ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Linearity of Sensitive Concepts in Language Models

Sarah Schröder[1] and Valerie Vaquet[1] and Barbara Hammer[1] *

1- Bielefeld University - AG Machine Learning
Inspiration 1, 33615 Bielefeld - Germany

Abstract. Identifying how sensitive attributes like ethnicity are encoded in language models can yield valuable insights in terms of fairness. This knowledge could enhance explanations of model decisions, aid in mitigating social biases, or indicate under-represented minorities. Based on the literature on fairness and explainable AI, it should be possible to learn sensitive attributes such as gender or ethnicity with linear methods. Unfortunately, there are not many papers on the intersection of concept learning and fairness. On the other hand, too many fairness papers restrict their evaluation to binary gender and do not consider more complex test cases. So, it is not entirely clear whether all sensitive attributes and identity groups are encoded linearly in language models. Hence, we evaluate this question on a broad selection of identity groups, datasets, and language models.

1 Introduction

Research has shown that language models (LM) are prone to reproduce social biases and stereotypes from their training data and thus can exacerbate existing problems. Hence, fairness must be addressed when developing models that have an impact on people's lives. One common approach to mitigate bias in language models is the removal of sensitive attributes from the model's embeddings or the entire model [1, 2]. However, knowledge and sensitivity to marginalized groups in a model might also be beneficial to produce more fair and generally better outcomes [3]. For instance, dialect can be an important context while also revealing aspects of the user's identity. In addition, under-representation of marginalized groups in language models could be harmful as well. In this context, we explore the usability concept learning [4, 5, 6] to identify representations of sensitive attributes in language models. We will refer to this as learning 'sensitive concepts'. Learning such concepts can improve model transparency or even yield pseudo-labels for identity groups when labeling the entire training data is too costly. Earlier work [7] has shown that existing concept learning methods achieved limited correlations with ground truth labels when looking beyond simple cases (detecting binary gender based on pronoun usage). Possible reasons for these shortcomings are: (i) The LMs do not represent the sensitive concepts well enough, (ii) the quality of ground truth labels is too inconsistent to yield reliable results, and (iii) the tested methods (all linear) are not complex enough to capture how the concepts are represented by the LMs. This paper will mainly focus on the third hypothesis while shedding some light on the others.

*Funding in the scope of the BMFTR project KI Akademie OWL under grant agreement No. 16IS24057A is gratefully acknowledged.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

2 Foundations and Related Work

Related works typically use linear methods to extract semantically meaningful concepts from deep language or vision models [4, 5, 6]. Since LMs often only add a linear layer on top of the transformer for classification, this seems reasonable. However, the literature focuses on the most important concepts for some tasks which typically do not involve sensitive attributes. In the fairness literature, it is common practice to use linear methods such as bias subspaces or gender directions to capture the encoding of sensitive attributes [8, 9]. In terms of binary gender, the literature agrees that one linear feature suffices to capture the concept [8], but it is not clear whether this holds in general [10].

3 Setup

To obtain a comprehensive picture, we consider embeddings from a large variety of language models and datasets for our analysis. For more details on the datasets and training setup, refer to our implementation on Github[1].

Language Models and Embeddings We use the embeddings (last hidden states) of pretrained Language Models to learn sensitive concepts. While finetuning to a downstream dataset is a common practice for smaller models like BERT, finetuning entire LLMs is far more unlikely, and those models might have already seen most of the publicly available data. We use mean-pooled embeddings (computing the mean of token embeddings) which worked best with pretrained models. We use BERT, Roberta, Deberta, XLNET, Deepseek, Llama, Opt, Albert and Pythia models (2-4 models of different size, up to 7B parameters) from Huggingface and the OpenAI embedding models (via API, models not public).

