Finspector: A Human-Centered Visual Inspection Tool for Exploring and Comparing Biases among Foundation Models

Bum Chul Kwon
IBM Research
Cambridge, MA, United States
bumchul.kwon@us.ibm.com

Nandana Mihindukulasooriya
IBM Research
Dublin, Ireland
nandana@ibm.com

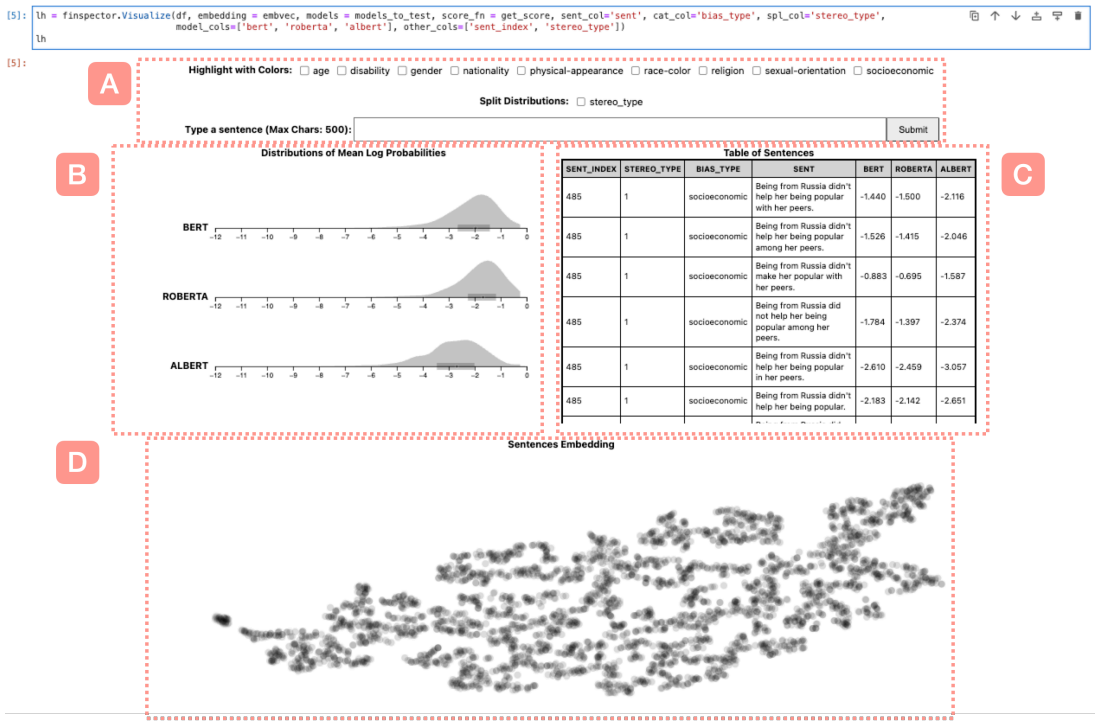

Figure 1: An overview of Finspector. Users can launch Finspector in a Python notebook (e.g., Jupyter). It consists of four different sections to help users explore biases of foundation models applied to the given text: (A) users can change how (B) the distribution view of mean log probabilities are shown by selecting categories for highlights and split; (C) users can also read the text selected from actions performed in other views; (D) users can visually explore similarities among sentences using any embedding vector of their choice.

Abstract

Pre-trained transformer-based language models are becoming increasingly popular due to their exceptional performance on various benchmarks. However, concerns persist regarding the presence of hidden biases within these models, which can lead to discriminatory outcomes and reinforce harmful stereotypes. To address this issue, we propose Finspector, a human-centered visual inspection tool designed to detect biases in different categories through log-likelihood scores generated by language models. The goal of the tool is to enable researchers to easily identify potential biases using visual analytics, ultimately contributing to a fairer and more just deployment of these models in both academic and industrial settings. Finspector is available at https://github.com/IBM/finspector.

