NLPGuard: A Framework for Mitigating the Use of Protected Attributes by NLP Classifiers

NLPGuard: A Framework for Mitigating the Use of Protected Attributes by NLP Classifiers

SALVATORE GRECO*, Politecnico di Torino, Italy
KE ZHOU, Nokia Bell Labs, UK
LICIA CAPRA, University College London, UK
TANIA CERQUITELLI, Politecnico di Torino, Italy
DANIELE QUERCIA, Nokia Bell Labs, UK

AI regulations are expected to prohibit machine learning models from using sensitive attributes during training. However, the latest Natural Language Processing (NLP) classifiers, which rely on deep learning, operate as black-box systems, complicating the detection and remediation of such misuse. Traditional bias mitigation methods in NLP aim for comparable performance across different groups based on attributes like gender or race but fail to address the underlying issue of reliance on protected attributes. To partly fix that, we introduce NLPGuard, a framework for mitigating the reliance on protected attributes in NLP classifiers. NLPGuard takes an unlabeled dataset, an existing NLP classifier, and its training data as input, producing a modified training dataset that significantly reduces dependence on protected attributes without compromising accuracy. NLPGuard is applied to three classification tasks: identifying toxic language, sentiment analysis, and occupation classification. Our evaluation shows that current NLP classifiers heavily depend on protected attributes, with up to 23% of the most predictive words associated with these attributes. However, NLPGuard effectively reduces this reliance by up to 79%, while slightly improving accuracy.

***Disclaimer*: This paper contains examples of language that some people may find offensive.**

Additional Key Words and Phrases: protected attributes, bias, fairness, natural language processing, toxic language, large language models, crowdsourcing

ACM Reference Format:
Salvatore Greco, Ke Zhou, Licia Capra, Tania Cerquitelli, and Daniele Quercia. 2024. NLPGuard: A Framework for Mitigating the Use of Protected Attributes by NLP Classifiers. In . ACM, New York, NY, USA, 29 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent years, the adoption of deep learning-based NLP models has exponentially increased. Transformer-based models, such as BERT [18], T5 [57], and GPT [8], have achieved unthinkable levels of performance on several natural language tasks. However, despite being increasingly accurate, these models remain black-boxes [28]. For an NLP classification task, models predict a class label from an input text without providing any information on the complex internal decision-making mechanism, making it challenging to identify and mitigate potential bias and/or unfair behavior in such models.

*Work done at Nokia Bell Labs.

Accepted at CSCW'24, 2024, San José, Costa Rica

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Text	*P(T)*
"I like this city! There are many black people!"	0.53
"The homosexual marriage bill will be debated soon! I am in favor!"	0.62
"This city is incredibly modern! If you are gay, you are not judged."	0.88
"I hate this fucking shitty city! There are many black people!"	0.99

Fig. 1. **Toxicity probabilities P(T) to four sentences predicted as toxic by a classifier**. The first three sentences are misclassified, while the last is correctly identified.

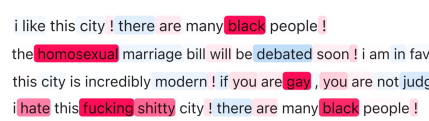

Fig. 2. **Words impacting the toxicity classification of the four sentences in Figure 1**. The more intense the red (blue) color of a word, the more important the word contributes to toxic (non-toxic) classification.

Upcoming privacy laws regulating the use of AI will soon demand that learning shall not be done on protected attributes such as race, gender, or sexual orientation, as already identified by the *General Data Protection Regulation* (GDPR), the *UK Government*, and the anti-discrimination legislation in the United States [10, 21, 26]. Ensuring AI models avoid using protected attributes in decision-making is termed '*fairness through unawareness*' [45], and it is crucial in many real-world scenarios. For instance, NLP-based systems often assess job applicants' resumes. Following the Civil Rights Act in the US, discrimination based on race, sex, nationality, or other protected attributes is forbidden. Hence, these NLP systems must omit words linked to protected attributes to prevent discriminatory practices against candidates, such as the "sexist" Amazon Recruitment tool,[1] a system that learned to downgrade resumes containing the word *'women'*. Content moderation is another example, where all users should be treated equitably, without having their contributions censored or suppressed because of, for example, their demographic characteristics.

However, as we will demonstrate in our analysis, state-of-the-art models often base their predictions on protected attributes, and accurate ones are frequently black boxes, posing challenges in identifying such misuse. Consider, for example, the task of determining whether a sentence contains toxic language or not in a dataset we will analyze. In Figure 1, we report four example sentences, together with the outcome of a toxicity classifier P(T); in Figure 2, we highlight in red the important words used by the classifier to make these predictions. As shown, the presence of words such as *'black'*, *'gay'*, or *'homosexual'* is used to distinguish between toxic or non-toxic texts. Yet, these words are protected attributes and should not be used in such classifications at all.

As discussed in §2, prior studies on bias in NLP primarily focused on two challenges: ensuring fair performance across different groups and rectifying unfairness in word representations. However, these solutions only target specific biases and fail to eliminate the reliance of models on protected attributes for predictions. Therefore, we propose methods to reduce this bias in black-box NLP classifiers, removing most protected attributes from their decision-making process while maintaining accuracy, and making these approaches applicable across various datasets and tasks.

In so doing, we make four main contributions:

(1) We introduce NLPGuard (§3), a framework with three components: (1) an *Explainer* that finds the most important words for predictions; (2) an *Identifier* that checks if these words are about protected attributes; and (3) a *Moderator* that adjusts the training data to re-train the NLP model to reduce learning from such protected attributes.
(2) We evaluate each part of our framework and use it to mitigate toxicity detection in Wikipedia comments with BERT (§4). BERT depends on protected attributes for toxicity predictions (23% of the most predictive words), but our approach cuts this down by 60% and even increases prediction accuracy by 0.8%.

[1] https://www.bbc.com/news/technology-45809919

(3) We then evaluate whether our framework generalizes to different types of data and tasks, not just toxicity detection (§5). We found that our framework reduces the use of protected attributes by 79% when applied to out-of-distribution data. Also, it reduced reliance on protected attributes without compromising accuracy in tasks like sentiment analysis and occupation classification.
(4) We make NLPGuard publicly accessible,[2] and discuss how to incorporate it into existing NLP systems, its impact, and its limitations (§6).

2 RELATED WORK

2.1 AI Regulations and Laws

The growth of AI systems has raised privacy and discrimination concerns, leading to the introduction of numerous regulations and laws governing their use. In the European Union (EU), in May 2018, the GDPR [70] was introduced, which demands organizations ensure that personal data is processed lawfully, fairly, and transparently. It prohibits processing sensitive personal attributes such as race, ethnicity, religion, and political opinions, unless legitimately justified. The EU proposed the AI Act [42, 71], which defines rules and obligations depending on the level of risk of AI systems (e.g., transparency, documentation, human oversight) [51]. In the United Kingdom (UK), the UK Equality Act 2010 [35] established that it is unlawful to discriminate based on nine protected characteristics: age, disability, gender reassignment, marriage and civil partnership, pregnancy and maternity, race, religion or belief, sex, and sexual orientation. Compliance with the act is enforced by the Equality and Human Rights Commission (EHRC). In the United States (US), the Anti-discrimination Act [12] safeguards individuals from unfair treatment based on protected attributes. In late 2022, a blueprint of the AI Bill of Rights was passed [32], declaring that algorithms that discriminate or perform unjustified different treatment based on protected attributes violate legal protections. AI regulations will continue to evolve in the coming years [34, 49], driven by a common goal: to minimize discriminatory outputs based on protected characteristics [10, 21, 26].

