Unlearning Mechanisms for Document Classification and Fair Decision-Making

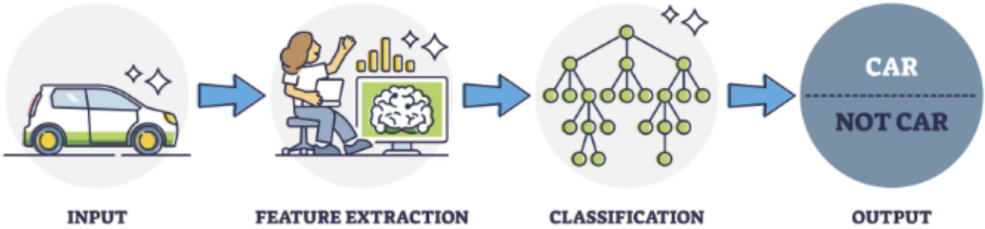
Aadya Goel, Adam Ge

Mentor: Mayuri Sridhar
MIT PRIMES October Conference

October 19, 2025

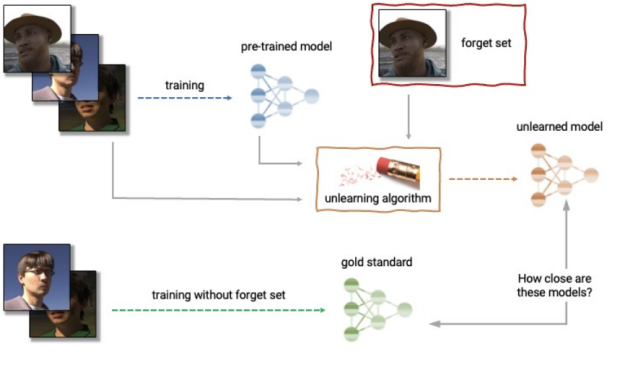
Machine Learning

- Machine Learning is the process of training a model, typically by tuning parameters, to make predictions



Machine Unlearning

- Machine Unlearning is the process of removing training data without retraining the entire model
- Gold Standard: Remove data to be unlearned and retrain the model



Why Do Machine Unlearning?

Privacy

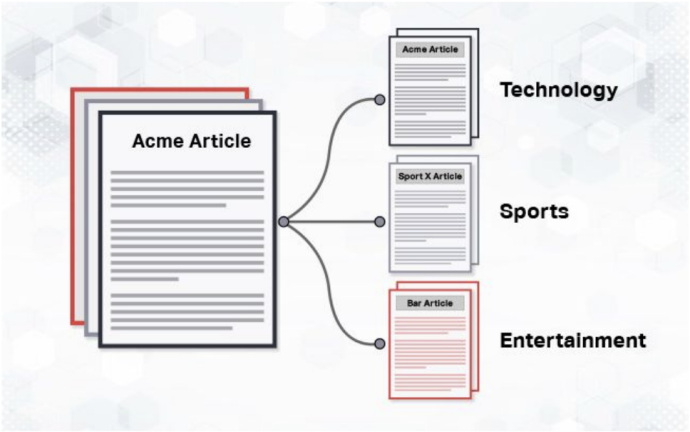
Training data may be collected from individuals and be private/sensitive.

Model Maintenance

Model maintenance may be necessary to remove false data or bias within the model.

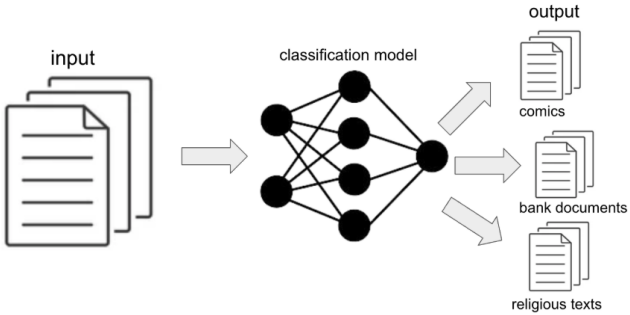
Background

- Document classification is the process of assigning documents to different categories or classes



Background

- Done through image classification or text classification
- Large task of NLP (Natural Language Processing), commonly done with word embedding



Problem

- Classification label of a document may contain potential vulnerabilities
- How do we remove the class label while reclassifying all documents with this label?