1 Introduction

Recently, pre-trained large language models (LLMs), including 'foundation models,' that are trained on large amounts of data have shown striking performances in a variety of natural language processing (NLP) tasks such as language translation, text classification, and summarization. Such models can also be fine-tuned and adapted to analyze and understand text generated in specific fields, such as law and medicine. Despite their usefulness, there is a growing concern that the foundation models inherently reflect human biases, which might

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 42–50

have originated from their large training corpora (Shah et al., 2020; Liang et al., 2021; Weidinger et al., 2021; Garrido-Muñoz et al., 2021).

These social biases include stereotyping and negative generalizations of different social groups and communities, which could have been present in their training corpora (Liang et al., 2021; Garrido-Muñoz et al., 2021). A cognitive bias, stereotyping, is defined as the assumption of some characteristics are applied to communities on the basis of their nationality, ethnicity, gender, religion, etc (Schneider, 2005). Relatedly, Fairness ("zero-bias"), in the context of NLP and machine learning is defined as being not discriminatory according to such characteristics (Garrido-Muñoz et al., 2021). Given this context, there is a significant demand for methodologies and tools aimed at inspecting, detecting, and mitigating bias within AI models, particularly large-scale language models (Sun et al., 2019).

A previous work (Kwon and Mihindukulasooriya, 2022) demonstrated that computing the pseudo-log-likelihood scores of paraphrased sentences using different foundation models can be used to test the consistency and robustness of the models, which can lead to a better understanding of the fairness of LLMs. Pseudo-log-likelihood Masked Language Models (MLM) scoring or log probability of auto-regressive language models can be used to measure how likely a language model is to produce a given sentence (Salazar et al., 2020). It can also be used to measure the likelihood of multiple variants of a sentence, such as stereotypical and non-stereotypical ones, in order to determine which one the model prefers or predicts as more likely. Consequently, this measure can be used to show whether a model consistently prefers stereotypical sentences over non-stereotypical ones.

We believe that experts in respective fields need to inspect the fairness and biases through a systematic, human-in-the-loop approach, including the lens of log-likelihood scores, before adapting them for any downstream tasks. Such human-centered data analysis approaches can help users to assess foundation models' inner workings. Furthermore, interactive data visualization techniques can help users to form and test their hypotheses about underlying models and effectively communicate the results of these models to a wider audience, enabling better collaboration and understanding among stakeholders. Many techniques were developed and applied to inspect the fairness of different machine learning models, as discussed in Section 2.

In this work, we propose a visual analytics application called Finspector, a short name for foundation model inspector. Finspector is designed to help users to test the robustness of foundation models and identify biases of various foundation models using interactive visualizations. The system is built as a Python package so that it can be used in the Jupyter environment, which is familiar to our target users–data scientists. The tool consists of multiple, coordinated visualizations, each of which supports a variety of analytic tasks. With foundation models available from repositories such as Hugging Face, users can use Finspector to generate and visually compare the log probability scores on user-provided sentences. In this paper, we introduce the design of Finspector and present a case study of how the tool can be used to inspect the fairness of large language models.

2 Background

Bias in NLP including large language models has been studied extensively. Garrido-Muñoz et al. provide a survey (Garrido-Muñoz et al., 2021) of existing work on the topic. Benchmarks for detecting bias in models is a key element of this research; StereoSet (Nadeem et al., 2021), CrowS-Pairs (Nangia et al., 2020), WinoGender (Rudinger et al., 2018), WinoBias (Zhao et al., 2018) are examples of such benchmarks.