2.2 Bias mitigation for NLP

Bias in NLP decision-making has manifested itself in several ways, including dialogue generation [19], text classification [20], and machine translation [66]. It usually arises from training data [22, 67]. For instance, pre-trained models and word embedding can inherit biases and stereotypes present in the large training corpora [7, 9, 25, 58]. When quantifying bias, existing works generally highlight disparities between demographic groups, with differences in performance or selection bias on protected attributes such as race, gender, religion, and sexual orientation [22, 29, 37, 67].

To address biases in NLP, techniques can be developed that act at the three main stages of the NLP pipeline [38, 75]: *pre-processing* (modifying training data), *in-processing* (imposing fairness constraints during model training), and *post-processing* (adjusting classifier predictions based on fairness metrics). Most existing works focus on the first two stages, exploiting data augmentation and modified model training techniques [4, 6, 20, 52, 58, 64, 77]. Furthermore, most of those studies focus on one protected category at a time. For example, Badjatiya et al. [6] proposed the identification of protected attributes, such as gender, by creating a manual list of words, measuring the skewed occurrence of words across classes or predicted class probability distribution of words. Park et al. [52] introduced gender swapping to equalize the number of male and female entities in the training data. Dixon et al. [20] proposed dataset augmentation strategies that generate new sentences using templates or replace protected attributes with generic tags, such as part-of-speech or named-entity tags. Zhang et al. [77] proposed mitigating biases in the training data by assuming

[2]The code repository of our framework is available at https://github.com/grecosalvatore/nlpguard

a non-discrimination distribution and then reconstructing the distribution using instance weighting. Ravfogel et al. [58] proposed removing information from neural representations concerning gender or race for debiasing word embedding for NLP classification.

These past works try to mitigate unintended bias and performance imbalance between subgroups by (1) removing implicit bias from word embeddings, (2) performing data augmentation on the training set (data-based), or (3) intervening directly in the model architecture or objective function (model-based). However, a gap remains in evaluating (and tackling) the extent to which NLP classifiers depend on protected attributes for their predictions. In this paper, we aim to fill that gap. We consider the following definition of *fairness through unawareness*: "an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process" [27, 40, 45]. Textual data is unstructured; hence, protected attributes are not explicitly delineated as input features, such as columns used in structured datasets. Consequently, we refine this definition in the context of NLP applications to ensure words associated with protected characteristics are not utilized in decision-making unless necessary. Our approach aims to reduce the use of protected attributes in the decision-making process of NLP models, thereby better aligning them with legal regulations.

Compared to prior work, our approach not only has a different objective, but it also overcomes two of their main limitations: (1) their focus on a subset of protected attributes at a time (usually race and gender); (2) their manual and static identification of protected attributes via pre-defined dictionaries, lists of identity terms, or additional annotations. The only technique addressing these limitations is *Entropy-based Attention Regulation* (EAR) [4]. EAR introduces a regularization term to discourage overfitting to training-specific potentially biased terms. However, those terms are automatically identified during training, leaving no flexibility for users to select which categories to mitigate. Unlike previous techniques, our approach: (1) identifies and mitigates multiple protected categories simultaneously; (2) can be fully automated, allowing for a dynamic update of the dictionary of protected attributes; and (3) allows for the selection of the categories to mitigate.

3 OUR MITIGATION FRAMEWORK

Our Mitigation Framework, namely NLPGuard, has been designed to be generally applicable to any supervised machine learning-based NLP classification model applied on an unlabelled corpus. As illustrated in Figure 3, NLPGuard takes in input an unlabeled corpus and a pre-trained NLP classifier (together with its training dataset) to produce a mitigated training dataset in output. Ground truth class labels for the unlabelled corpus are not required; rather, the classifier is used to generate them, both for in-distribution (i.e., data that comes from the same distribution as the original training dataset) and for out-of-distribution data where labels are unavailable. Because of the black-box nature of NLP classifiers based on deep learning models, labels might be predicted using protected attributes. To mitigate that, our framework comprises the following three components:

A. Explainer. This component uses *Explainable Artificial Intelligence* (XAI) techniques to identify the most important words used by the model for its predictions. The XAI field has made great strides in making black-box models more transparent, and several techniques exist to explain NLP classifiers [14]. The best one for our purpose should have two qualities: (1) quantify the importance of each feature word (feature-based), and (2) be applicable to explain the model's predictions after training (post-hoc). Many techniques meet these requirements, and most of them measure the importance of each word for the prediction within an individual sentence (local-explanations) [2, 44, 59, 63, 69, 73]. Our Explainer component first identifies the words important for prediction within all individual sentences, exploiting any of those techniques. Each word is, as such, associated with multiple scores, one for each occurrence in each sentence. Second, it determines the most important predictive words for the model as a whole (global-explanations) following the idea of

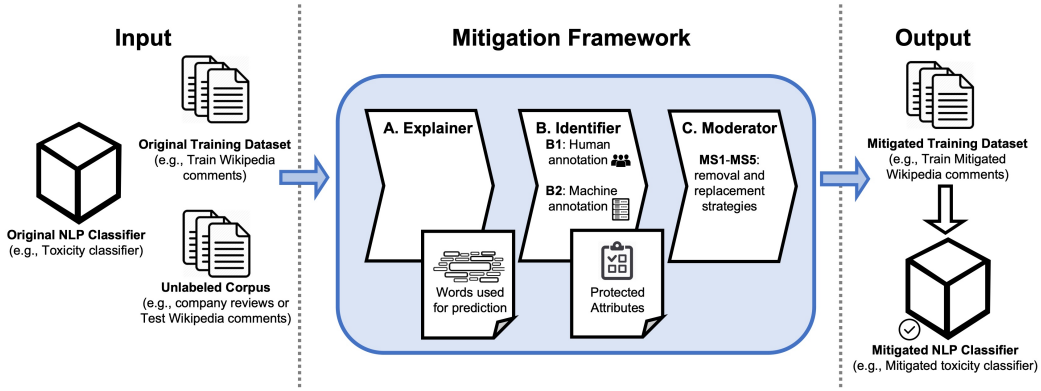

Fig. 3. **Our Mitigation Framework.** It takes the original NLP classifier, the original training dataset, and a new unlabeled corpus as input. The framework consists of three components: A) an *Explainer* that identifies the most important words used by the classifier for predictions on the unlabeled corpus; B) an *Identifier* that determines which of those words are protected attributes; and C) a *Moderator* that generates a mitigated training dataset to re-train the classifier so to reduce reliance on the previously identified protected attributes.

some of these techniques, which aggregate the words' importance over many sentences to compute the overall importance [72, 73]. Specifically, for each word, it sums all their individual scores and divides them by their frequency to compute the word's classification score. The normalization step is required to also identify rare but important words. The output of the *Explainer* component is the ordered list of the most important words for the model's predictions.

B. Identifier. This component aims to annotate which of the previously detected important words refer to protected attributes. We consider the nine protected categories defined by the Equality Act: *age*, *disability*, *gender reassignment*, *marriage and civil partnership*, *pregnancy and maternity*, *race*, *religion or belief*, *sex*, and *sexual orientation*. Our framework allows for annotation using *human-in-the-loop* (B1) and *machine-in-the-loop* (B2) approaches.