Datasets We use (i) handcrafted datasets of stereotypical statements (StereoSet [11], CrowSPairs [12], WinoQueer [13]) where it was feasible to ensure good label quality by reviewing the data and adjusting labels to fit our task, (ii) automatically labeled real-world datasets (BIOS [14], TwitterAAE [15]) and real-world hate-speech detection datasets (SBIC [16], Jigsaw [17], ImplicitHate [18]) annotated by MTurk workers. Labeling protocols of the real-world datasets differ: In BIOS binary gender is labeled based on pronoun usage, in TwitterAAE african-american english (AAE) dialect. In the hate-speech datasets a broader selection of groups was labeled: In Jigsaw this refers to all mentions of identity groups, whereas in SBIC and ImplicitHate only groups targeted by hate-speech were labeled. Noteworthy, potential annotation quality issues (complex task, limited annotator agreement[16]), different selections of groups and labeling protocols complicate the comparison of concepts learned on the different datasets.

Training Setup In our experiments, we train a linear layer or a Multi-layer-perceptron (MLP) with one hidden layer on top of LM's embeddings. Learning sensitive concepts is modeled as multi-label classification. To avoid optimizing learning rates for each LM, we use the Salsa Optimizer[19].

[1] https://github.com/HammerLabML/PreSeCoLM

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

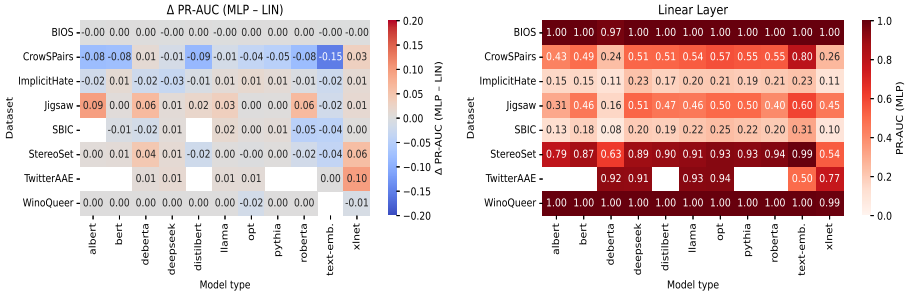

Fig. 1: Results per model and dataset combination. Left: Difference of PR-AUC between MLP and linear classifier, right: PR-AUC of linear layer for reference. Missing results occur when classifiers did not converge after multiple tries.

4 Experiment 1: Testing the linearity assumption

In our first experiment, we aim to verify if the common approach of using linear concepts or bias subspaces is adequate to represent sensitive concepts across different language models and test domains, under the research question

RQ1: Are linear models sufficient to learn sensitive concepts?

4.1 Experiment Setup

Using the full selection of language models and datasets (Section 3), we train the MLP and linear layer on the sensitive concepts. We apply 4-fold cross-validation and report the mean PR-AUCs over CV folds. For a few instances, the optimizer could not converge to a proper solution. In those cases, we repeated the training a couple of times, ultimately resulting in success rates between 0.81 to 1.0 for the models, where Winoqueer, TwitterAAE and the hate-speech datasets accounted for the majority of failure cases. To compare the performance of linear and MLP classifiers, we paired all results of the respective classifiers with otherwise identical settings (LM, dataset, identity group) and ran a t-test under the hypothesis that the linear classifier would outperform the MLP. We conducted the t-test over all results as well as for individual LMs, datasets, and groups.

4.2 Results

The t-test confirmed our hypothesis (PR-AUC of linear classifier $>$ PR-AUC of MLP) with a test statistic of 4.977 and $p < 0.01$, but only a diminishing effect (Cohen's $d = 0.079$). We further tested for specific LMs, datasets and individual groups, running the t-test for the hypotheses that either MLP or linear layer would outperform the other. The following datasets and models showed significantly better performance for one classifier with $p < 0.01$: The linear layer lead to better results on SBIC ($d = 0.10$) and CrowSPairs ($d = 0.35$), the MLP on TwitterAAE ($d = 0.55$) and Jigsaw ($d = 0.47$). Among the

