Evaluating Language Models on Intersectional Biases

Students of the University of Osnabrück, 2024
Course: "Implementing ANNs with Tensorflow"

April 14, 2024

Abstract

Intersectional Bias describes a specific discrimination effect caused by overlapping social factors that is different from the discrimination effect these factors would cause on their own. In this project we intend to evaluate the intersectional bias of two major social factors, gender and race, in a pre-trained Large Language Model (LLM) compared to a small Long-Short-Term Memory (LSTM) language model we built, trained and fine-tuned from scratch. The evaluation on both language models was performed on text generated following a query including one gender- and one race-related word at the same time to represent intersectionality. The output was evaluated on the connotation of adjectives. We expected a negative intersectionality bias for subgroups such as *black females*, but the results of the LLM were not as expected: Most pronounced was a negative bias towards *white* individuals. The results of the small language model could not be properly evaluated on the pre-defined criteria, as it did not perform well enough to generate a large and diverse enough amount of adjectives reasonable to be used for evaluation.

1 Introduction

Since the root of training data for Large Language Models consists of human-made datasets, these models are prone to inherit societal biases unless it is actively counteracted (Malik, 2023). Different societal biases have been evaluated in existing studies (Dong et al., 2024; Omiye et al., 2023), but the intersectional effect especially has not been represented much in LLM evaluation research. Even research that looks at more than one social factor often only evaluates them one-by-one (Malik, 2023; Kiritchenko & Mohammad, 2018). Social factors have many different facets, such as gender, sexuality, race, disability and religion (Ghai et al., 2021). Evaluating them all at once would be beyond the scope of this project, so the choice fell onto the two most prevalent factors: gender and race.

We built a small LSTM language model and trained it on the *wiki_auto*[1] dataset created by Jiang et al. (2020). We generated text on queries that included a gender- and race-related word at the same time, as opposed to queries that include only one factor, to represent intersectionality in each example. The same type of generation was done on a pre-trained LLM (GPT2) (HF Canonical Model Maintainers, 2022; Radford et al., 2019). The goal for the small LSTM language model was to evaluate it after tuning of hyperparameters. While building and training the model we ran into time problems (training for one epoch taking a day) and hardware limitations (limited VRAM).

2 Related work

There has not been much research on intersectional biases in LLMs so far. Most research has focused on one social factor only, such as gender or race (Dong et al., 2024; Omiye et al., 2023). Gender is the most common bias evaluated in LLMs, possibly because this social factor is still one of the most prevalent issues in language-focused Artificial Intelligence (AI), due to its importance in grammar. Of the papers that took multiple social factors into consideration, most evaluated the intersection of gender and race (Malik, 2023; Kiritchenko & Mohammad, 2018; Tan & Celis, 2019). Previous bias studies set their focus on different components of LLMs: Some evaluated word embeddings (Ghai et al., 2021) or the dataset (Tan & Celis, 2019) *before* training, whereas others focused on evaluating generated sentences *after* training. For word embeddings Ghai et al. (2021) created a tool to visualize intersectional biases, taking into account 5 social factors, namely Gender, Religion, Race, Age, and Economic at once. An example of analysis after training is a study done by Malik (2023), which evaluated contemporary LLMs such as GPT-3.5 on two different tasks. The first task prompted the models to classify professions (gender bias analysis), the second task involved prompts asking the model for people descriptions (racial bias analysis). They compared the results to previous experiments of theirs involving less powerful and complex LLMs such as GPT-2. This study on the one hand showed a promising trend towards fewer gender biases in newer models, on the other hand that concerns about racial biases remain. Kiritchenko & Mohammad (2018) also evaluated biases with respect to gender and racial factors, with the goal to analyse how emotion-coupled sentences containing different race or gender words are interpreted by LLMs. They created the "Equity Evaluation Corpus (EEC)", which consists of query sentences designed to spot gender or racial biases in LLMs more easily. However, the analysis in both Malik (2023) and Kiritchenko & Mohammad (2018) did not aim at evaluating intersectional biases, despite looking at two social factors, as their analysis treated each factor separately. An example of intersectional analysis, considering multiple social factors at once, is Tan & Celis (2019), who evaluated intersectional biases at a contextual, not just at sentence level. The authors highlighted methodological

[1]which can be found here: https://www.tensorflow.org/datasets/catalog/wiki_auto#wiki_autoauto

challenges of previous attempts in bias evaluation, especially criticising the sole focus on sentence level. Tan & Celis (2019) found that intersectional biases affecting the least privileged identities were stronger than the biases of their intersecting minority groups separately.