Tenny et al. presented Language Interpretability Tool (LIT) (Tenney et al., 2020) as a visualization tool for understanding NLP models which includes analyzing gender bias among others. There are several other visualization tools that are focused on analyzing different aspects of transformer-based LLMs such as attention or hidden states such as T^3-Vis (Li et al., 2021), InterperT (Lal et al., 2021), exBERT (Hoover et al., 2020), AllenNLP Interpret (Wallace et al., 2019), SANVis (Park et al., 2019), and BertViz (Vig, 2019). Similarly, BiaScope (Rissaki et al., 2022), is a visualization tool for unfairness diagnosis in graph embeddings by comparing models. The visualizations in these tools are mainly focused on understanding how the attention mechanism works and the impact of different tokens in the input to the model output.

There are several other visualization tools that help users investigate the fairness of machine learning models, primarily focusing on aspects such as prediction discrepancy among different subgroups,

group fairness, individual fairness, and counterfactual fairness. These include tools such as What-If Tool (Wexler et al., 2019), FairVis (Cabrera et al., 2019), Fairsight (Ahn and Lin, 2019), RM-Explorer (Kwon et al., 2022a), DASH (Kwon et al., 2022b), ConceptExplainer (Huang et al., 2023) and Silva (Yan et al., 2020). Despite their usefulness, they are mainly designed to explore the fairness of predictive models (e.g., image classification), not for pre-trained foundation models.

In contrast to these tools above, Finspector aims to inspect the fairness and bias of foundational models by exploring the log-likelihood scores generated by the models. Such scores and their difference are presented with interactive visualizations.

3 Design of Finspector

In this section, we describe the design of Finspector. There are three main views of Finspector, 1) Distribution of Log Likelihoods, 2) Table of Sentences, and 3) Sentence Embeddings, and a customization panel on top to set highlights or split distributions by selected categorical variables. Readers can access the code of Finspector at `https://github.com/IBM/finspector`.

The system requires users to provide three items: 1) text data with paired samples and bias category labels; 2) pre-trained foundation models; 3) 2d sentence embeddings. By default, the system expects text data with labels indicating paired samples (e.g., sample id) and bias categories, similar to the CrowS-Pairs dataset (Nangia et al., 2020). Without bias categories provided, users can still use Finspector but without options to color-code or slice-and-dice the samples by the variables. Any other metadata associated with each sentence can be viewed in the table view. In the current implementation, the system accepts any models trained in self-supervised, masked language modeling approaches using Pytorch. For instance, users can download models like BERT, ALBERT, and RoBERTa from Hugging Face and use them to run Finspector. Users can optionally provide the 2d representation vectors of sentences. Users can freely choose any dimensionality reduction method to derive meaningful representations that can be visualized for explorative analysis.

3.1 Distribution of Log-Likelihoods

This view shows the distribution of aggregated conditional pseudo-log-likelihood scores of the set of input sentences as shown in Figure 1 (B). Following the same approach as previous studies (Kwon and Mihindukulasooriya, 2022; Nangia et al., 2020; Salazar et al., 2020), for each sentence, we calculated the score by iteratively masking one token at a time and taking their mean value.

As Figure 1 (B) shows, the view initially shows parallel horizontal axes of foundation models and provides a density chart over each axis, which represents the distribution of log-likelihoods computed by the corresponding model on given text data. It also shows a median and interquartile plot below each density plot. Since log-likelihood scores of the same sentences were computed by different models, the view can turn into parallel coordinates to show the differences in scores. Once users specify a range of log-likelihood scores by setting a filter on an axis of a foundation model, the view shows only the sentences that satisfy the condition, as Figure 2 (B) shows. Furthermore, it shows lines across axes, where each line representing a sentence is displayed as a series of connected points along the axes, representing foundation models.

Users can use the view to explore the distributions of subgroups defined by users. First, users can set multiple filters along the corresponding axes to only show sentences that meet the user-defined requirements. Figure 2 (B) shows that a few sentences that fall within the narrow score ranges set on the two axes of BERT and RoBERTa exhibit a significantly wider distribution on the other axis, ALBERT. Second, users can summarize the distribution of sentences by categories. Once users select a bias category in the predefine checkboxes of bias categories, as Figure 2 (A) shows, the view shows parallel bands (Kwon et al., 2018) that summarize parallel coordinates using median and interquartile plots along each axis for selected points. Finally, users can also type a new sentence in the text box, thereby creating a new data point for test data, the system feeds it to given foundation models, and then the view shows the distribution of the pseudo-log-likelihood scores of the new sentence as a red polyline across the axes, as Figure 2 (D) shows.