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

model families, BERT ($d = 0.24$), RoBERTa ($d = 0.29$) and text-embedding-3 ($d = 0.58$) achieved better results with the linear layer. Better MLP performance was observed for all DeBERTa models ($d = 0.28$), for Llama-3.2-1B ($d = 0.34$), xlnet-large-cased ($d = 0.21$) and Pythia-410m ($d = 0.36$), although the effects are not consistent in other Llama, XLNET and Pythia models. Figure 1 shows the PR-AUC difference for the model families and datasets. Notably, there is a high variance of PR-AUC scores for individual groups in the same dataset. Similarly, some groups achieve high scores on one dataset and significantly lower ones on another. We did not find obvious patterns that consistently hold across datasets. Similarly, there is no clear correlation between PR-AUCs and the group ratio in the datasets. We suspect that the dataset-specific labeling patterns might overshadow any potential effects. In conclusion, the linear classifier is generally a good choice. Most results were at least on par with the MLP, so the simpler classifier would be the better choice. We found the results of two datasets noteworthy: On TwitterAAE results were missing (no convergence) for many models and text-embedding-3 models (which generally showed superior results) only achieved random-guessing performance. This could be a sign that the dialect information does not persist in the last hidden layers of those models. Secondly, for Jigsaw the MLP almost consistently outperformed the linear layer. One could hypothesize that sensitive concepts in the real-word dataset might require a more complicated model to capture them compared to simpler datasets like CrowSPairs. However, it is not clear why this behavior does not show in SBIC and ImplicitHate. In addition, we discovered that specific LMs were better combined with the MLP. So, in the next step, we investigate those models closer.

5 Experiment 2: Generalization or overfitting?

As shown in the previous Section, linear models are generally a valid choice for learning sensitive concepts. However, we also found several models that worked better with MLP. In particular, this was observed on possibly more complex datasets. This leads to the question if the non-linear classifier can better extract the concepts from complex datasets, in a way that benefits generalization. Another explanation could be overfitting to dataset-specific labeling patterns, which might harm generalization. To address this question, we test cross-dataset transfer of sensitive concepts under the Research Question

RQ2: Do the MLP's performance hold in cross-dataset transfer?

If the MLP truly allows for better generalization, this should yield some benefits in dataset transfer as well. In case of overfitting, we would expect the MLP performance to drop stronger in transfer compared to the linear layer.

5.1 Experiment Setup

We focus on Pythia, Xlnet, Deberta, Llama models (those that consistently performed better with MLP) and the Deepseek and Roberta models where the MLP showed benefits on Jigsaw (at least for some groups). Again, we train both classifiers on all the datasets, but test on unseen data from the training dataset

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

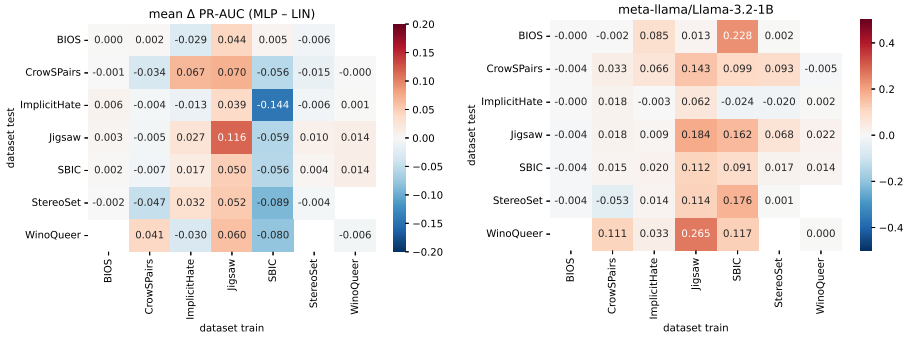

Fig. 2: Difference of PR-AUC (MLP-linear), mean results over all models (left) and Llama-3.2-1B results (right). Results are aggregated over all models (left) and all groups per dataset/ transfer case. Missing results occur if two datasets do not share any labels.

(in-domain) and on the other datasets (transfer). Here, we used a train-test split, so in-domain results can differ from the previous experiment.