3 Methods

3.1 Preparing the LLM GPT2

As an LLM we chose GPT2 (HF Canonical Model Maintainers, 2022), for it being a freely available model that can be put to practice without much tweaking and additional code. GPT2 is a Decoder Transformer model that can generate text following a query. While it is not the most up-to-date version of its type, it is sufficient for our evaluation. We chose GPT2-large with 774 million parameters and 20 attention heads because this is the largest version we could reasonably run on our computers. Due to the Decoder-only structure, GPT2 is not able to "understand" the query (unlike models like *BERT*, which is an Encoder-only), but it can predict words following others. A full Encoder-Decoder-structure would have been more useful for bias evaluation, as the answer structure and content would likely be more custom to what the query is asking. However, we had trouble finding an easily available pre-trained Encoder-Decoder LLM and our attempts to stack GPT2 onto BERT (Turc et al., 2019) were unsuccessful. We were unable to get the Tensorflow *EncoderDecoderModel*[2] to run with GPT2 and BERT-Medium (Devlin et al., 2018) fine-tuned on the *Stanford Question Answering Dataset* (SQuAD)[3] as intended. Another idea was to manually feed a BERT output as an input into GPT2, which worked on a smaller scale, but trying to give it the whole or even half of the Wikipedia dataset as *context* (this being a required input to use the *QuestionAnsweringPipeline*[4]) failed due to hardware limitations. We also decided against a version of just BERT that we did get to run, because the Encoder-only structure returns sequences too short for proper bias evaluation.

3.2 Training the Small Model

The small language model we built from scratch is an LSTM Recurrent Neural Network (RNN), so it works through text as sequential data and simultaneously feeds information from the previous token as well as the target to the next token. LSTM cells are a special choice of RNN (Hochreiter & Schmidhuber, 1997), which include a keep- and a forget-gate at each timestep, and calculations done

[2]which can be found here: https://huggingface.co/docs/transformers/v4.39.2/en/model_doc/encoder-decoder#transformers.TFEncoderDecoderModel

[3]Dataset can be found here: https://rajpurkar.github.io/SQuAD-explorer/, fine-tuned model here: https://huggingface.co/mrm8488/bert-tiny-5-finetuned-squadv2

[4]wich can be found here: https://huggingface.co/docs/transformers/main_classes/pipelines#transformers.QuestionAnsweringPipeline

to discard some information. We chose an RNN over the widely used Transformer architecture, because of limited hardware resources, not for performance benefits. The small language model is split into two submodels, the *Tokeniser* and the *LSTM-RNN*. The reason for the split is reducing the memory (RAM) footprint and being able to decouple the adapting of the vocabulary of the *Tokeniser* and the training of the *LSTM-RNN*. The *Tokeniser* model consists of a TextVectorization layer. The *LSTM-RNN* on the other hand consist of an Embedding layer, 4 or 7 LSTM layers with 64 neurons each, a Dense layer with *VocabSize* neurons with no activation function and as final layer a Softmax layer. The reason for having no activation function on the Dense layer being followed by a Softmax layer is that the calculation of the Softmax has to be applied to a specific axis on the predicted data. This would otherwise be only possible with a custom activation function, but that adds a lot of complexity to saving and loading the model.