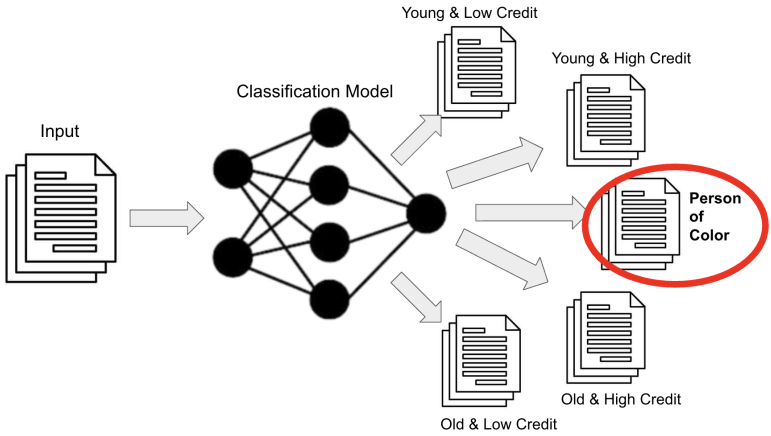
Problem

- Classification label of a document may contain potential vulnerabilities
- How do we remove the class label while reclassifying all documents with this label?

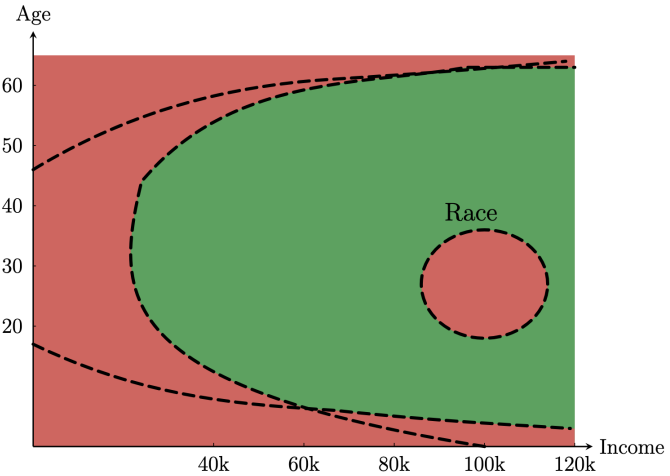
Example

Loan approval should not be dependent on the race of a person, though a model might incorrectly consider it.

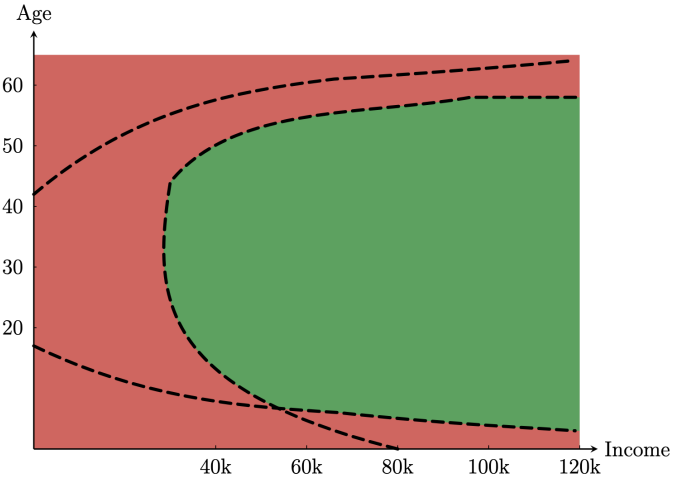
Loan Approval Classification



Decision Boundaries Pre-Unlearning

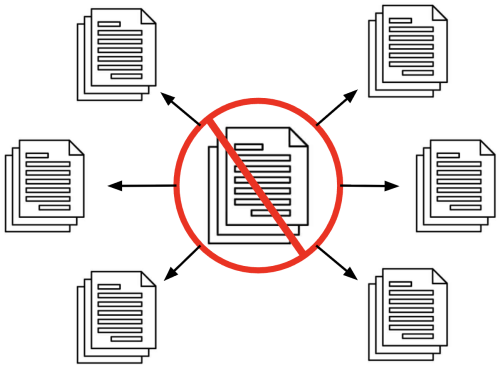


Decision Boundaries Post-Unlearning



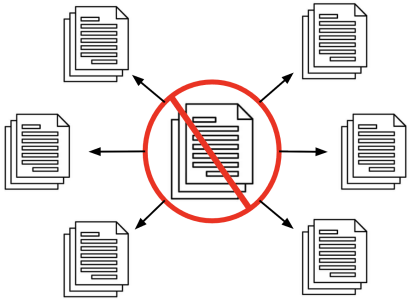
State of the Art

- Randomly distributes documents within the remaining classes
- Overall accuracy drops from 94.03% to 83.57%, indicating loss of utility



State of the Art

- Randomly distributes documents within the remaining classes
- Overall accuracy drops from 94.03% to 83.57%, indicating loss of utility



How can we map each document to its next-top class?

