns in the given condition lexicons with some protected entities. Sha [2020] presented an entity aware language model called Entity-Elmo. Zhang et al. [2020] introduces a hard-constrained based text generation using insertion transformer Stern et al. [2019] in a non auto-regressive manner.

Huang et al. [2020] quantifies and de-biases the induced bias in the pre-trained language models on sentiment classification task. Conditional lexicons are expanded with protected entities using language models (GPT-2) and then evaluated sentiment distribution with respect to those entities to quantify the bias involved in the pre-trained language models and perturbed examples are coupled with original expanded sentence in the proposed model to de-bias the language model by minimizing the fairness loss (cosine similarity of representations of the given two inputs obtained from language model). Feder et al. [2020] analysed counterfactual language models for generating counterfactual example generation and measuring bias in the language models.

Qutaineh et al. [2020] proposed a novel data augmentation approach using Monte Carlo Tree Search (MCTS) as the optimization strategy and incorporating entropy as one of the optimization criteria in their active learning solution.

Evaluating the output of NLG models has received quite a bit of attention in recent times. Ribeiro et al. [2020] introduced checklists to evaluate NLP models in general. Tevet and Berant [2020] proposed a score to measure diversity, while Agarwal et al. [2021] introduced an unfairness score.

Zhang et al. [2021]. CoLAKE Sun et al. [2020] introduced a word knowledge embedding which is jointly learnt instead of learning the knowledge embeddings independent of the language task. We have modeled our implementation around the same approach and compare our implementation with CoLAKE. Li and Liang [2021], Yu et al. [2020], He et al. [2019], Khalifa et al. [2020], Keskar et al. [2019] proposed a method for controllable text generation using what they call as control codes.

GraphMask Schlichtkrull et al. [2021], GNN Explainer Ying et al. [2019], PGM-Explainer [Vu and Thai, 2020] are some of the recent efforts in explaining GNN model predictions. We use the GraphMask method since their work was on NLP tasks like our task here. The original paper uses GraphMask for explainability in question answering and semantic role labelling.

3 Our Approach

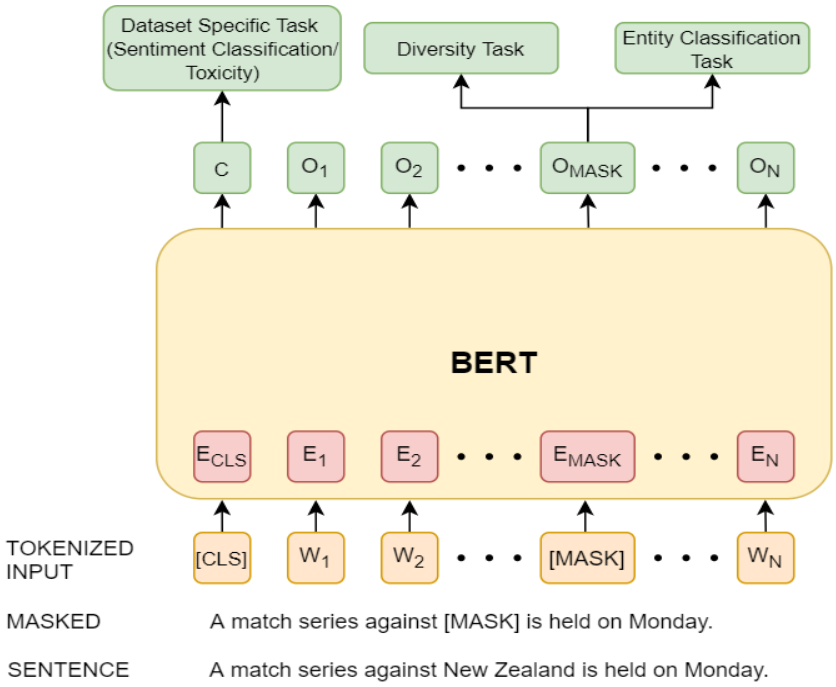


Figure 2: Mask prediction for fair data generation

There have been number of approaches to introduce constraints on the language model output. Our requirement is to restrict mask predictions to personal data entities and to make the data sufficiently diverse and controllable for fairness. Sha [2020] proposed an approach where the tokens predicted by the language models could be restricted to a list of values.