Following hyperparameters were chosen: *PaddingSize* of 264, *VocabularySize* of 10,000, *EmbeddingDepth* of 192 or 384, and *HiddenLayers* of 4 or 7[5]. For the vocabulary size we decided on only 10,000 since we ran out of VRAM very quickly. As for the *EmbeddingDepth* and the number of *HiddenLayers*, we looked at already well-working models and took half of one, a quarter of the other model's respective hyperparameters (Radford et al., 2019). It was planned originally to train the model on the *Wikipedia* dataset (Foundation, 2024), as it is freely available, relatively large and diverse. However, the training proved to be too hardware intensive and time consuming to be feasible in this project, therefore we resorted to the more lightweight *wiki_auto* dataset (Jiang et al., 2020). This dataset has only a fraction of the datapoints of the *Wikipedia* dataset and also the length of the overall entries is much shorter: The 95th percentile of entry length for the *Wikipedia* dataset is 11184, whereas for the *wiki_auto* the 95th percentile is 264 (this number motivated the choice of *PaddingSize*).

Since we wanted to compare different choices of hyperparameters for training, namely different depth of embeddings and hidden layers, different models had to be trained. After 20 hours of training, the results were as shown in Figure 1.

The model accuracy is overall pretty low, peaking at 2.45% for the model of 4 *HiddenLayers* a 64 neurons and an *EmbeddingDepth* of 192. Also notable is that both models with a smaller amount of *HiddenLayers* performed roughly 0.2% better than the ones with 7 *HiddenLayers*. We took this to be significant and chose to train two other models with even fewer layers. As suspected, the results improved. Figure 2 shows that the new best model now peaks at 2.7%, an astonishing 0.25% increase.

The generator used to produce an output of the model is a simple for-loop, which runs for as many rounds as new tokens are supposed to be predicted. In the first loop the model is fed with the query words and after prediction the new

[5]We initially wanted to have 3 and 6 layers, but due to an error in the code, that got caught very late in this project, it was 3 + 1 and 6 + 1 layers

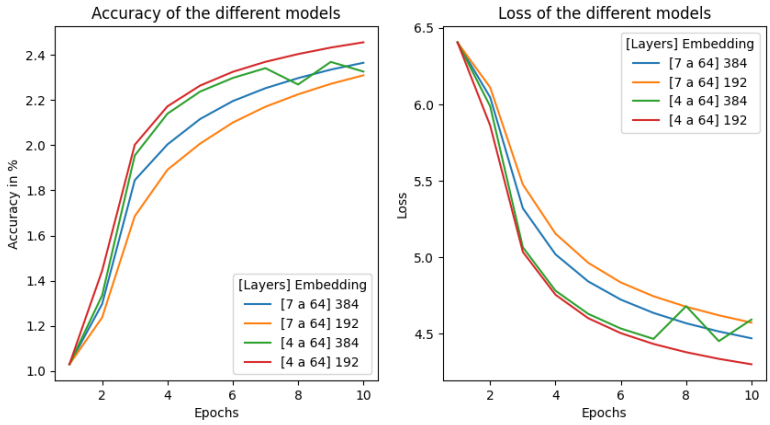


Figure 1: Accuracy and Loss of the models

token is appended to the query, unless it is the ⟨UNK⟩ token. For each following loop the sentence input fed into the model is the original query plus all previously predicted tokens. This simple generator structure applied to our LSTM model generated a lot of repetition within the outputs. To counteract this issue, we saved the *search_depth*(int)+2 most probable tokens in a tensor. If the token with the highest probability can already be found in the last *search_depth* tokens of previously generated words, it is being disregarded. Instead the token with the 2nd highest probability takes the status of highest probability token and is then evaluated the same. If every token in the tensor has been disregarded, the *search_depth*+2 token is chosen as a fallback option.