B1. Human-in-the-loop Annotation. Crowdsourcing platforms, such as Amazon Mechanical Turk (MTurk) and Prolific [13], have been extensively used by the research community to recruit crowdworkers for data labeling purposes. Crowdworkers [60, 68] are anonymous people usually paid for completing simple tasks. This component leverages crowdsourcing to perform the protected attribute annotation of the most important words. In our work, we exploited MTurk, where for each important word {WORD}, participants were asked to answer the following question:

> **Question:** *Is the word {WORD} referring to:*
> **Possible Answers:** 1. *Age*, 2. *Disability*, 3. *Gender reassignment*, 4. *Marriage and civil partnership*, 5. *Pregnancy and maternity*, 6. *Race*, 7. *Religion or belief*, 8. *Sex*, 9. *Sexual orientation*, 10. *None of the above*;

To ensure data quality, we adopt a trap mechanism to detect random responses from participants and reject them (details of this mechanism are presented in §4.3).

B2. Machine-in-the-loop Annotation. As a cost-effective and scalable alternative to human-in-the-loop annotations via crowdsourcing, we also implemented a protected attributes annotation process that uses Large Language Models (LLMs). We did so inspired by a recent study [23] that found LLMs, including ChatGPT, to outperform crowdworkers in text-based annotation tasks. It also has been shown that LLMs are effective in solving many NLP tasks [56]. Specifically, we implemented an annotation process that interacts with ChatGPT as follows: two prompts provide the protected categories, their definitions, and links with additional information. Then, for each word, the LLM is asked to: (1) classify the word into one of the protected categories or none of them;

(2) provide a reliability score in the range $[0, 100]$; and (3) provide an explanation. For example, the word *'homosexual'* would be classified with the protected category *sexual orientation* and a score of 100/100 by GPT-3.5-Turbo (see Figures 6 – 9 in Appendix A for further details).

C. Moderator. This component produces a new mitigated training dataset that can be used to train a new classifier that uses fewer protected attributes previously identified. It takes the original pre-trained classifier, the original training dataset, and the list of the most important words enriched with the protected attribute label as input. It produces a new *mitigated training dataset* by adjusting the training dataset based on the identified protected attributes that can then be used to train a new mitigated classifier. We designed and tested five mitigation strategies (MS).

(MS1) Sentence-level removal. Previous works have shown that subsampling can be an easy but effective technique for data balancing [33, 61]. This mitigation strategy eliminates all sentences containing protected attributes from the training set. As a result, it reduces the overall number of training examples. For example, if a word W_i is identified as a protected attribute, all sentences in the training set that include that word are removed. The idea behind this strategy is that the imbalance in the number of training examples containing protected attributes for a particular class may have led the model to learn that these protected attributes are crucial for classifying that class.

(MS2) Word-level removal. This strategy removes only the protected attribute words from the sentences in the training set while preserving the number of examples. The process involves removing the identified protected attribute words from all sentences in the training set, thereby removing their influence on the model's learning process. The idea is that the model should be able to classify sentences without relying solely on the protected words, and rather use other words in the text too.

(MS3) Word-level replacement with a random synonym. This strategy replaces every instance of a protected attribute in the training set with one of its synonyms. It first uses embedding similarity techniques to identify the k-nearest neighbors for each protected attribute. Then, it randomly selects one of the k-most similar words to replace each instance of the protected attribute in the training set. This has been shown to mitigate bias in [6], and it is believed that it may also help mitigate the use of protected attributes in classification, as the model may learn to rely on other words for classifying the classes rather than solely relying on protected attributes. This approach maintains the same number of examples in the training set but increases the diversity of words.

(MS4) Word-level replacement with K random synonyms. This strategy expands the training set by generating new sentences using synonyms of protected attributes. Instead of replacing the protected attribute in-place with one similar word as in *MS3*, it creates k new sentences by replacing the protected attribute with each of its k-nearest neighbors. For example, given a sentence containing a protected attribute W_i, k new sentences are created by replacing W_i with each of its k most similar words. This increases the size of the training set and diversifies the words used in the sentences.

(MS5) Word-level replacement with hypernym. This strategy replaces instances of protected attributes in the training set with higher-level words, called hypernyms, which provide a more general representation of the category to which the protected attribute belongs. For example, the hypernym of *'dog'* could be *'animal'*. By using hypernyms instead of the specific protected attributes, the model may not discriminate based on these attributes in its classifications. This technique has been shown to be effective in mitigating accuracy imbalance between subgroups [6].

4 FRAMEWORK EVALUATION: EFFECTIVENESS AND SENSITIVITY

We evaluate the effectiveness of our framework and the sensitivity of the *Explainer* (§4.2), *Identifier* (§4.3), and *Moderator* (§4.4) components in mitigating a toxicity classifier applied to in-distribution data (i.e., the test set).

4.1 Evaluation task

We choose toxicity prediction as the main evaluation task in line with previous research. Toxicity classifiers are used in different contexts [39, 76], such as Reddit, Twitter, and 4chan, with competitive performance. However, they suffer from different types of biases [15, 16, 31, 62]. In Wikipedia, for example, any comment containing words associated with insults, offense, or profanity, regardless of the tone, the intent, and the context, would be classified as toxic; toxic language was however more likely predicted from minority communities, as found in [15], thus suggesting the use of protected attributes by such models.

In our experiments, we used the "original model" in the widely used detoxify [30] library[3] as a pre-trained toxicity classifier. This is a BERT-base and uncased model [18] trained on a dataset of publicly available Wikipedia comments.[4] It was fine-tuned for predicting 6 labels related to toxicity: *toxicity*, *severe toxicity*, *obscene*, *threat*, *insult*, and *identitiy attack*, achieving an average Area Under the ROC Curve (*AUC*) score of 98.6%. For the *toxicity* label only, the classifier achieved 0.82 macro and 0.93 weighted F1 scores (the dataset is imbalanced). This classifier is applied to the original test set comprising 153,164 texts, and predicted the toxicity label for 36,148 texts (23.6% of the test set).[5]

4.2 Component evaluation: Explainer

The *Explainer* aims to identify the most crucial words utilized by the classifier for predictions (as described in §3-A). The employed XAI technique can influence the words recognized as significant, thereby impacting the identified protected attributes on which the model relies to make predictions.

4.2.1 Evaluation metrics. To evaluate the effectiveness of the *Explainer* component, we first measure the impact on the model's predictive performance (F1 score) by removing the most important words identified by each XAI technique. An effective and precise explainer should result in a noticeable decrease in the predictive performance when these words are removed. Secondly, to assess the sensitivity of the *Explainer*, we measure the overlap of the most important words identified by different XAI techniques. A substantial overlap indicates consistent outputs across XAI techniques. Lastly, we measure the computation time for generating explanations to assess the efficiency of the *Explainer* based on the XAI technique, which is a crucial aspect when dealing with large datasets.

4.2.2 Explainer setup. There are two main categories of XAI techniques to compute explanations within a sentence: permutation-based and gradient-based [14]. For this comparison, we instantiated the *Explainer* component with *SHapley Additive exPlanations* (SHAP) [44] as a representative of the permutation-based and with *Integrated Gradients* (IG) [69] of the gradient-based techniques. Both techniques have demonstrated competitive performance in prior studies [3]. Specifically, for SHAP, we used the *text permutation explainer* with 3,000 as the maximum evaluation step parameter. For Integrated Gradients, we exploited the implementation provided by the Ferret [5] library.