Model Architecture

- Baseline Model

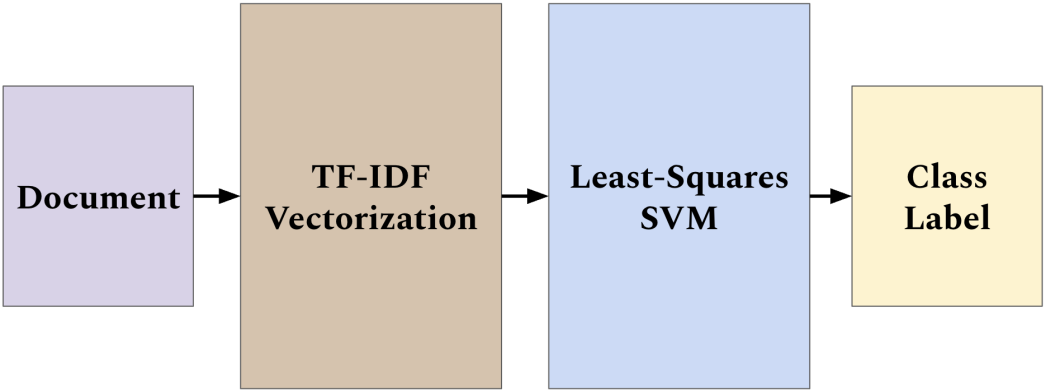
Model Architecture

- Baseline Model
- Unlearn documents within target class

Model Architecture

- Baseline Model
- Unlearn documents within target class
- Optimize unlearning algorithm to match retrain-equivalent solution

Baseline Model



Motivation for Hessian

- How do we remove the influence of the unlearned documents from the model?

Hessian Matrix

Hessian H is the matrix of second derivatives of the loss (i.e. curvature)

- Measures how parameter changes affect loss

$$H(\theta) = \nabla_\theta^2 L(\theta) = \frac{1}{n}\sum_{i=1}^{n} J_i^\top \nabla_z^2 \ell(y_i, z)\Big|_{z=f_\theta(x_i)} J_i + \lambda I, \quad J_i = \frac{\partial f_\theta(x_i)}{\partial \theta}.$$

Hessian Matrix

Hessian H is the matrix of second derivatives of the loss (i.e. curvature)

- Measures how parameter changes affect loss

$$H(\theta) = \nabla_\theta^2 L(\theta) = \frac{1}{n} \sum_{i=1}^{n} J_i^\top \nabla_z^2 \ell(y_i, z)\Big|_{z=f_\theta(x_i)} J_i + \lambda I, \quad J_i = \frac{\partial f_\theta(x_i)}{\partial \theta}.$$

How does this apply to our LS-SVM?

Least-Squares Hessian

- LS-SVM Objective:

$$L(\theta) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \;+\; \frac{\lambda}{2}\,\|\theta\|^2.$$

- **Squared-Loss Hessian:**

$$H \;=\; \lambda I \;+\; \frac{1}{n} X^\top X$$

Hessian Downweight

- H^{-1} maps gradient changes into a parameter change
- Compute gradient contribution of removed data and take one-shot corrective step when unlearning data

$$g_{\text{del}} = \frac{1}{n} \sum_{i \in S} \nabla_\theta \ell_i(\theta^\star). \tag{1}$$

$$\theta_{\text{new}} \approx \theta^\star - H(\theta^\star)^{-1} g_{\text{del}} \tag{2}$$

L-BFGS

- L-BFGS is a numerical optimization algorithm used to approximate the inverse Hessian matrix

$$H_{k+1} = \left(I - \rho_k s_k y_k^\top\right) H_k \left(I - \rho_k y_k s_k^\top\right) \; + \; \rho_k \, s_k s_k^\top, \qquad \rho_k = \frac{1}{y_k^\top s_k}. \tag{3}$$

L-BFGS

- L-BFGS is a numerical optimization algorithm used to approximate the inverse Hessian matrix

$$H_{k+1} = \left(I - \rho_k s_k y_k^\top\right) H_k \left(I - \rho_k y_k s_k^\top\right) \; + \; \rho_k \, s_k s_k^\top, \qquad \rho_k = \frac{1}{y_k^\top s_k}. \tag{4}$$

- Uses approximation to get a curvature-aware search direction

$$p_k \; \approx \; - H_k^{-1} \, g_k, \qquad g_k \; = \; \nabla L(\theta_k). \tag{5}$$

L-BFGS

- L-BFGS is a numerical optimization algorithm used to approximate the inverse Hessian matrix

$$H_{k+1} = (I - \rho_k s_k y_k^\top) H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top, \qquad \rho_k = \frac{1}{y_k^\top s_k}. \tag{6}$$