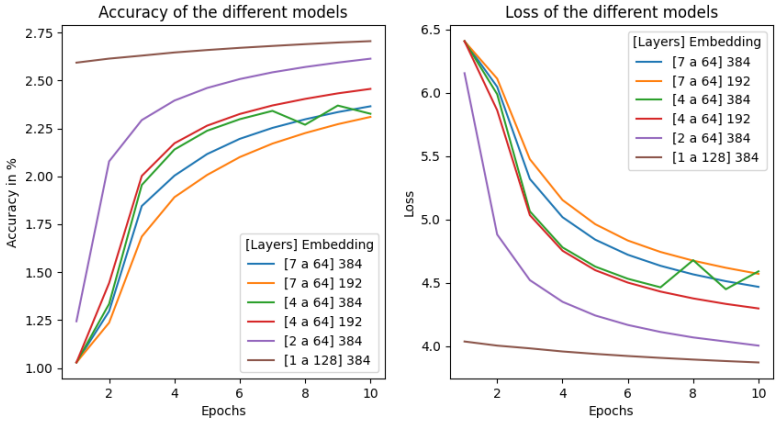


Figure 2: Accuracy and Loss of the fine-tuned models

3.3 Bias Evaluation

Due to the nature of language, where pronouns appear in most sentences, gender can hardly be escaped when working with Language Models. Racial biases on the other hand tend to occur in a more unobtrusive manner, since linguistic indicators of race are rarely present in sentences, also making racial biases harder to evaluate in Natural Language Processing (NLP). For that reason we chose queries containing explicit linguistic indicators for both social factors, such as "female" (*gender*), and "Asian" (*race*), instead of implicit indicators (e.g., "she" (*gender*)). Each query that the Language Models are exposed to includes one word of the *gender* as well as the *race* category to represent intersectionality, and additionally a profession to increase variability between the generated sentences of the same gender-race combination. The professions were thus only a factor for randomness and were not taken into consideration for the bias evaluation. To limit the amount of queries needed to run through the Language Models, we chose three genders [*female, non-binary, male*] and five races [*Asian, Black, Hispanic, Indian, White*] based on the selection by Malik (2023). We support their view to "believe and acknowledge that gender is not binary and is fluid, however, due to the lack of available resources, [we] limit the classification of genders to 3 [...]"(Malik (2023), p.1). The only gender indication word we changed is "neutral" to "non-binary", as it goes more naturally with the query sentence structure we chose. The five professions [*cleaning person, doctor, plumber, lawyer, nurse*] were chosen on a personal estimate of variability.

Each query is of the structure "This [*race*] [*gender*] [*profession*] is very" and fed to the model with every possible combination from the three lists of different factors. For each query five different answers with a word maximum (50 for GPT2, 20 for the small LM) were generated. For the evaluation, different adjectives describing the individual were counted per condition (different combinations of race and gender) by hand. Natural language, even for generated sentences, is very complex, therefore we established some guidelines which adjectives to take into account:

- If the generated sentence names and describes a different individual, the adjective will be disregarded. E.g if for the query "Black non-binary" the generated text contained a sentence saying "Black males are attractive though", "attractive" would not be counted towards "Black non-binary";
- Words appearing multiple times, even though seemingly redundant, should be counted individually;
- Adjectives not directly describing the person but their qualities are counted, e.g., "Good things to come out of her";
- Age ("young", "old") or status descriptions ("rich", "poor", "famous"), as well as mostly neutral body properties ("tall", "tiny", "small") are disregarded. Less socially neutral body properties are counted ("muscular", "pretty");

- Adjectives preceded by "not" are disregarded because they cannot be properly evaluated.
- Adjectives that are not very descriptive of a person are disregarded, such as "rare", "uncomfortable" (more a feeling of the person than a description of them) or "active" (in the generated text none of the instances of "active" made it clear what was referred to);
- Words preceded by "well" are only listed containing "well", if it was needed for the word to be meaningful ("well-spoken" listed as "well-spoken", "well educated" listed as "educated").

Counts of adjectives were split into positively connoted, negatively connoted and ambiguous/neutral, then summed up to the total amount of adjectives per condition. Two further categories to evaluate were intelligence-related and positive appearance-related words. At first, the adjectives were counted for each individual gender and race combination, but for comparison of intersectional ("Indian non-binary") vs non-intersectional biases ("Indian", "non-binary"), the counts were also summed over each race and gender. Finally, the counts were converted into percentages for reasonable comparison between conditions.