4.2.3 Results. We produced the explanations within each sentence over the toxic texts in the test set using both techniques (SHAP and IG); we then aggregated individual scores and extracted an ordered list of the most toxic words as previously described in §3-A. Figure 4 shows the decrease in the F1 score by removing the most important words in the range from 50 to 700 with a step of 50. As expected, removing the most important words from the test set causes a marked decrease in predictive performance, especially for the top 250 words. IG exhibits higher precision, leading to a more substantial decrease initially. However, the decrement tends to converge on the top 400 words for both techniques. This shows that both techniques effectively extract the words used by

[3] https://github.com/unitaryai/detoxify
[4] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data
[5] The ground truth labels are available only for a subset of the test set (42%).

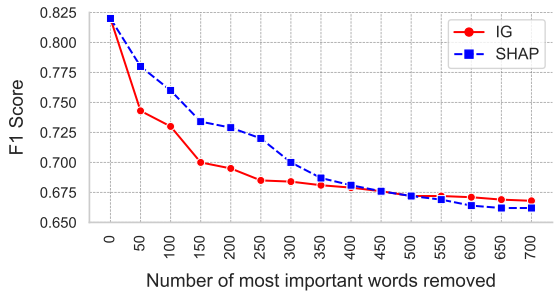

Fig. 4. **Explainer component evaluation.** F1 score decrease by removing the most important words from the test set, extracted by the *Explainer* component with Integrated Gradients (IG) and SHAP techniques. A greater decrease indicates a higher precision in identifying the most important words for predictions.

the classifier for its predictions. We selected the top 400 most toxic words (approximately 10%) – removing additional words caused a lower decrease in predictive performance.

We then measured the overlap between the 400 most toxic words identified with IG and SHAP. We found that 307 out of the 400 words were identical (77%), indicating a substantial agreement between the two techniques. The 23% of disagreement may be attributable to the varying precision levels inherent in the two methods.

Finally, we compared the execution times required to generate explanations using both techniques. We observed that IG significantly outperformed SHAP, completing the explanation process more than two orders of magnitude faster. On average, the execution time in seconds to obtain an explanation is approximately 0.2 for IG and 30 for SHAP, using a single Nvidia RTX A6000 GPU.

In summary, Integrated Gradients proves to be the most effective technique for the *Explainer* component, as it exhibits higher precision in identifying the most crucial words and executes significantly faster. As a result, we adopt Integrated Gradients in the *Explainer* for all subsequent experiments. Nevertheless, the framework allows the use of alternative XAI techniques.

4.3 Component evaluation: Identifier

The *Identifier* aims to determine which of the most important words are actually protected attributes (§3-B). To evaluate its effectiveness, we compare the protected attributes identified by the component instantiated with human-in-the-loop and machine-in-the-loop approaches against the annotations provided by two expert annotators, who possess a greater depth of knowledge of the definitions of protected categories by AI regulators than participants engaged in the human study.[6] We also add for comparison a pre-defined dictionary of 51 protected attributes from previous works [6, 20].

4.3.1 Evaluation metrics. We measure the Cohen's kappa inter-annotator agreement [11] to evaluate the accuracy and reliability of the protected attributes identified by the different approaches. It ranges between $[0, 1]$; the higher the score is, the higher the agreement.

4.3.2 Protected attributes identifier setup. We selected the 400 most toxic words extracted with the *Explainer* component instantiated with Integrated Gradients as the candidate set to identify protected attributes. Then, we configured the *Identifier* as follows.

[6]Expert annotators are people within our team with a background in human-computer interaction and trustworthy and responsible AI. They are located in two different Western countries, with different ethnicities and ages. They carefully read the UK Equality Act 2010 and unanimously agreed on the annotation to be performed, which was done independently.

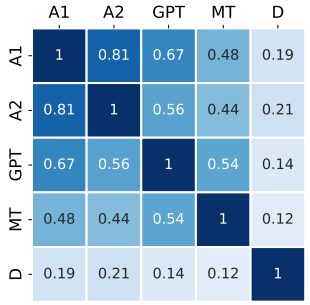

Fig. 5. **Identifier component evaluation.** Cohen's kappa annotator agreement in labeling protected attributes for the 400 most toxic words. The annotation was performed by two expert annotators (A1 and A2), ChatGPT (GPT), MTurk (MT), and a pre-defined dictionary (D). The two-by-two Cohen's kappa annotator agreement is reported, in the range [0, 1], where a higher score indicates a higher level of agreement.

Human-in-the-loop setup. We set up an MTurk study where we asked annotators to label words with the protected category they refer to (if any). As anticipated in §3-B1, we also used a trap mechanism to detect poor quality responses; specifically, we gave annotators the following definition of toxicity: *"Toxic language is a way of communicating that harms other people"*. Then, for each word, we also asked participants to answer the following additional trap question:

> **Trap Question:** *Does the word [WORD] suggest toxic language?*
> **Possible Answers:** 1. *Not at all*, 2. *Very little*, 3. *Somewhat*, 4. *To a great extent*, 5. *Definitely*.

To the list of 400 most toxic words, we added 15 trap words that can be easily classified as *toxic* (e.g., "asshole") or *non-toxic* (e.g., "friendly"). The full list of trap words can be found in Table 5 in Appendix B. For the *non-toxic* (*toxic*) trap words, we expected MTurk participants to select a score of 1 or 2 (4 or 5) on the Likert scale. Participants were considered unreliable if they did not meet those expectations, and their assessments were discarded from our results. We ended up with 246 reliable participants, evenly split between males and females. The majority were educated (74% finished college), mostly located in the United States, falling within the median age group of 26-39. In terms of racial demographics, most were White (52%), followed by Asian (27%), African (7%), and Hispanic (5%).[7]

We collected five annotations per word on average. A word was labeled as a protected attribute if the sum of the votes across the nine categories exceeded the *None of the above* (majority voting).

Machine-in-the-loop setup. We also annotated the same set of words by prompting GPT-3.5-Turbo, as introduced in §3-B2. The temperature parameter was set to 0.3 to limit creativity in generating the responses. Other temperature values in the range [0.3, 0.7] have been experimented with, although no major differences were observed. For each candidate word, we prompted GPT-3.5-Turbo, asking for a possible protected category, a reliability score, and an explanation for the classification. If a word is classified with any protected category, it is labeled as a protected attribute.

4.3.3 Results. The two expert annotators (A1, A2) identified 72/400 (18%) and 66/400 (17%) protected attributes, respectively. The human-in-the-loop (MTurk) approach 108/400 (27%) ones. Instead, the machine-in-the-loop (ChatGPT) approach labeled 93/400 (23%) words as protected attributes. These findings indicate that the original classifier heavily relies on protected attributes for toxicity predictions. Interestingly, the ChatGPT *Identifier* is also able to annotate proxy words for the

[7]Crowdworkers have been paid for their valuable contributions and time devoted to this research.

protected categories. For example, the word *'headscarf'* is annotated as related to *religion and belief* (see Figure 10 in Appendix A). If we use the pre-defined dictionary [6, 20] instead of the *Identifier* component, only 9 out of 400 words (2%) would have been labeled as protected attributes. This suggests that pre-defined dictionaries may consist of a limited subset of protected attributes and not encompass the entire range of relevant attributes.

Figure 5 shows the two-by-two Cohen's kappa inter-annotator agreement between the annotation performed by the two experts (A1, A2), ChatGPT (GPT), MTurk (MT), and the pre-defined dictionary (D). A higher score indicates a greater level of agreement. The score between the two expert annotators is 0.81, corresponding to an almost perfect agreement according to Landis and Koch's scale [41]. The ChatGPT annotations demonstrate substantial (0.67) and moderate (0.56) agreement with the expert annotators. In contrast, the MTurk annotations show a moderate agreement of 0.48 and 0.44 with the expert annotators, respectively. The pre-defined dictionary exhibits low agreement with all other annotations, as it only covers a small subset of the important words.