We can follow a similar approach by using a list of protected variables (of personal data entities) which can be predicted for a mask. However in order to generate values that are sufficiently diverse, we treat this as a collective learning task.

We introduce losses for enforcing constraints on language model based generation: Cross entropy loss for generating entities of same type, KL divergence loss for diverse candidate entity generation, cross entropy loss for downstream task specific fair entity substitution. Consider a given input $S = (w_1, ..., w_i, ..., w_n)$, with corresponding entity type label $E = (e_1, ..., e_i, ...e_n)$, the goal of this task is to replace a personal entity w_i of type e_i in S with another entity $\hat{w}_i$ of type e_i.

3.1 Entity type factor

Non auto-regressive language models like BERT Devlin et al. [2019] estimate the probability distribution of a word w, given a context as $p(w_i|w_{<i}, w_{i+1:n})$ 2 $\mathbb{R}^{|V|}$, where $|V|$ is the number of (sub)words in the language model vocabulary. To enforce the entity type constraint, we introduce entity sequence labeling task and the loss corresponding for that task as:

$$L_{Seq} = \sum_{i=1}^{n} \sum_{e_c} \log p(e_c|w_i), e_c \; 2 \; \{e_1, ..., e_n\}$$

3.2 Diversity factor

To generate a diverse candidate entity for replacement, we leverage the idea of maximizing the entropy of word prediction probability Madaan et al. [2021], which is similar to minimize the KL divergence between the word prediction probability $p(w_i|w_{<i}, w_{i+1:n})$ and and an uniform distribution u, using the lemma $H[w_i] = \log|V| \quad KL(\mu||u)$, where w_i is estimated by distribution $p(w_i|w_{<i}, w_{i+1:n})$ with mean as μ, and H is the Shannon Entropy of w_i, and $|V|$ is the number of values w_i can take on from the vocabulary. Hence the KL divergence loss

$$L_{Diversity} = KL(p(w_i|w_{<i}, w_{i+1:n})||v), v \quad Uniform(1/|E_i|)$$

where E_i is the set of entities with type e_i.

3.3 Debiasing factor

To enforce the fair entity substitution, we leverage the idea of debiasing language model on downstream tasks Liang et al. [2020]. We utilize the bias representation h_{bias} estimated from gender words and remove that representation from the sentence representation h_{sent} of the instances for the downstream task. We optimize this debiasing scheme using the task specific loss,

$$L_{Task} = CrossEntropy(p(\hat{y}|h_{sent} \quad h_{bias}), y)$$

Finally, we optimize all the losses in an end-to-end setting as following:

$$\min L_{Total} = L_{MLM} + \alpha_1 L_{Seq} + \alpha_2 L_{Diversity} + \alpha_3 L_{Task}$$

where L_{MLM} is the Masked Language Modeling loss to estimate $p(w_i)$, and α_1, α_2, α_3 are hyperparameters.

Knowledge Embeddings

We use the methods described in Vannur et al. [2021] to populate a property graph of people. The details of the populated graph are as shown in Table 1.

	IMDB Reviews	**Jigsaw**
Documents	50000	1999516
Sentences	328331	3460461
Entities	19099	58986
Relations	2331	11161
Entity Types	23	27
Relation Types	2	4

Table 1: Statistics on the property graphs generated

Once we have populated a knowledge graph, there are a number of methods by which we could incorporate the knowledge embeddings for mask prediction.

We began by using the knowledge graph to generate prior probabilities for the candidate personal data entities. For any given entity, we generated a list of similar entities using the GNN embeddings,

and used them to generate the diversity factor as discussed in Section 3.2. We can also generate knowledge embeddings and use it along with the language embeddings like done in Sun et al. [2020]. We can also generate sentence embeddings for the sentiment classification task described in Section 3 and use them along with the language embeddings. We leave these two experiments for a future version of this work.