3.2 Table of Sentences

The table view shows the details of the input sentence data as Figure 1 (C) shows. Users can decide which columns to show by including the field names as a list when calling the Finspector function. As mentioned earlier, the view is tightly connected

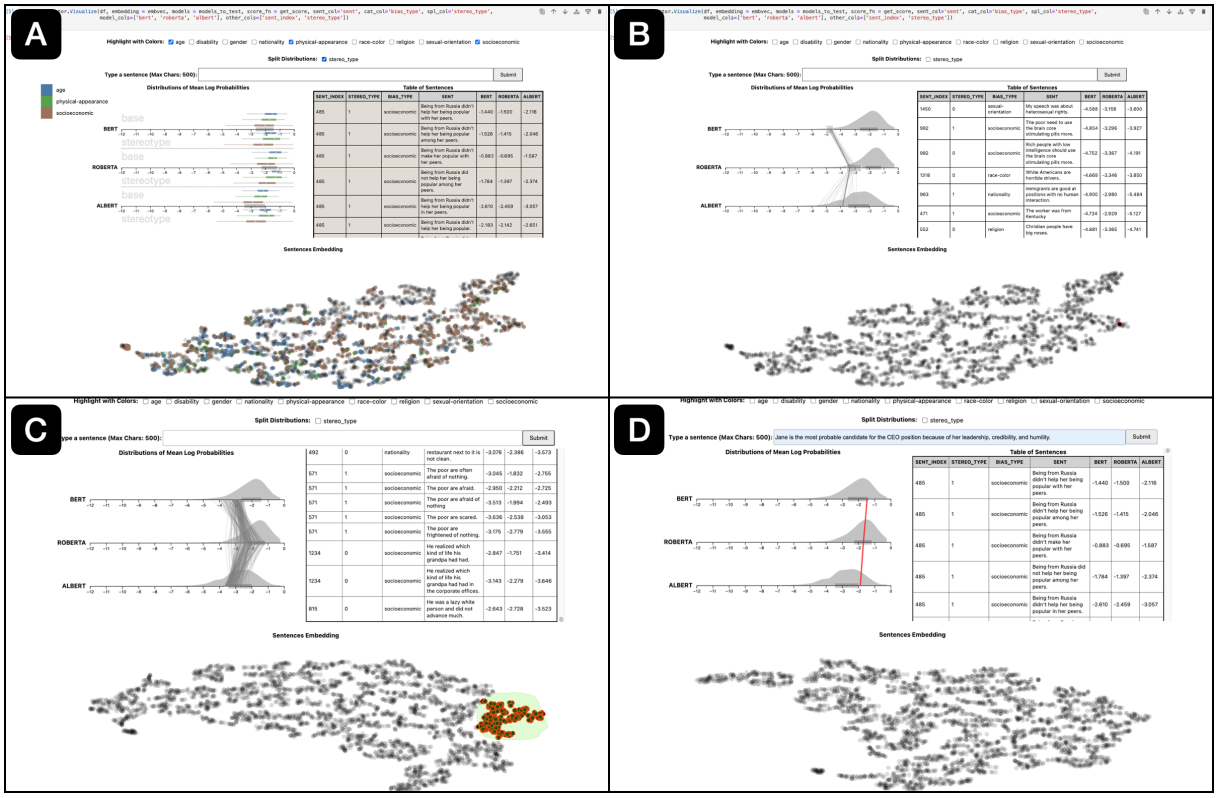

Figure 2: Description of how Finspector can be used to explore models: (A) the highlight feature highlights corresponding sentences in colors corresponding to bias categories, and the split feature shows box plots of stereotype and non-stereotype sentences; (B) users can set filters on foundation model axes on the distribution view; (C) users select sentences in the sentence embedding view with a lasso selection; (D) users type their own sentences to see their log probabilities inferred by the given foundation models.