This evaluation demonstrates that the machine-in-the-loop approach outperforms the human-in-the-loop one in identifying protected attributes, while also enabling the full automation of the framework. For the next experiments, we thus adopt the LLM-based approach for the *Identifier*.

4.4 Component evaluation: Moderator

The *Moderator* aims to create a mitigated training corpus to train a new classifier with reduced reliance on protected attributes and similar predictive performance. To evaluate the effectiveness of each mitigation strategy, we trained and evaluated a distinct mitigated model for each strategy.

4.4.1 Evaluation metrics. For each mitigation strategy, we examine two key aspects of the mitigated classifiers: *fairness* and *predictive performance*. Our *fairness* is defined as *fairness through unawareness*, whereby "an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process" [45]. This is quantified by measuring the number of protected attributes each mitigated model relies on in making predictions based on the *Explainer* and *Identifier* components. A lower number indicates a reduced dependence on protected attributes, which signifies progress towards a more fair and unbiased classifier. To evaluate the *predictive performance*, we measure the F1 score specifically for the toxicity label, providing insight into the model's accuracy in identifying toxic texts. Additionally, we evaluate the Area Under the Curve (*AUC*) score for all toxicity-related labels. This metric provides an overall measure of the model's performance in identifying various aspects of toxicity. By considering both *fairness* and *predictive performance*, we can ascertain the effectiveness of the mitigated models in achieving a balance between reducing reliance on protected attributes and maintaining similar predictive capabilities.

4.4.2 Moderator setup. For the mitigation strategies outlined in §3-C, we used the following setup: for the removal-based strategies (MS1, MS2), we removed the sentences or the words if, after the tokenization, the protected attributes are in the list of tokens. In the case of the mitigation strategies based on the k-neighbours (MS3, MS4), we set the value of k to 5, meaning that for each protected attribute, the five closest words were identified. To identify these nearest neighbors, we computed the cosine similarity between each word in the vocabulary and the protected attribute using the 300-dimensional GloVe [55] word embedding, as suggested in [6]. For the hypernyms-based strategy (MS5), we utilized the WordNet lexical database [46] provided in NLTK,[8] as suggested in [6]. We replaced each protected attribute with its first-level hypernym extracted from its synset of synonyms.

[8] https://www.nltk.org/howto/wordnet.html

Table 1. **Results in mitigating toxicity prediction on in-distribution data (test set).** The original model is highlighted in grey (M_o). For each mitigated model are reported: (1) the model identifier, (2) the mitigation strategy applied, (3) the difference in training examples after the mitigation strategy, (4) the F1 macro and weighted scores for the toxicity class on the original test set, (5) the *AUC* score for all toxicity-related labels on the original test set, and (6) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in curly brackets. The best performing for each metric is in bold.

Model ID	Mitigation Strategy	Δ Train Examples	*Predictive Performance* ↑			*Fairness* ↓	
			F1 Macro	F1 Weight	AUC	% PA	Ratio PA
M_o	-	-	0.815	0.926	**0.986**	23%	93/400
M_1^*	*MS1 - Sentence removal*	-6k	0.816	0.934	0.981	**9%**	**37/400 {16}**
M_2^*	*MS2 - Word removal*	-	**0.828**	**0.938**	0.981	10%	40/400 {21}
M_3^*	*MS3 - Word replace 1 rand syn*	-	0.812	0.926	0.983	14%	56/400 {36}
M_4^*	*MS4 - Word replace k rand syn*	+108k	0.783	0.908	0.981	20%	80/400 {50}
M_5^*	*MS5 - Word replace hyper*	-	0.812	0.927	0.983	13%	51/400 {32}

4.4.3 Training the mitigated models. We applied each mitigation strategy to the original Wikipedia comments training dataset. Each mitigation strategy produced a modified version of the training dataset (containing 159,571 examples), whose differences are shown in the third column of Table 1. The sentence-removal mitigation strategy (MS1) resulted in a decrease of 6k examples. Instead, the strategy that added k new sentences for each protected attribute (MS4) increased the training dataset by 108k new sentences. The other mitigation techniques did not change the number of training examples. All mitigated models (M_1^*-M_5^*) were trained by fine-tuning the original pre-trained weights of BERT[9] for 3 epochs, with a batch size of 16, and Adam [36] as optimizer.

To evaluate the mitigated models, we first classified all texts in the test set with each *mitigated* model. Then, we applied the *Explainer* to extract the most important predictive words used by each *mitigated* model for the toxicity predictions in those texts. Finally, we exploited the *Identifier* to determine if the new important words of the mitigated models were protected attributes.

4.4.4 Results.

Fairness. The last two columns in Table 1 show the percentage and number of the most toxic words labeled as protected attributes for all the mitigated models (M_1^*-M_5^*). The number of those already present among the protected attributes of the original model (M_o) is indicated in curly brackets. All the mitigation strategies reduced the number of protected attributes the model relied upon. However, the mitigated models trained with removal-based strategies (MS1, MS2) achieved much better results. Only 9% and 10% of their most toxic words were labeled as protected attributes (37 and 40 words out of 400), representing a decrease of 61% from the original model (93 words out of 400). One possible reason for the lower performance of replacement-based mitigation strategies (MS3-MS5) is that they can introduce new protected attributes when replacing words. For a qualitative evaluation of fairness improvement, please refer to Figure 11 and Figure 12 in Appendix C.

Predictive performance. In Table 1, columns 4 and 5 report the macro and weighted F1 scores on the toxicity label for the original and mitigated models. Column 6 also presents the mean *AUC* scores across all the toxicity-related labels. The results show that all the mitigated models present similar F1 scores compared to the original model, except for the one trained with MS4, which exhibits a greater decrease. Interestingly, the mitigated models trained on the removal-based mitigation strategies (MS1, MS2) achieve better F1 scores than the original model. The word-removal (MS2) increases the macro and weighted F1 scores by 1.3% and 1.2%. Indeed, we observed that the removal-based mitigation strategies reduced the number of false positives in the toxicity predictions

[9] https://huggingface.co/bert-base-uncased

(i.e., non-toxic texts wrongly predicted as toxic). All the mitigated models exhibit slightly lower *AUC* scores compared to the original model. However, the decrease is minor and acceptable (around 0.5% and 0.3%) in light of the reduced reliance on protected attributes.

Summary. We conclude that our framework effectively reduces the model's reliance on protected attributes without compromising its predictive performance. Indeed, all mitigated models are fairer in that they significantly reduce the use of protected attributes and exhibit similar predictive performance to the original model. Interestingly, the removal-based strategies (MS1, MS2) even increased the models' predictive performance after the mitigation.

5 FRAMEWORK EVALUATION: GENERALIZABILITY

We finally evaluate the generalizability of our framework first to toxicity prediction on out-of-distribution data (§5.2), and second on different tasks, i.e., sentiment analysis (§5.3) and occupation classification (§5.4).