4 Experiments

We perform our experiments on the Jigsaw toxicity dataset and the IMDB reviews dataset. An example sentence 3 and the different personal data entities generated are shown in Table 2.

4.1 Experimental Setup

We conduct our experiments on a single machine with 16 GB CPU memory and 16 GB GPU memory for most of the data pre-processing and knowledge graph embedding. We use a shared TPU instance for pre-training the language model on our datasets, though these can easily be performed on the CPU/GPU with more time. We use PyTorch framework (Paszke et al. [2019]) for most of the tasks. We use the popular BERT language model and then pre-train on our datasets namely IMDB Reviews and Jigsaw.

4.2 Results

The performance of our Fair Data Generation (FDG) model and the baselines measured using perplexity are as shown in Table 3. We calculate perplexity as the exponential of average negative log-likelihood of a sequence. For a tokenized sequence $S = (w_i, ..., w_n)$, the perplexity is calculated as:

$$PPL(S) = exp(\frac{1}{n}\sum_{i}^{n} \quad \log p(w_i \boldsymbol{j} w_{<i}, w_{i+1:n}))$$

We observe that the perplexity of our proposed FDG model is worse than the baseline BERT model. The reason behind this worse performance with respect to perplexity is the trade-off between the context association and bias association in the language model. Consider the example from Liang et al. [2021].

The man performing surgery is a doctor bias association
The man performing surgery is a doctor context association

Here, the language model predicts "*doctor*", when the input context is "*The man performing surgery is a*", which is a correct prediction according to the context "*surgery*", but the output is biased with respect to the association between "*man*" and "*doctor*". A pre-trained language model trained on a large (probably biased) corpus like BERT focuses more on context association and the debiased language model like our proposed model FDG focuses more on removal of bias association, and hence generates less associated words with respect to the word "*man*" and as a result, achieves poor perplexity score compared to the previous pre-trained BERT model.

Table 2 show the generated data by masking entities in an example sentence. The generated entities have the same number of tokens as in the original entity.

Bruce Campbell, the British actor went to Hollywood and enthralled audiences!

Figure 3: Example input to the Fair Data Generation model with masks highlighted

Given our goal to use this fair data generation model to produce personal data entities, measuring the ability of the model to produce a diverse and large number of entities is important. So we compare the number of entities in the original datasets as shown in Figures 4a and 4b. The number of mask predictions is an hyperparameter (10 in this example run). But the generated sentences are annotated with a PII extraction pipeline which is agnostic to the original and generated text.

Entity	Classification	Mask Prediction
Actor	job_title	Director, Actor, Write, Actress, Filmmaker, Producer, Soldier, Detective, General, Gangster
Hollywood	location	Hollywood, War, America, Vietnam, France, Germany, Fever, London, ##Ani, ##Tan
British	nationality	American, Hollywood, British, ##Sie, Irish, Indian, ##N, ##Wan, Born, Oz
Bruce Campbell	name name x	John John ., David Davids, Michael . Er, Robert On, James Leey, George Jack, Jack Mc , Richard Jamese, William Robertt, Peter George Lee

Table 2: Entities predicted by Fair Data Generation using Language Models

Model	Dataset	Perplexity#
LSTM	Jigsaw	1632.45
	IMDB	1265.80
BERT	Jigsaw	323.17
	IMDB	223.15
FDG	Jigsaw	434.15
	IMDB	352.85