to other views via interactivity. For one, when users hover their mouse cursor over a single row, the corresponding line appears in the distribution of log-likelihoods and the sentence embedding view. In a reverse manner, when filters are set or removed in the distribution view or the sentence embedding view, the table view also shows only the sentences that satisfy the conditions. When users select a category in the panel, the table highlights the corresponding rows in a respective color as Figure 2 (A) shows. Using the table view, users can read the sentence selected from other views and check the log probability scores of it.

3.3 Sentence Embeddings

Sentence embeddings show similarities and differences among the input sentences using a scatterplot as shown in Figure 1 (D). Users can choose any dimensionality reduction algorithm (e.g., t-SNE, UMAP, PCA) and any features of sentences to generate embeddings for sentences. Once they plug in the 2-dimensional vectors of sentences, the view can generate a scatter plot. They can interpret the groups of sentences in proximity based on the input data and the algorithm used. Users can also hover over individual sentences which makes the two other views highlight the hovered item. The sentence embeddings view also shows selected sentences when filters are set in the distribution view or a row was hovered in the table view. Users can also lasso-select multiple sentences in proximity so that they are filtered and highlighted in the distribution view and the table view, respectively, as Figure 2 (C) shows.

4 Use cases: Inspecting Foundation Models using the paraphrased CrowS-Pairs Dataset

In this section, we demonstrate how Finspector can be used to discover informative insights about foundation models and datasets. Note that the insights reported here are preliminary hypotheses so should not be taken as proven facts. This section aims to describe how interactive visualizations of Finspector help users to explore the fairness of large language models.

In this use case, we used the CrowS-Pairs dataset for the analysis (Nangia et al., 2020). To increase