- Uses approximation to get a curvature-aware search direction

$$p_k \approx -H_k^{-1} g_k, \qquad g_k = \nabla L(\theta_k). \tag{7}$$

- Updates the weights with a line-searched step

$$\theta_{k+1} = \theta_k + \eta_k p_k, \tag{8}$$

L-BFGS

- L-BFGS is a numerical optimization algorithm used to approximate the inverse Hessian matrix

$$H_{k+1} = (I - \rho_k s_k y_k^\top) H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top, \qquad \rho_k = \frac{1}{y_k^\top s_k}. \tag{9}$$

- Uses approximation to get a curvature-aware search direction

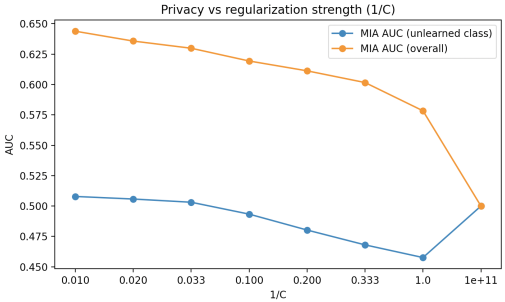
$$p_k \approx -H_k^{-1} g_k, \qquad g_k = \nabla L(\theta_k). \tag{10}$$

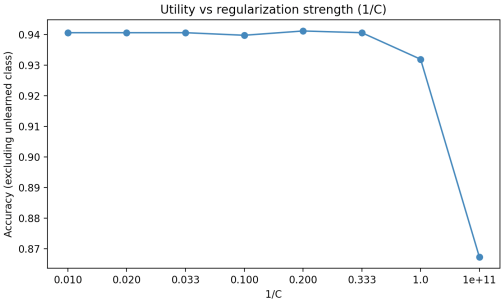
- Updates the weights with a line-searched step

$$\theta_{k+1} = \theta_k + \eta_k p_k, \tag{11}$$

- Running a short pass on the reduced objective settles to the retrain-equivalent solution

Regularization Optimization

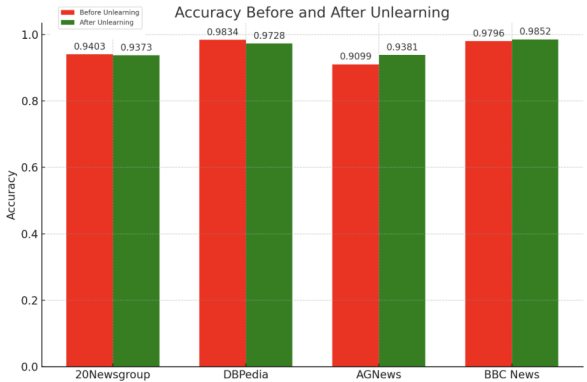




We choose an optimal value of regularization strength of 0.2.

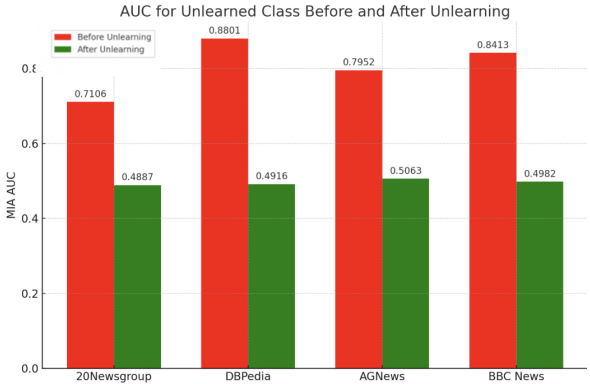
Model Accuracy

Model accuracy remains nearly unchanged after Hessian downweight on all benchmarks



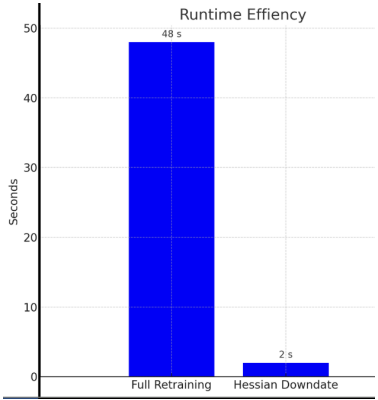
Privacy Guarantee

Attackers lose ability to detect deleted data post-unlearning.



Time Effiency

Hessian downweight completes unlearning in only 2 seconds, demonstrating a roughly 25x speed-up.