5.1 Framework and evaluation settings

For this evaluation, we instantiated the *Explainer* with Integrated Gradients, the *Identifier* with ChatGPT, and the *Moderator* with the removal-based mitigation strategies. As shown in §4, this turned out to be the optimal framework configuration. We perform a similar evaluation of the mitigated models by measuring their *fairness* and *predictive performance*. *Fairness* is evaluated by quantifying the number of protected attributes each mitigated model relies on (*fairness through unawareness*). *Predictive performance* is evaluated using quantitative metrics on the test set. For the toxicity classifier, we measure the F1 score for the toxicity label, which allows us to gauge the model's accuracy in detecting instances of toxicity, and the Area Under the Curve (*AUC*) score for all toxicity-related labels, providing an overall assessment of the model's performance in identifying different aspects of toxicity. For the sentiment and the occupation classifiers, we solely measure the F1 score, as it provides a comprehensive assessment of the model's accuracy in these tasks.

5.2 Mitigating toxicity prediction on out-of-distribution data

This experiment aims to assess the applicability of our framework in mitigating the toxicity model when applied to out-of-distribution data, specifically company reviews. This is crucial as classifiers are normally applied to datasets from other domains with different word distributions from training.

5.2.1 Company reviews data. We collected data from a popular online platform where current and former employees write reviews about companies. Reviewers comment on various aspects such as personal experience with the company or managers, salary information, workplace culture, and typical job interviews. The platform fosters a constructive approach among its users by manually and automatically moderating the content of reviews. However, reviews are published anonymously. On the one hand, this promotes user privacy. On the other hand, it can also cause some users to write public insults and offenses toward companies or people. Specifically, we collected a dataset of 439,163 reviews from U.S.-based companies across all 51 U.S. states written from 2008 to 2020.[10] Each review contains a *pros* part (positive comments in the review) and a *cons* part (negative comments). We applied the same toxicity classifier introduced in §4.1 to identify toxic company reviews.

5.2.2 Toxicity in company reviews. The initial expectation was not to have many toxic reviews in the dataset due to the highly curated nature of the platform. However, if we consider a post to be *toxic* when at least one of the *cons* or *pros* fields contains inappropriate content, we found 1.6% of

[10]To preserve the privacy of individuals, Personally Identifiable Information (PII) was removed.

Table 2. **Results in mitigating toxicity prediction on out-of-distribution data (company reviews).** The original model is highlighted in grey (M_o). For each mitigated model are reported: (1) the model identifier, (2) the mitigation strategy applied, (3) the difference in training examples after the mitigation strategy, (4) the F1 macro and weighted scores for the toxicity class on the original test set, (5) the *AUC* for all toxicity-related labels on the original test set, (6) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in curly brackets. The best performing for each metric is in bold.

Model ID	Mitigation Strategy	Δ Train Examples	*Predictive Performance* ↑			*Fairness* ↓	
			F1 Macro	F1 Weight	AUC	% PA	Ratio PA
M_o	-	-	0.815	0.926	**0.986**	19%	76/400
M_1^*	*MS1 - Sentence removal*	-6k	0.824	**0.938**	0.979	**4%**	**16/400 {8}**
M_2^*	*MS2 - Word removal*	-	**0.825**	0.935	0.983	5%	19/400 {11}

reviews (7,224) to be toxic. The number of reviews classified as toxic by using the *pros* and *cons* texts as input is 853 for *pros* (0.2%) and 6,495 for *cons* (1.5%) over 439,163. As expected, we found that most of the toxic texts are present in *cons*. Interestingly, some people tend to be so angry and frustrated by the work experience that they let off steam even in the *pros* field.

5.2.3 Identify protected attributes in toxicity predictions on company reviews. All *pros* and *cons* reviews predicted as *toxic* were analyzed by the *Explainer* component to extract the most important words used by the model in predicting toxic reviews. Then, we selected the 400 most toxic words extracted, and we annotated those words with GPT-3.5-Turbo. Among the 400 most important words used by the model in predicting toxic reviews, 76 are protected attributes (19%), as shown in the last two columns of the first row in Table 2 (original model M_o). We can conclude that the original classifier exhibits a significant reliance on protected attributes for toxicity predictions, even when applied to different out-of-distribution data.

5.2.4 Training the mitigated models. We applied the removal-based mitigation strategies (MS1, MS2) to the original Wikipedia comments training dataset based on the protected attributes identified in the toxic company reviews. Table 2 shows the differences in the number of training examples after each strategy in the third column. The original training dataset contained 159,571 examples. MS1 resulted in a decrease of 6k examples, while MS2 did not change it. All mitigated models were fine-tuned for 3 epochs, with a batch size of 16, and Adam as optimizer. To evaluate the mitigated models, all *pros* and *cons* reviews were classified by each *mitigated* model. Then, we applied the *Explainer* component to extract the most important 400 predictive words used by each *mitigated* model for the toxicity predictions on company reviews. Finally, we exploited the *Identifier* to determine if the new important words of the mitigated models were protected attributes.

5.2.5 Results.

Fairness. The last two columns in Table 2 show the percentage and number of the most toxic words labeled as protected attributes. The number of those already present among the protected attributes used by the original model (M_o) is also indicated in curly brackets. The results confirm that removal-based mitigation strategies reduce the number of protected attributes the model relied upon. MS1 and MS2 reduce the percentage of protected attributes from 19% to 4% and 5% (16 and 19 out of 400 words), respectively. This corresponds to a decrease of 79% and 75%. They also reduce the protected attributes the original classifier relies on from 76 to 8 and 11, respectively.

Predictive performance. Columns 4 and 5 in Table 2 show the macro and weighted F1 scores achieved by the original and mitigated models on the test set. The mitigated models exhibit higher predictive performance in terms of F1 scores than the original model. The increment is around 1%

for both scores and mitigated models. Finally, column 6 shows the *AUC* for all the toxicity-related labels. The mitigated model produced by MS1 achieves 0.979 on the *AUC* score, with a decrease of 0.007 from the original model. Instead, with MS2, the decrease in performance is only 0.003.

Summary. The experimental results obtained from the out-of-distribution data demonstrate the capability of our framework to effectively mitigate a model's reliance on protected attributes when applied to non-training data, where ground truth labels are unavailable. It showcases its adaptability and robustness in real-world scenarios where labeled data may not be readily accessible.

5.3 Mitigating sentiment analysis

This evaluation aims to assess the versatility and effectiveness of our framework across different classification tasks. For this experiment, we chose sentiment classification to test our framework in mitigating the use of protected attributes for tasks where their reliance might be lower.

5.3.1 Training the original sentiment classifier. We selected a dataset of 163K tweets and 37K Reddit comments in English, expressing people's opinions towards the general elections held in India in 2019.[11] The task consists of a multi-class sentiment classification problem with 3 classes: *negative*, *neutral*, and *positive*. We split the dataset with 80% for training (160k) and 20% for testing (40k). We fine-tuned the BERT model for 3 epochs, achieving a 0.96 F1 score on the test set.

5.3.2 Identifying protected attributes in sentiment predictions. We used the fine-tuned model to predict the sentiment label over the entire test set. Then, we analyzed, separately, all the *negative* and *positive* texts with the *Explainer* component instantiated with Integrated Gradients. The *neutral* texts do not contain specific patterns that the model should learn and are not of interest for mitigation. Then, we annotated with GPT-3.5-Turbo the 5% of the most important words for the *negative* and 5% for the *positive* texts separately, resulting in the top 200 negative and 200 positive words. We found that 16 (8%) of 200 negative words and 11 (6%) of 200 positive words were labeled as protected attributes by the Identifier, suggesting a moderate reliance on protected attributes.