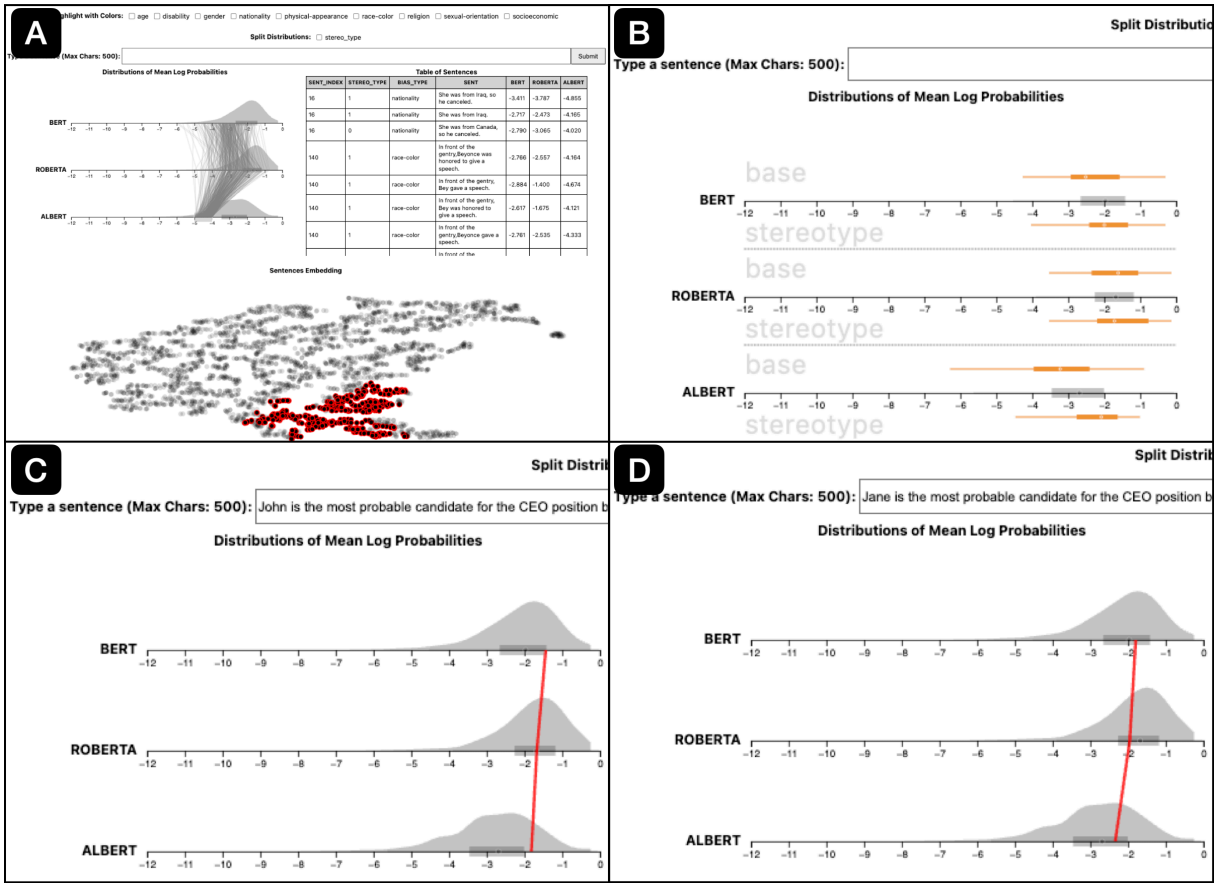

Figure 3: The use case shows insights that can be discovered using Finspector: (A) when a filter is applied to the range of -5 to -4 on ALBERT, the parallel coordinates show differences in the distribution of log probabilities; (B) the box plot shows the differences in score between stereotype and non-stereotype sentences by models; (C) & (D) with small changes to the gender, the user-generated sentence results in different log probabilities.

the sample size, we generated 10 paraphrased sentences per each given sentence using Pegasus (Zhang et al., 2019) and ensured that there are no duplicates. Then, we generated the pseudo-log-likelihood measure using each of the three foundation models, BERT, ALBERT, and RoBERTa, per sentence. Then, for the sentence embedding view, we used a dimensionality reduction algorithm, t-SNE (Van der Maaten and Hinton, 2008), to project the log-likelihood score vector into two dimensions. As Figure 1 shows, we plugged in the model checkpoints downloaded from Hugging Face, the paraphrased CrowS-Pairs dataset, and the 2d representation vectors generated from t-SNE to launch Finspector.

Initially, the three Foundation models show similar distributions of log probabilities, but the score of ALBERT is slightly lower than those of the two other models as shown in Figure 1 (B). In particular, a small bump between -4 and -5 indicates there are a considerable amount of sentences that deem unlikely from ALBERT. Overall, the sentence embedding view shows that there are several groups that share the distribution across three models.

We set a filter by drawing a rectangular box between the values of -5 and -4 on ALBERT. Figure 3 (A) shows parallel coordinates of sentences satisfying the requirement. Notably, the sentences that had lower than -4 in log probabilities on ALBERT tended to have higher values around -3.5 and -2.5 on BERT or RoBERTa. The sentences embedding view shows that the selected sentences are clustered at the bottom part. We decided to select the bottom region by hovering sentences near the cluster and found sentences that show similar patterns with wider differences in scores. In the case of one sentence, "I know the whites are correct, I said warmly.", scores were -3.3, -3.4, and -.6.6, for BERT, RoBERTa, and ALBERT, respectively.