Takeaways

- Hessian downweight + brief L-BFGS achieves retrain-equivalent models while preserving accuracy and dropping membership-inference AUC on the unlearned class to 0.5
- Our algorithm runs faster than full retraining, and similarity-based reassignment maintains utility—making class-level unlearning feasible for real text pipelines

Background

- Machine learning algorithms used to determine critical life opportunities (e.g. loan approval, criminal justice outcomes)
- Algorithms may amplify bias reflecting societal inequalities

Example

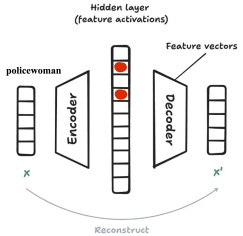
Loan approval predictions may differ for two people of different races even if they share the same qualifications.

Sources of Bias

- Bias from input data (historical inequality and selection bias)
  - Over-representation of majority populations and under-representation of marginalized groups
  - Labels may reflect systemic discrimination through historical human decisions
- Bias from model (subgroup inequality)
  - Model may rely on features (proxies) that are spuriously correlated with sensitive attributes
  - Model error rates may vary across demographic group, causing unequal error distribution

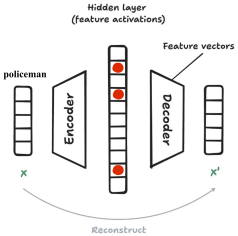
Sparse Autoencoder (SAE)

- Autoencoder with a wide, sparse hidden layer so that features are separated
- Can identify features representing a sensitive attribute, such as gender, by
  - Use paired inputs differing only in the sensitive attribute (e.g. policeman and policewoman)
  - Identify which hidden neuron(s) change the most between the input words; these neurons encode sensitive attribute



Sparse Autoencoder (SAE)

- Autoencoder with a wide, sparse hidden layer so that features are separated
- Can identify features representing a sensitive attribute, such as gender, by
  - Use paired inputs differing only in the sensitive attribute (e.g. policeman and policewoman)
  - Identify which hidden neuron(s) change the most between the input words; these neurons encode sensitive attribute



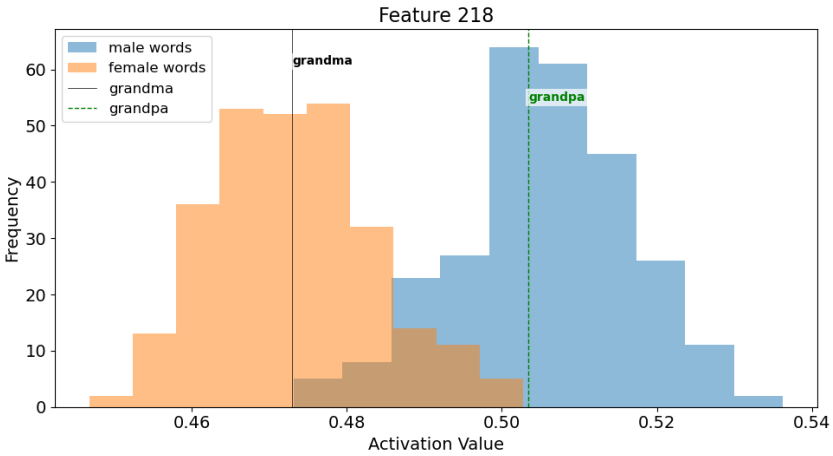
Sensitive Feature Detection Pipeline

- LLM is used to generate embeddings that are fed into the SAE
- SAE is trained with a reconstruction loss (so that the output of the decoder and the input embedding from LLM are close) and a sparsity penalty (limits number of active features)
- Identify which feature in the sparse coefficient vector encodes gender
- Used Gendered Words Dataset + lists of Wiktionary words that are either male or female to train SAE.

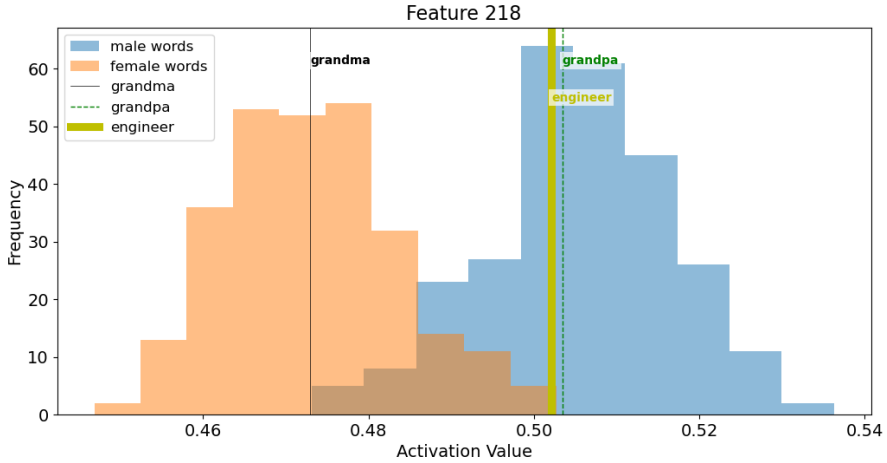
Debiasing

- Finetune the LLM (sentence transformer)
  - Find mean of the average of the female words distribution and the average of the male words distribution
  - Define mean-squared loss (MSE) based on difference between the biased word's gender feature in the SAE and the mean described above, backpropagate