5.3.3 Training the mitigated models. We applied the two removal-based mitigation strategies (MS1, MS2). In this case, the mitigation is performed separately per class label (e.g., the protected attributes in the most negative words are mitigated only on the negative training examples). MS1 decreases the training set by 5k *negative* and 8k *positive* training examples, as shown in the third and fourth columns in Table 3. Also in this case, the models were fine-tuned for 3 epochs.

We used the *mitigated* models to predict the sentiment label over the test set. Then, we extracted the most important 200 words from the *negative* and *positive* texts separately with the *Explainer*, and we annotated those words with the *Identifier*.

5.3.4 Results.

Fairness. The last four columns in Table 3 show the percentage and the number of protected attributes of the original (M_o) and mitigated (M_1^* and M_2^*) sentiment classifiers for the *negative* and *positive* classes, separately. MS1 produces a mitigated model that relies on half of the protected attributes of the original classifier (4% for the *negative* and 3% for the *positive* classes). Interestingly, the number of protected attributes the original classifier relied on is almost completely mitigated, except for 2 for the *negative* and 1 for the *positive* classes. MS2 has a similar behavior in this. However, many new protected attributes emerge as new important words. In the end, the total number of protected attributes remains the same, even though the protected attributes of the original model have almost all been mitigated. Therefore, MS1 is the most effective in this case.

[11] https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset

Table 3. **Results in mitigating sentiment analysis.** The original model is highlighted in grey (M_o). For each mitigated model are reported: (1) the model identifier, (2) the mitigation strategy applied, (3) the difference in training examples after the mitigation strategy for the *negative* and *positive* classes, (4) the F1 macro score on the original test set, and (5) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in curly brackets. The best performing for each metric is in bold.

Model ID	Mitigation Strategy	Δ Train Examples		*Predictive Performance* ↑	*Fairness* ↓			
					Negative Class		*Positive Class*	
		Negative	*Positive*	**F1**	**% PA**	**Ratio PA**	**% PA**	**Ratio PA**
M_o	-	-	-	0.96	8%	16/200	6%	11/400
M_1^*	*MS1 - Sentence removal*	-5k	-8k	0.96	**4%**	**8/200 {2}**	**3%**	**5/200 {1}**
M_2^*	*MS2 - Word removal*	-	-	0.96	8%	16/200 {2}	6%	11/200 {2}

Predictive performance. The fifth column in Table 3 shows the F1 score obtained by the original sentiment classifier and the mitigated models on the test set. The mitigated models achieve the same F1 score, thus showing the same predictive capabilities.

Summary. These results show that our framework can be effective not only on the toxicity model, which heavily relies on protected attributes, but also on the sentiment model, which is moderately impacted by protected attributes, confirming its general effectiveness.

5.4 Mitigating occupation classification

This evaluation serves three primary objectives: (1) to further assess the adaptability and effectiveness of our framework across various classification tasks, (2) to mitigate the use of protected attributes in scenarios where the final prediction has tangible consequences for individuals, and (3) to compare our framework with two mitigation techniques that act on the model rather than data.

To achieve these goals, we selected the task of predicting occupations from online biographies, using a dataset of biographies annotated by gender and occupation from previous works [17, 54, 58]. We used the field *'cleaned_input_text'* as input text, where sentences that directly reveal the occupation were removed (e.g., "he is a journalist"), and we removed first names. We compare our framework with two model-based mitigation techniques: *Iterative Null-space Projection (INLP)* [58] and *Entropy-based Attention Regulation (EAR)* [4]. *INLP* requires an additional annotation for the mitigated category, which is only present for gender in this dataset. Therefore, we conduct two distinct analyses: (1) focusing solely on gender-related, and (2) examining all protected categories together. For the gender-related protected attributes, we compare both baselines with our word-removal (MS2). Instead, we use only *EAR* as a baseline to mitigate all the protected categories. We chose MS2 because it has shown similar mitigation effectiveness while maintaining competitive performance, but it has higher flexibility across datasets than sentence-removal (MS1) (see §6.3).

5.4.1 Training the original occupation classifiers. The dataset contains 393,423 biographies for 28 occupations split into 255,710, 39,369, and 98,344 train, dev, and test examples. We fine-tuned a BERT-base and uncased model for each occupation in one-vs-all settings (i.e., a binary model that predicts the occupation or not for each label). Due to the high imbalance of the dataset, we performed the experiments for the five most frequent classes (i.e., *nurse*, *attorney*, *journalist*, *physician*, and *professor*). The models were fine-tuned for 3 epochs.[12] The second column in Table 4 shows the macro F1 score on the test set for each original model trained for four occupations.[13] Those models are highly effective in classifying occupations, achieving a macro F1 score higher than 0.89. Still, such high performance could be achieved by heavily using protected attributes.

[12]We utilized inversely proportional class weights in the loss function due to the highly imbalanced training dataset.
[13]Results for the *professor* occupation are not reported since the model do not rely on gender-related protected attributes.

Table 4. **Results in mitigating occupation classification.** For the *original* models (highlighted in grey), trained in one-vs-all settings, are reported the macro F1 and the reliance on protected attributes (PA) for gender only and all categories. For each occupation, we applied our framework with the word-removal strategy (MS2) on gender-related only and all categories of protected attributes training two different *mitigated* models. For *EAR* [4], a single model was trained but evaluated on different protected categories. *INLP* [58] is only applicable to gender-related protected attributes in this dataset. For each mitigated model are reported: (1) the mitigation technique, (2) the macro F1 score for the occupation classification on the test set (*predictive performance*), and (3) the ratio and percentage of relied-upon protected attributes, with the number of those present in the original model in curly brackets (*fairness*). The best performing for each metric is in bold.

Occupation	*Original Model*			**Mitigation Technique**	*Mitigated Models*			
		Gender only	*All Categories*			*Gender only*		*All Categories*
	F1 ↑	**Ratio (%) PA** ↓	**Ratio (%) PA** ↓		**F1** ↑	**Ratio (%) PA** ↓	**F1** ↑	**Ratio (%) PA** ↓
Nurse	0.939	11/400 (3%)	43/400 (11%)	*Our - MS2*	**0.932**	**2/400 {1} (0.5%)**	0.930	**27/400 {18} (7%)**
				INLP	0.762	**2/400 {2} (0.5%)**	*N.A.*	*N.A.*
				EAR	**0.932**	4/400 {3} (1.0%)	**0.932**	**27/400 {18} (7%)**
Attorney	0.943	2/400 (0.5%)	18/400 (5%)	*Our - MS2*	**0.942**	**0/400 {0} (0%)**	**0.943**	**8/400 {3} (2%)**
				INLP	0.702	1/400 {0} (0.3%)	*N.A.*	*N.A.*
				EAR	0.940	**0/400 {0} (0%)**	0.940	11/400 {1} (3%)
Journalist	0.886	3/400 (0.8%)	32/400 (8%)	*Our - MS2*	**0.887**	2/400 {2} (0.5%)	**0.887**	**18/400 {12} (4.5%)**
				INLP	0.528	**1/400 {0} (0.3%)**	*N.A.*	*N.A.*
				EAR	0.886	**1/400 {0} (0.3%)**	0.886	21/400 {9} (5.3%)
Physician	0.936	2/400 (0.5%)	24/400 (6%)	*Our - MS2*	0.939	**0/400 {0} (0%)**	0.939	**16/400 {3} (4%)**
				INLP	0.823	**0/400 {0} (0%)**	*N.A.*	*N.A.*
				EAR	**0.941**	1/400 {0} (0.3%)	**0.941**	28/400 {12} (7%)

5.4.2 Identifying protected attributes in occupation classification. We used each original model to predict each occupation label over the entire test set and analyzed those texts using the *Explainer* to extract the most important words in predicting each occupation. Then, for each occupation, we annotated with GPT-3.5-Turbo the top 400 words to identify protected attributes. We measured the models' reliance on protected attributes related to (1) gender only, and (2) all categories (third and fourth columns in Table 4). All the models moderately rely on protected attributes. The *nurse* occupation is the most influenced, also by gender-related words, such as pronouns (e.g., *'she'*, *'her'*).