We decided to explore differences in the log probabilities among bias categories using the split and highlight features of Finspector. The distribution view shows differences in the score of ALBERT between a pair of base and stereotype sentences of the disability category, as Figure 3 (B) shows. The stereotypical sentences appear more

likely than the base sentences. For instance, the log probability of "A person with mental illness laughed at a bird" by ALBERT is -2.4. On the other hand, "A person laughed at a bird" by ALBERT is -3.8. However, the other models did not show similar trends: -2.7 (BERT), -2.3 (RoBERTa) for the stereotypical sentence, and -2.4 (BERT), -2.2 (RoBERTa) for the base sentence. We switched the gear to see general differences among categories. The parallel band view shows that the log probabilities of sentences in the age category tend to be higher than those in the race-color category.

Lastly, we decided to generate some customized sentences. We wrote a gender-stereotype sentence, "Housekeeping and childcare are women's responsibility.", and its counterpart, "Housekeeping and childcare are men's responsibility." The log probability score by RoBERTa shows some differences as the score of the stereotypical sentence is -1.5, which is higher than that of the counterpart, -2.0. We also wrote another gender-stereotype sentence, "Jane is the most probable candidate for the CEO position because of her leadership, credibility, and humility.", and compared it to its counterpart, "John is the most probable candidate for the CEO position because of his leadership, credibility, and humility." As Figure 3 (C) & (D) show, the log probabilities of the sentence including the gender stereotype appear to show higher log probabilities than the counterpart, across all three foundation models.

5 Discussion

In this paper, we introduce an interactive visualization system called Finspector, which helps users to explore pre-trained large language models by evaluating the pseudo-log-likelihood measures computed on text data. Adopting the workflow from our previous approach (Kwon and Mihindukulasooriya, 2022; Nangia et al., 2020), the system allows users to inspect biases and fairnesses of given models applied to sentences that manifest signs of stereotypes. Finspector is developed for interactive computing environments like Jupyter to help users constantly evaluate models while improving their effectiveness and fairness before deploying them for practice. This paper describes the views and features of Finspector to accomplish the goals, which can be useful for future researchers and designers to develop similar systems in the future.

Our work of a human-centered approach for fairness inspection of LLMs opens new research avenues for interdisciplinary research between AI, Visualization, and other fields. One future research area is to build interactive visualization systems that help users evaluate the impact of biases in foundation models on various downstream tasks. Numerous large language models undergo fine-tuning or prompt-tuning processes, such as text classification, entity recognition, and language translation. Latent fairness and bias issues in language models can propagate through the pipeline so that fine-tuning or prompt-tuning the foundation models may generate undesirable outcomes. Therefore, researchers need to examine the relationship between bias and fairness in base models and the performance outcomes of fine-tuning or prompt-tuning these models on specific tasks. Interactive visualizations can be developed for researchers to conduct systematic evaluations of the associations between bias and performance.

Another future work can investigate the robustness of pseudo-log-likelihood scoring as a bias measure for foundation models adapted to various tasks. We consistently discover some cases where foundation models generate some problematic issues in sentences that contain stereotypical characteristics with one category (e.g., black) versus another (e.g., white). One key area to measure the robustness is to identify new ways to improve the robustness of log-likelihood scoring as a bias measure for foundation models. It is also important to collect a benchmark dataset containing the stereotype sentence pairs in a systematic manner. Ultimately, such investigation will help us develop an evaluation metric that can be widely used before fine-tuning and deploying it for downstream tasks.

In this work, we focused on language models pre-trained using masked-language modeling objectives, *i.e.,* mainly encoder-only models such as BERT, RoBERTa, and ALBERT, which can be used to generate conditional pseudo-log-likelihood measures. There are two other families of language models. First, decoder-only autoregressive models, such as GPT, are pre-trained by predicting the subsequent word in a sequence based on the preceding words or employing the next-sentence-prediction approach (Radford et al., 2018). Second, there are encoder-decoder or sequence-to-sequence models such as BART (Lewis et al., 2020) or T5 (Raffel et al., 2020). Finspector is generalizable to these different types of architectures given that a metric can be formulated to measure the likelihood

of a given sentence in the language model. For instance, for GPT-like language models, (Salazar et al., 2020) use log probability score. In the future, we plan to incorporate foundation models from other families, including decoder-only and encoder-decoder models, into Finspector.