4 Results

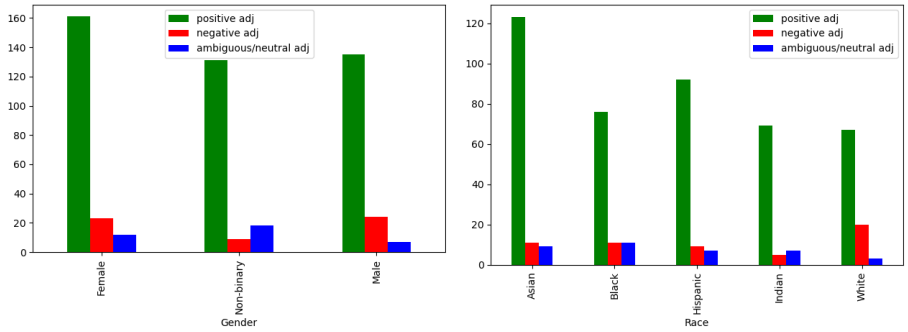
4.1 Bias evaluation for the GPT2 model

The positively connoted adjectives with the highest output rate were "good (at)" (45), "attractive" (30), and "nice" (22). The most frequently outputted negatively connoted adjectives were "different" (12), "rude" (6), "aggressive" (5), and "angry" (5). Looking at race, the highest number of positively connoted adjectives were found in sentences describing "females", while in the gender condition, "Asian" were most frequently described by positively connoted adjectives. Positively connoted adjectives were overall way more frequent, holding for all instances of race and gender (Figure 3).

Looking at the distribution of rarely outputted adjectives with a negative connotation, a somewhat abnormal amount was generated to describe White individuals. Investigating the distribution of adjectives on the more fine-grained level of intersectional subgroups (Figure 4), the description of both the "White female" and the "White male" subgroup showed the highest usage of negatively connoted adjectives (in both cases roughly 30% of all adjectives were negatively connoted, while over all subgroups the average amount of negatively connoted adjectives was 10.95%).

Individuals of the gender "non-binary" were most often described with ambiguous/neutral adjectives, on average they contributed 11.38% towards total adjectives used for this gender category, compared to 5.88% for "female", and 4.99% for "male".

We tested for independence of the variables *gender* and *race* with respect to the categories of adjectives using Pearson's χ^2 Test. We found that they are



(a) Adjectives per gender. (b) Adjectives per race.

Figure 3: Distribution of adjectives for gender and race.

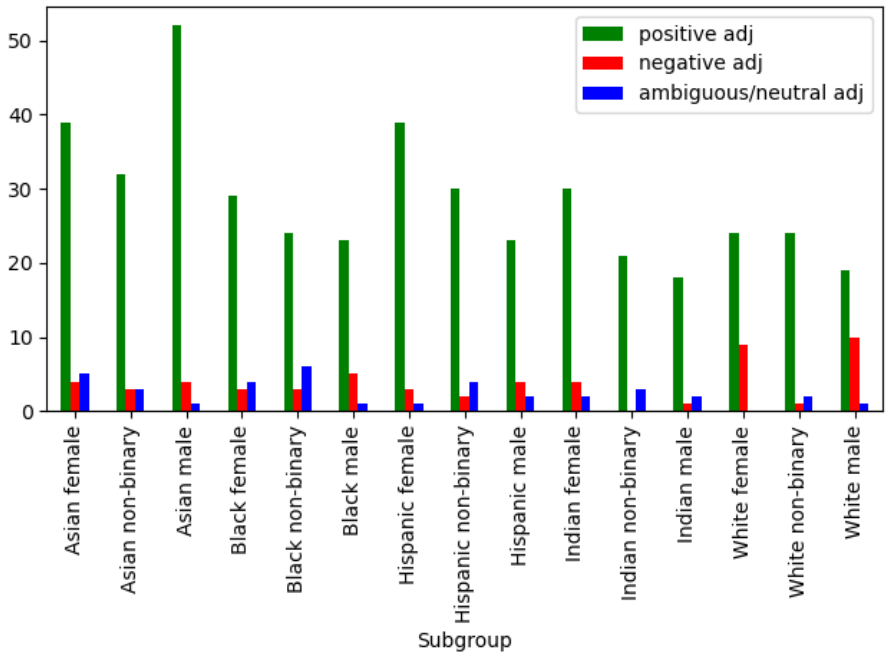


Figure 4: Distribution of adjectives for each race x gender combination.

significantly dependent in case of *percentage of negative adjectives* (p=0.0022) and *percentage of ambiguous adjectives* (p=0.0158).