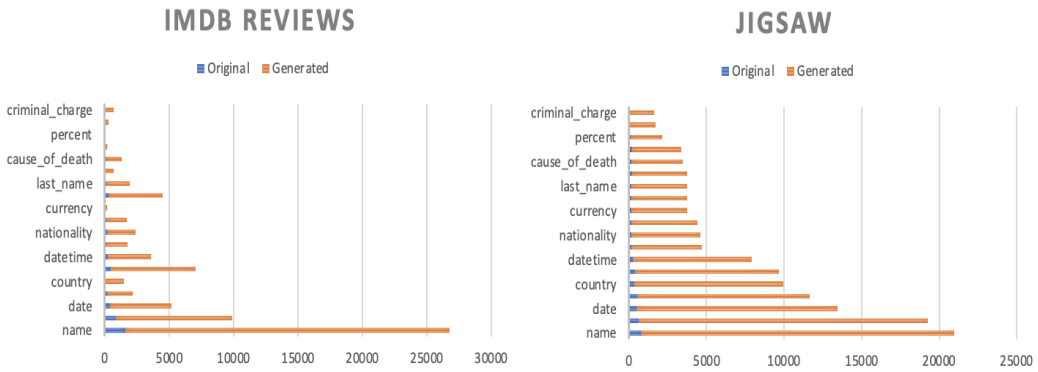
Table 3: Performance of our FDG model on the mask prediction task

5 Testing

In this section, we present results from three kinds of tests that we performed on the output text, that includes the predicted masks. We test for behaviour - the generated text should be as natural as ground truth data, fairness - the predicted entities should not be biased against minority communities in the real world, and adversarial - should be able to withstand known adversarial attacks.

5.1 Behavioural Testing

Following the Checklists idea proposed in Ribeiro et al. [2020], we perform a behavioural test by using the output of our Fair Data Generation (FDG) model output as input to a fine grained entity classification model described in Nagpal et al. [2022]. We begin by classifying all the entities in the raw IMDB and Jigsaw datasets.



(a) IMDB Reviews

(b) Jigsaw

Figure 4: Entity types in 5000 IMDB and Jigsaw samples, and corresponding entities generated by our FDG model

Next, we report the performance on a random sample of 5000 entities, that we used as input in Section 4. We are observing if the performance of a downstream task like fine grained entity classification is

Dataset	Acc.	Macro F1	Micro F1
IMDB	0.979	0.768	0.992
IMDB Gen	0.962	0.80	0.983
Jigsaw	0.943	0.830	0.969
Jigsaw Gen	0.942	0.832	0.967

Table 4: Results for BERTEC model for behevioral testing on the original and generated datasets

similar both in the original dataset and the dataset where entities have been replaced by our FDG model. As shown in Table 4, there is no noticeable difference in performance.

5.2 Fairness Testing

Fairness can be measured as group fairness and individual fairness (Lohia et al. [2019]). To evaluate group fairness in the generated data, we use disparate impact as described in Bellamy et al. [2019]. For individual fairness, we generate *counterfactuals* by perturbing sensitive tokens present in original texts and check whether predictions change or not. We use the *unfairness score* defined in Agarwal et al. [2021] to evaluate individual fairness. A lower percentage ($<$10%) of original samples having unfairness makes the model individually fair.

Dataset	**Attribute**	**minority class**	**count**	**majority class**	**count**	**DI**	**label**
IMDB	religion	other	51	Christianity	62	0.8226	fair
	gender	F	1945	M	7555	0.2500	biased
Jigsaw	religion	other	826	Christianity	1032	0.8004	fair
	gender	F	4646	M	21522	0.2159	biased

Table 5: Group fairness evaluation on the original IMDB and Jigsaw datasets

In Table 5, we present the disparate impact results using the 80% rule on the original dataset. Using this rule, we identify the majority class based on the frequency of occurrence of different values in each protected attribute. If the ratio of other values count over the majority class count is above 0.8, then the dataset is not biased on that protected attribute. We observe that both the datasets are fair on the religion attribute but biased on the gender attribute.

Dataset	Attribute	Unfairness (%)	Label
IMDB	gender	2.5	fair
	race	0	fair
	religion	2.94	fair
Jigsaw	gender	5.3	fair
	race	4.54	fair
	religion	2.08	fair

Table 6: Individual fairness evaluation on the FDG model output.

In Table 6, we present the *unfairness score* in % associated with sensitive attributes like *gender*, *race*, and *religion* in two downstream tasks, sentiment Analysis in IMDB dataset, and toxicity classification in Jigsaw Dataset using our proposed FDG model. We observe that our FDG model generates fair output wrt all the protected attributes considered.