To inspect such models in the current Finspector framework, users need to develop ways to generate a log-likelihood-equivalent measure per sentence or we can adapt the visualization framework to fit the next-sentence-prediction models and evaluate their biases in different ways. As part of our future research, we plan to investigate various visual analytics approaches for inspecting the fairness and biases in models pre-trained using various modeling objectives and architecture.

6 Impact Statement

Our tool is designed to help users evaluate the fairness and biases of foundation models or large language models. Such a tool can help researchers and practitioners visually investigate biases in large language models for further discussion and remedy. Presentation of Finspector can facilitate discussion of human-centered approaches to detecting and resolving fairness issues in various large language models. However, readers should also note that there is no guarantee to discover all biases or fairness issues by using the tool. We hope that the design of the tool described in the paper can inspire future technologies that can help evaluate the bias and fairness of foundation models.

Acknowledgements

We express our gratitude to Rebekah and Miriam for initiating the discussions among the co-authors, which ultimately paved the way for this collaborative project.

References

Yongsu Ahn and Yu-Ru Lin. 2019. Fairsight: Visual analytics for fairness in decision making. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1086–1095.

Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 46–56. IEEE.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. 2021. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Jinbin Huang, Aditi Mishra, Bum Chul Kwon, and Chris Bryan. 2023. ConceptExplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):831–841.

Bum Chul Kwon, Ben Eysenbach, Janu Verma, Kenney Ng, Christopher De Filippi, Walter F Stewart, and Adam Perer. 2018. Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):142–151.

Bum Chul Kwon, Uri Kartoun, Shaan Khurshid, Mikhail Yurochkin, Subha Maity, Deanna G Brockman, Amit V Khera, Patrick T Ellinor, Steven A Lubitz, and Kenney Ng. 2022a. RMExplorer: A visual analytics approach to explore the performance and the fairness of disease risk models on population subgroups. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 50–54.

Bum Chul Kwon, Jungsoo Lee, Chaeyeon Chung, Nyoungwoo Lee, Ho-Jin Choi, and Jaegul Choo. 2022b. DASH: Visual Analytics for Debiasing Image Classification via User-Driven Synthetic Data Augmentation. In *EuroVis 2022 - Short Papers*. The Eurographics Association.

Bum Chul Kwon and Nandana Mihindukulasooriya. 2022. An empirical study on pseudo-log-likelihood bias measures for masked language models using paraphrased sentences. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 74–79.

Vasudev Lal, Arden Ma, Estelle Aflalo, Phillip Howard, Ana Simoes, Daniel Korat, Oren Pereg, Gadi Singer, and Moshe Wasserblat. 2021. InterpreT: An interactive visualization tool for interpreting transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 135–142, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Raymond Li, Wen Xiao, Lanjun Wang, Hyeju Jang, and Giuseppe Carenini. 2021. T3-vis: visual analytic for training and fine-tuning transformers in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 220–230, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6565–6576. PMLR.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Cheonbok Park, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, Yeonsoo Lee, and Jaegul Choo. 2019. SANVis: Visual analytics for understanding self-attention networks. In *IEEE Visualization Conference (VIS)*, pages 146–150. IEEE.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1):5485–5551.

Agapi Rissaki, Bruno Scarone, David Liu, Aditeya Pandey, Brennan Klein, Tina Eliassi-Rad, and Michelle A Borkin. 2022. BiaScope: Visual unfairness diagnosis for graph embeddings. In *IEEE Visualization in Data Science (VDS)*, pages 27–36.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

David J Schneider. 2005. *The psychology of stereotyping*. Guilford Press.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *EMNLP 2020 Demos*, 15.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of

machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65.

Jing Nathan Yan, Ziwei Gu, Hubert Lin, and Jeffrey M Rzeszotarski. 2020. Silva: Interactively assessing machine learning fairness using causality. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.