5.2 Results

As Figure 2 shows, the MLP benefits do not hold in general despite selecting models that benefited from MLP in our first experiment. Interestingly, even in-domain the mean PR-AUC scores are often higher for the linear classifier. However, when training on Jigsaw the MLP significantly outperforms the linear layer in-domain and the effect transfers to other datasets. Notably, Jigsaw is our largest dataset with 10x-20x the training samples compared to SBIC and ImplicitHate, which should aid generalization, even in the MLP with more parameters. Most other training datasets yield either mixed results or similar performance for both classifiers. Mixed results between transfer cases could be explained by the selection of groups in the datasets. For instance, WinoQueer and StereoSet do not share any labels. When training on SBIC the MLP consistently underperforms with significant drops in specific transfer cases. We also report the highest variance of PR-AUCs between models when training on SBIC. This could be an indication for label inconsistencies or very specific labeling patterns in the dataset, which harm transfer and make results more inconsistent. While most models perform rather consistently, we do find some exceptions: In particular, with Llama-3.2-1B embedings the MLP outperforms the linear layer even on SBIC and we observe much higher MLP scores for the transfer from SBIC to BIOS and Jigsaw to WinoQueer, even exceeding the in-domain benefits. This emphasizes that for certain LMs non-linear concept classifiers are beneficial, though this should be carefully validated on a case to case basis. In particular, label quality and specific identity groups seem to have a big influence, and certainly require more attention.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

6 Discussion

This paper focused on the question whether sensitive concepts are encoded linearly in language models. Our first experiment showed that, in general, linear classifiers are the better choice. There were some notable exceptions though, as shown in our second experiment. While specific language models might yield better results when combined with a non-linear classifier, differences between datasets indicate that the amount and complexity of data as well as label consistency play an important role. Since most bias benchmarks are rather small, we emphasize that linear models should be the first choice, especially on LLM embeddings. On larger benchmarks and with careful validation however, non-linear methods could yield better results. Future work should investigate label quality of the real-world datasets more closely. In addition, a more finegrained analysis of specific identity groups could yield valuable insights.

References

[1] Belrose et al. Leace: Perfect linear concept erasure in closed form. *NIPS*, 2023.

[2] Iskander et al. Shielded representations: Protecting sensitive attributes through iterative gradient-based projection. ACL, 2023.

[3] Amara et al. Erasing more than intended? how concept erasure degrades the generation of non-target concepts. In *IEEE/CVF*, 2025.

[4] Kim et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). PMLR, 2018.

[5] Koh et al. Concept bottleneck models. PMLR, 2020.

[6] Fel et al. Craft: Concept recursive activation factorization for explainability. 2023.

[7] Schroeder et al. Evaluating concept discovery methods for sensitive attributes in language models. *ESANN*, 2025.

[8] Bolukbasi et al. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *NIPS*, 29, 2016.

[9] Caliskan et al. Semantics derived automatically from language corpora contain human-like biases. *Science*, 2017.

[10] Anwar et al. Foundational challenges in assuring alignment and safety of large language models. *TMLR*, 2024.

[11] Nadeem et al. StereoSet: Measuring stereotypical bias in pretrained language models. ACL, 2021.

[12] Nangia et al. Crows-pairs: A challenge dataset for measuring social biases in masked language models. 2020.

[13] Felkner et al. WinoQueer: A community-in-the-loop benchmark for anti-LGBTQ+ bias in large language models. ACL, 2023.

[14] De-Arteaga et al. Bias in bios: A case study of semantic representation bias in a high-stakes setting. FAccT, 2019.

[15] Blodgett et al. Twitter Universal Dependency parsing for African-American and mainstream American English. ACL, 2018.

[16] Sap et al. Social bias frames: Reasoning about social and power implications of language. ACL, 2020.

[17] Borkan et al. Nuanced metrics for measuring unintended bias with real data for text classification. In *world wide web conference*, 2019.

[18] ElSherief et al. Latent hatred: A benchmark for understanding implicit hate speech. ACL, 2021.

[19] Kenneweg et al. No learning rates needed: Introducing salsa - stable armijo line search adaptation. *IJCNN*, 2024.