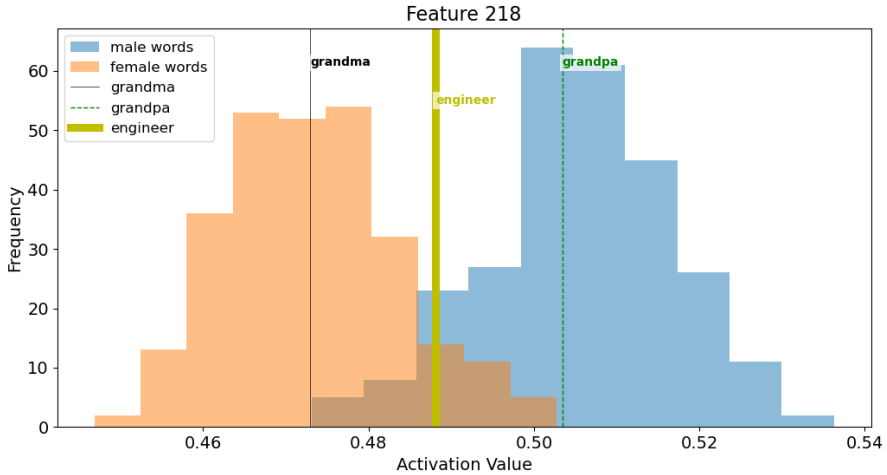
Activation Values for Feature 218 for Inherently Gendered Words