5.3 Adversarial Testing

For checking the FDG model against adversarial attacks and to evaluate its robustness, we use the methods in TextAttack Morris et al. [2020]. It performs certain transformations on the dataset with respect to certain constraints, producing new samples. An example adversarial change is as shown in Figure 5.

The transformations were only applied if they met the constraints such as max words perturbed limit of 5 words, disallowing the modification of words which have already been modified and Bert Score

The actors play wonderfully , especially Kenneth Branagh himself .
The actors play criminals , especially Kenneth Branagh.

Figure 5: Example adversarial change to the input.

Zhang et al. [2019] less than 0.8. As shown in Table 7, there is only a small difference between the original and adversarial sample results. This could be an indication that our FDG model is unaffected by the adversarial changes we introduced in the input.

Model	Acc.	Macro F1	Micro F1
IMDB Adv	0.969	0.763	0.984
IMDB Adv Gen	0.949	0.754	0.974
Jigsaw Adv	0.951	0.842	0.976
Jigsaw Adv Gen	0.946	0.839	0.971

Table 7: BERTEC entity classification results on adversarial samples and FDG output on the same.

6 Conclusion

In this work, we introduced a solution for generating fair datasets from unstructured data where the entities predicted are personal data entities. Our evaluation and analysis using behavioral, adversarial and fairness testing shows that the generated datasets closely resemble the original datasets, while improving on fairness metrics. There is a trade-off between the language model performance and bias removal, which will continue to motivate our future work on modeling a fair language model comparable with state-of-the-art pre-trained language models like BERT and GPT.

References

Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. *arXiv preprint arXiv:2102.13186*, 2021.

Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. Causalm: Causal model explanation through counterfactual language models. *CoRR*, abs/2005.13407, 2020.

Bin He, Di Zhou, Jinghui Xiao, Qun Liu, Nicholas Jing Yuan, Tong Xu, et al. Integrating graph contextualized knowledge into pre-trained language models. *arXiv preprint arXiv:1912.00147*, 2019.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 65–83. Association for Computational Linguistics, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. Towards debiasing sentence representations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5502–5515, 2020. doi: 10.18653/v1/2020.acl-main.488. URL https://doi.org/10.18653/v1/2020.acl-main.488.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR, 2021.

Pranay K Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R Varshney, and Ruchir Puri. Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2847–2851. IEEE, 2019.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 13516–13524, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17594.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.

Abhinav Nagpal, Riddhiman Dasgupta, and Balaji Ganesan. Fine grained classification of personal data entities using language models. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410, Online, November 2020. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist, 2020.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for nlp with differentiable edge masking, 2021.

Lei Sha. Gradient-guided unsupervised lexically constrained text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8692–8703, 2020.

Hamed Shahbazi, Xiaoli Z. Fern, Reza Ghaeini, Rasha Obeidat, and Prasad Tadepalli. Entity-aware elmo: Learning contextual entity representation for entity disambiguation, 2019.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 5976–5985. PMLR, 2019.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. Colake: Contextualized language and knowledge embedding, 2020.

Toyotaro Suzumura and Hiroki Kanezashi. Anti-Money Laundering Datasets: InPlusLab anti-money laundering datadatasets. `http://github.com/IBM/AMLSim/`, 2021.

Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation. *arXiv preprint arXiv:2004.02990*, 2020.

Lingraj S Vannur, Balaji Ganesan, Lokesh Nagalapatti, Hima Patel, and MN Tippeswamy. Data augmentation for fairness in personal knowledge base population. In *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2021 Workshops*, volume 12705, pages 143–152, 2021.

Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks, 2020.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNN explainer: A tool for post-hoc explanation of graph neural networks. *CoRR*, abs/1903.03894, 2019. URL `http://arxiv.org/abs/1903.03894`.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*, 2020.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. Smedbert: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. *arXiv preprint arXiv:2108.08983*, 2021.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. POINTER: constrained progressive text generation via insertion-based generative pre-training. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8649–8670. Association for Computational Linguistics, 2020.