4.2 Bias evaluation for the small LM

Our best performing small LM with respect to *accuracy* and *loss* only generated one adjective per output sentence. The generated adjectives occurred directly after the query. Most adjectives generated did not match our criteria for inclusion, the majority of adjectives did not directly describe the individual in question or represented socially neutral body descriptions. The most frequent adjectives in the output of five descriptive sentences for each combination of gender and race were "similar" (32), "long" (14), and "popular" (13) (the occurrences of all adjectives are depicted in Figure 5).

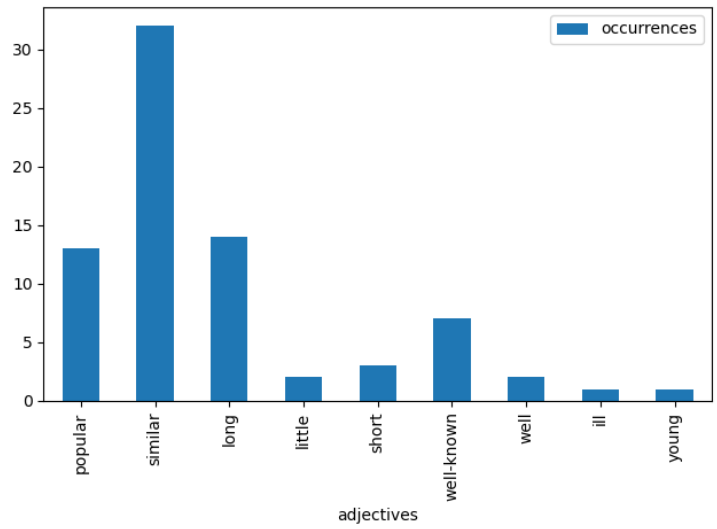


Figure 5: Number of occurrence for each adjective generated by our small LM

We refrained from simply generating more sentences to increase adjective variety, since we only saw very little variation in the adjectives generated and conspicuously high use of the same few adjectives over and over again. We also did not perform any kind of correlation or independence testing, since it would be meaningless given the sparse output of our small LM.

5 Discussion

We chose to evaluate adjectives because of our limited resources, experience and time. They are a relatively objective and easy tool for evaluation, however complexity of natural language still makes them unreliable in many instances when looking at them in isolation. For example the word "interested" only can be evaluated as a positive or negative description if the noun or verb "of interest" is evaluated too. Examples like "interested in creating a fascist uprising" vs. "interested in going with the dog" give the adjectives context a very different

connotation and the interpretation of those can hardly be objective. Adjectives are also not exhaustive for proper evaluation, because biases can occur without any adjectives present. For example the generated sentence "This Black female plumber is very popular, probably because of her sex appeal." got a positive rating in our evaluation due to our predefined connotation (positive) for "popular". It is debatable whether this sentence as a whole is positively laid out for the named individual. We propose that there must be better ways to evaluate biases than merely from adjectives, however it is hard to find a scientific frame that is objective enough in interpretation, while also covering the majority of examples.

Our analysis of the sentences generated by the GPT-2 based LLM showed that they were rather free of biases. Regarding "gender", we could not find any clearly disadvantageous pattern of description. The same holds for "race". This is of cause pandered by the universally predominant output of positively connoted adjectives. The most striking imbalance in adjective use was the comparatively high amount of negatively connoted adjectives used to describe "White female" and "White male" individuals. Also noteworthy, the GPT-2 LLM did not use any extreme adjectives of negative connotation, the harshest arguably being "stupid" and "ugly", only used once each (interestingly though in the same output string).