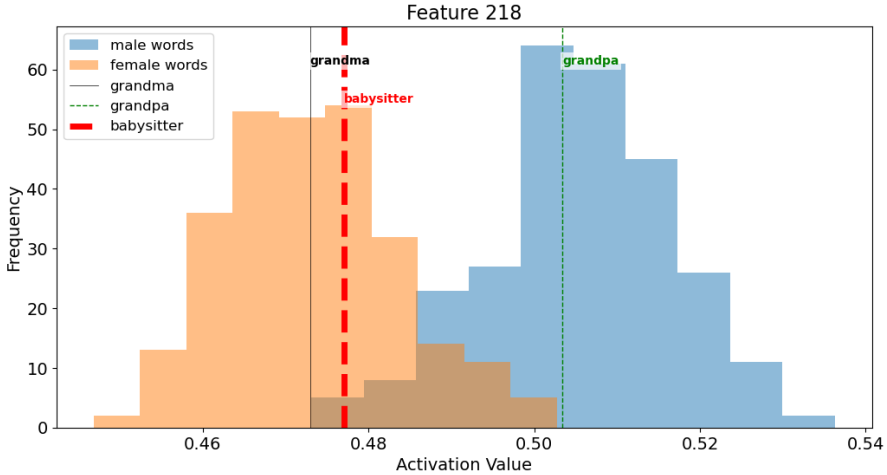
Engineer Feature Value Before Debiasing



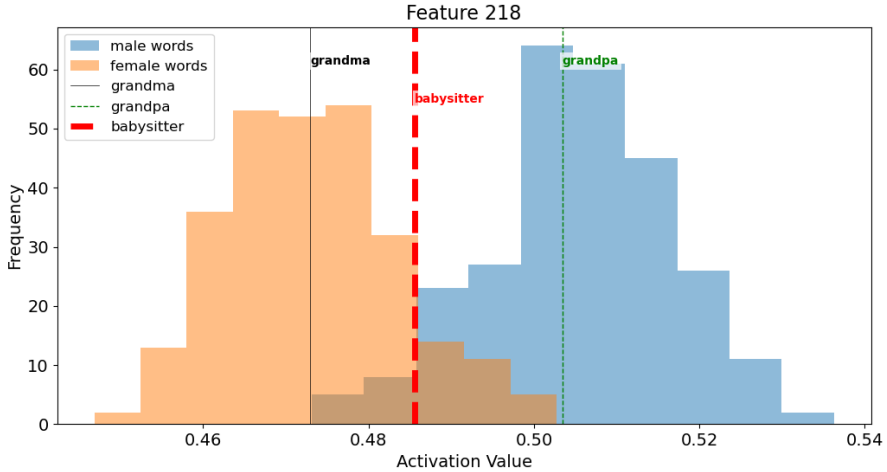
Engineer Feature Value After Debiasing



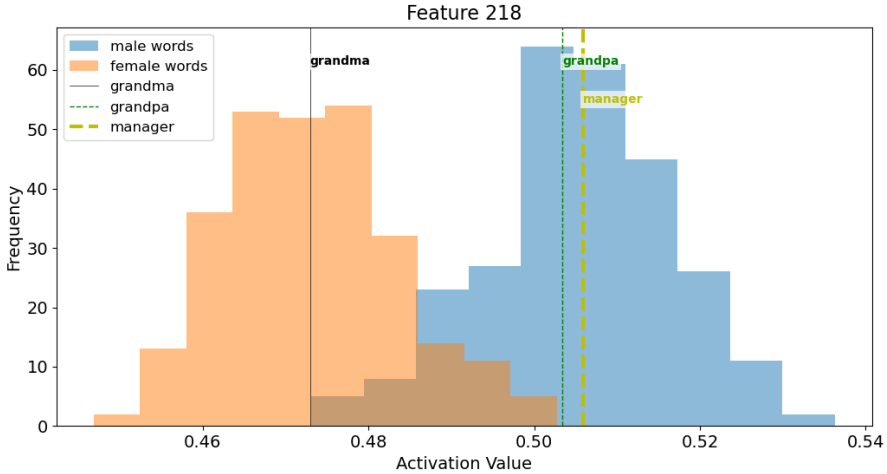
Babysitter Feature Value Before Debiasing



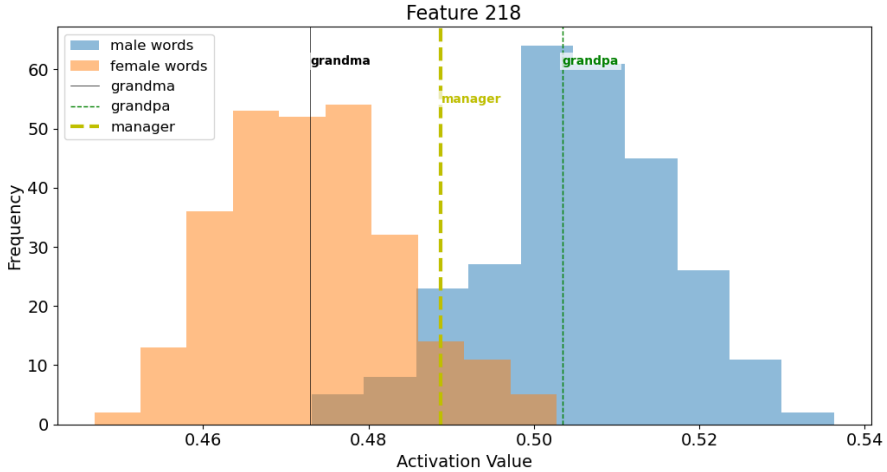
Babysitter Feature Value After Debiasing



Manager Feature Value Before Debiasing



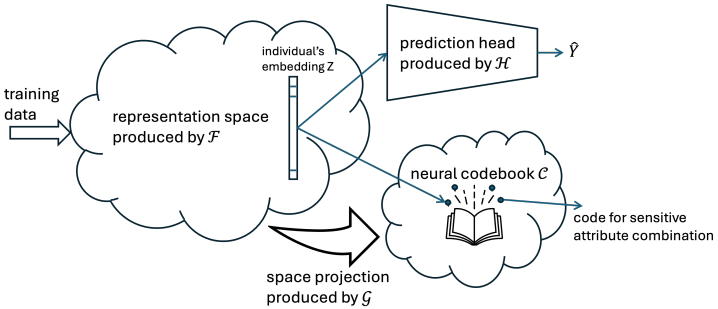
Manager Feature Value After Debiasing



Intersectional Bias

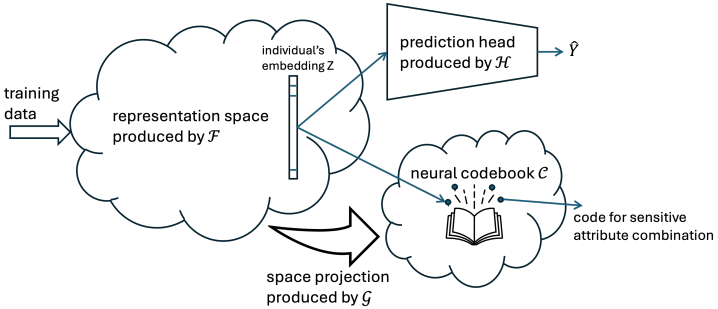
- Marginal fairness evaluates the fairness of a model one attribute at a time
- Harm is amplified at the intersection of multiple attributes
  - A model may be 90% accurate for a certain race and 90% accurate for a certain gender, but only 55 % accurate for the group with both the race and the gender.
- Intersectional groups grow exponentially, causing data sparsity and making fairness harder to achieve

Iterative Sifting Debiasing Model



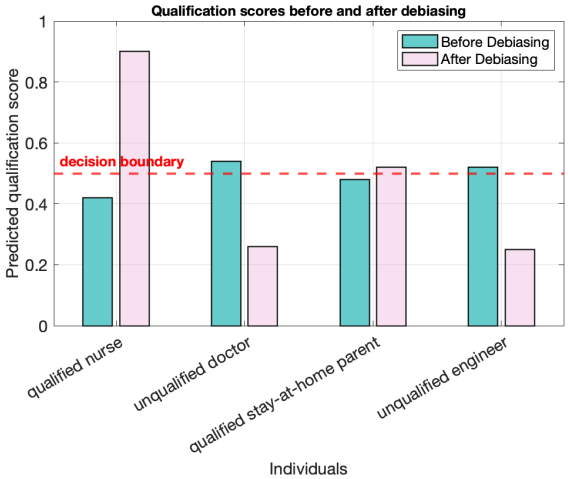
- $\mathcal{C}$ denotes the neural codebook. There are $2^2 = 4$ codes, denoting the 4 combinations of race (black/white) and gender (male/female).
- Each code in the codespace is the center of all the projected embeddings with the same sensitive attributes as the code.

Iterative Sifting Debiasing Algorithm

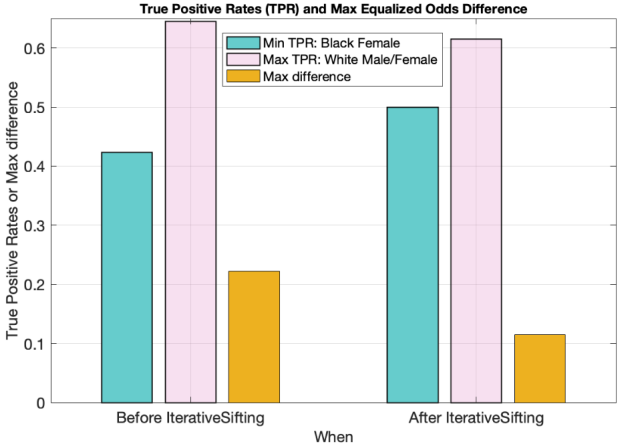


- The projection network learns to map the projected embeddings closer to the correct code in the codebook, which captures the sensitive information
- The model then makes the projected embeddings equally distant from the codes, which forces the embedding space to unlearn sensitive attribute information

Results


Qualification scores before and after debiasing
Before Debiasing
After Debiasing
decision boundary
Predicted qualification score
1
0.8
0.6
0.4
0.2
0
qualified nurse
unqualified doctor
qualified stay-at-home parent
unqualified engineer
Individuals

Results


True Positive Rates (TPR) and Max Equalized Odds Difference
Min TPR: Black Female
Max TPR: White Male/Female
Max difference
True Positive Rates or Max difference
0
0.1
0.2
0.3
0.4
0.5
0.6
Before IterativeSifting
After IterativeSifting
When

Takeaways

- Our SAE successfully identified and mitigated the gender-encoding features in the model's representations
- Our Iterative Sifting approach removed sensitive information and improved fairness without retraining

Conclusion

- We developed practical methods to do unlearning for both document classification models and to remove bias

Future Works

- Extend method to deep learning architectures (beyond linear SVMs)
- Apply unlearning metrics (e.g., certified unlearning bounds)
- Work on more datasets (graph datasets, larger datasets)
- Evaluate the task accuracy and semantic consistency of the LLM after unlearning (for both SAE and Iterative Sifting model)

Acknowledgments

- Thank you to Slava Gerovitch, Srini Devadas, and the rest of MIT PRIMES for making this project possible
- Thank you to our mentor, Mayuri Sridhar, for guiding our research, and supporting us in every way
- Thank you to our parents for supporting us through the program

References

- Lei Kang, Mohamed Ali Souibgui, Fei Yang, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. “Machine Unlearning for Document Classification”. arXiv preprint arXiv:2404.19031, 2024.
- Scieur, Damien, et al. "Generalization of Quasi-Newton methods: application to robust symmetric multisecant updates." International Conference on Artificial Intelligence and Statistics. PMLR, 2021.
- Charles O’Neill, Christine Ye, Kartheik Iyer, John F. Wu. "Disentangling Dense Embeddings with Sparse Autoencoders." arXiv preprint arXiv:2408.00657, 2024.
- Usman Gohar, Lu Cheng. "A Survey on Intersectional Fairness in Machine Learning: Notions, Mitigation, and Challenges." IJCAI 2023.