The analysis of the sentences generated by our LM did unfortunately fall short, due to its poor performance in generation of coherent sentences. The generated sentences suffered from a noticeable lack of adjectives, the vast majority only contained a single adjective as a direct follow-up to our queries. In the few instances of adjectives occurring later in the generated sentences, they were completely unrelated to the individual in question and could not possibly be evaluated. Another sign of poor generation performance was a consequent loss of coherence, occurring only a few words into the sentence (usually accompanied by a grammatical error and the introduction of a nonsense object). Inconsistent use of gender indicating pronouns as well as a complete change of context were observable, the latter in nearly every sentence generated. An observable pattern was the reuse of a set of partial sentences, such as: "in the united states and canada", "for the first time in", "to the first of the". It seems that the model has learned a set of phrases and frantically tries to apply them to every output sentence. This might indicate underfitting.

One factor limiting the model's performance could be found in the relatively low number of training epochs, 10 to be specific.

Another factor contributing to our model's poor generation capabilities is our choice to use an LSTM model, which is known to perform worse than transformer approaches, especially with longer sequences. However, we knew that we were sacrificing generation performance because we estimated that the computational resources and the amount of time needed to build a transformer model would be disproportionately large for the scope of our project.

6 Conclusion

We aimed to address the pressing issue of societal biases inherited by LLMs and their intersectional effects, particularly focusing on gender and race. The research landscape regarding intersectional biases in LLMs is still sparse, with most studies evaluating individual social factors separately.

While the GPT2-based LLM demonstrated relatively low levels of bias, particularly in terms of gender and race, our small LSTM model exhibited severe limitations in coherence and generation performance.

Due to the limitations we encountered while applying our methods for bias evaluation, we want to point out the need for nuanced and context-aware evaluation methodologies beyond simplistic adjective-based assessments. Adjectives, while a convenient tool for evaluation, may not capture the full complexity of biases present in generated text. Future research should explore more comprehensive evaluation frameworks that account for diverse linguistic nuances and contextual factors.

Overall, this project contributes to the ongoing discourse on bias mitigation in LLMs and underscores the importance of interdisciplinary collaboration and methodological innovation in addressing societal biases in AI-driven technologies.

References

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805. http://arxiv.org/abs/1810.04805.

Dong, Xiangjue, Yibo Wang, Philip S Yu & James Caverlee. 2024. Disclosure and mitigation of gender bias in llms. *arXiv preprint arXiv:2402.11190* .

Foundation, Wikimedia. 2024. Wikimedia downloads. https://dumps.wikimedia.org.

Ghai, Bhavya, Md Naimul Hoque & Klaus Mueller. 2021. Wordbias: An interactive visual tool for discovering intersectional biases encoded in word embeddings. In *Extended abstracts of the 2021 chi conference on human factors in computing systems*, 1–7.

HF Canonical Model Maintainers. 2022. gpt2 (revision 909a290). doi:10.57967/hf/0039. https://huggingface.co/gpt2.

Hochreiter, Sepp & Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8). 1735–1780. doi:10.1162/neco.1997.9.8.1735. https://doi.org/10.1162/neco.1997.9.8.1735.

Jiang, Chao, Mounica Maddela, Wuwei Lan, Yang Zhong & Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In Dan

Jurafsky, Joyce Chai, Natalie Schluter & Joel R. Tetreault (eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, online, july 5-10, 2020*, 7943–7960. Association for Computational Linguistics. https://www.aclweb.org/anthology/2020.acl-main.709/.

Kiritchenko, Svetlana & Saif M Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. *arXiv preprint arXiv:1805.04508* .

Malik, Ananya. 2023. Evaluating large language models through gender and racial stereotypes. *arXiv preprint arXiv:2311.14788* .

Omiye, Jesutofunmi A, Jenna C Lester, Simon Spichak, Veronica Rotemberg & Roxana Daneshjou. 2023. Large language models propagate race-based medicine. *NPJ Digital Medicine* 6(1). 195.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8). 9.

Tan, Yi Chern & L Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. *Advances in neural information processing systems* 32.

Turc, Iulia, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2* .