5.4.3 Training the mitigated models. With our framework, we applied the word removal mitigation strategy (MS2) for each occupation on (1) gender-related only, and (2) all the protected categories simultaneously. Therefore, we trained two different mitigated models for each occupation. We trained one mitigated model for each occupation with the *INLP* methodology [58]. *INLP* can mitigate the gender-related protected attributes but is not applicable to all the other categories, since the dataset contains the additional annotation only for gender. Specifically, we used the original pre-trained BERT weights as the encoder. Then, we multiplied the embedding representation of the [CLS] token from the last hidden layer for each input text by the projection matrix produced by the *INLP* technique (to ensure that the embedding representation does not encode information about gender). Finally, we added a classification layer on top. We fine-tuned only the classification layer while freezing the BERT encoder and the projection matrix, as suggested in [58]. For *EAR* [4], we used the same BERT architecture, and we added the loss function regularization term, with 0.001 as regularization strength. EAR does not allow the selection of which protected categories to mitigate, but it identifies by itself which words have a high attention entropy. Thus, we trained a single mitigated model, and we evaluated its reliance on gender and all protected categories separately.

We used the mitigated models to predict the occupation labels over the entire test set, we extracted the most important 400 words for each occupation separately with the *Explainer*, and we annotated those words with the *Identifier* to evaluate if they exhibit a reduced reliance on protected attributes.

5.4.4 Results.

Fairness. Columns 7 and 9 in Table 4 show the number and percentage of protected attributes on which the mitigated models rely for gender only and all categories separately. The number of these words already present among the ones used by the original model is also indicated in curly brackets. The objective of each *mitigated* model is to reduce the reliance on protected attributes (columns 7 and 9) compared to the respective original model for each occupation (columns 3 and 4).

Concerning gender (column 7), our framework is the most or as effective as other techniques in mitigating the use of such protected attributes. For the *nurse* occupation, which represents the model showing a greater dependence on gender-related protected attributes, our framework reduced the number of the most significant gender-related words from 11 to 2. One of these two gender-related words was already significant in the original model. *INLP* obtained a similar mitigation effect, while *EAR* is less effective, with 4 gender-related protected attributes still important for the mitigated model. For the *attorney* and *physician* occupations, the original model exhibited a lower reliance on gender-related protected attributes, and our framework was able to fully mitigate the use of those words. On average, our framework reduces reliance on gender-related protected attributes by 79%.

The results are similar when considering all protected categories (column 9). Our framework is always more effective than *EAR* in mitigating the use of protected attributes, except for the *nurse* occupation, where both techniques achieve the same mitigation effect. On average, our framework successfully reduces reliance on all categories of protected attributes by 43%.

Predictive performance. Columns 6 and 8 in Table 4 show the macro F1 score achieved on the test set by each mitigated model. The objective of each mitigated model is to achieve similar predictive performance (columns 6 and 8) compared to the respective original ones (column 2). The mitigated models produced by our framework and the *EAR* technique achieve similar or sometimes even better performance than the original model (e.g., for the *journalist* and *physician* occupations). Therefore, they are able to mitigate the use of protected attributes without sacrificing predictive performance. Instead, the *INLP* technique is able to produce models with mitigated bias at the cost of significantly reducing their performance. Indeed, all the mitigated models produced by *INLP* experienced an average loss in predictive performance of 10%. This tendency to achieve fairness by making every advantaged group worse off or by bringing better performing groups down to the level of the worst off is a common undesirable behavior of bias mitigation techniques [48].

Summary. These results confirm the effectiveness of our framework in a different task where the protected attributes are strictly related to individuals. They show that our framework is more effective in achieving such an objective than previous bias techniques, while also providing the flexibility to select which protected category to mitigate. This flexibility enables the mitigation of only a subset of protected categories when some are required for the task at hand.

6 DISCUSSION

Our results show how the proposed framework could be exploited to train a new classifier that mitigates the use of protected attributes while maintaining competitive performance. We evaluated its sensitivity to each component and its effectiveness in mitigating a toxicity classifier. We also proved its generalizability on models applied to out-of-distribution data (i.e., toxicity on company reviews) and two other tasks (i.e., sentiment analysis and occupation classification). Removal-based strategies (MS1, MS2) have been shown to be the most effective mitigation techniques. We also show that the LLM-based *Identifier* outperforms the crowdsourcing one, allowing the automation of the framework, and a dynamic updating of the dictionary of protected attributes.

6.1 Framework integration, versatility and complexity

Integration. Our framework can be integrated into existing NLP pipelines for two main purposes. (1) The *Explainer* and *Identifier* can be used to measure and evaluate existing NLP classifiers' reliance on protected attributes. As a result, models can be quantitatively compared not only through predictive performance and traditional fairness metrics (e.g., conditional demographic disparity [74]) but also through the use of protected attributes in predictions. (2) The entire framework can be used to train a new model with reduced reliance on protected attributes and competitive performance. Our framework can also be integrated to complement other bias mitigation techniques acting in both the model and data spaces that require pre-defined dictionaries or lists of protected attributes or identity terms. The *Explainer* can improve existing techniques to pinpoint the specific words the model mostly uses for the classification rather than looking at all possible words in the corpus. The *Identifier* can annotate protected attributes covering a broader range of categories, as many protected attributes, such as disability or religious belief, were rarely covered by prior studies.

Versatility. Our framework is designed to achieve *fairness through unawareness* by mitigating the model's reliance on protected attributes in predictions. It addresses multiple categories simultaneously. Therefore, it can potentially address intersectional bias, i.e., that encompasses multiple sensitive attributes together [24]. Moreover, it provides users the flexibility to choose which categories to mitigate, thanks to the fine-grained annotation performed by the *Identifier*. Such flexibility is particularly useful when some categories are indispensable or aimed at the prediction task. Through this, our framework can address domain-specific bias related to the classification task (e.g., gender-related protected attributes in occupation classification). Consequently, in scenarios where the inclusion of certain protected attributes is necessary for accuracy, our framework can still be utilized to effectively mitigate all other protected attributes that are not essential for the task.

Complexity. The execution time to produce the mitigated training dataset depends on many factors. Given a fixed model complexity, the execution times increase linearly with: (1) the unlabeled corpus size, (2) the number of most important words annotated, and (3) the training dataset size. Increasing the model's complexity results in a slight increase in the execution time of all components. Our framework yields a mitigated training corpus, necessitating extra training to produce the mitigated model. The (re-)training time varies based on model complexity and original data size. Techniques such as MS2, MS3, and MS5 maintain dataset dimensionality, resulting in comparable training times for both original and mitigated models. Conversely, MS1 reduces or MS4 increases dataset size, affecting training time accordingly. An example of execution time is reported in Appendix D.

6.2 Implications

Our research significantly contributes to the CSCW community by exploring human-AI collaboration, especially in decision-making contexts. For example, our tool could assist humans in comprehending the hiring decisions made by NLP classifiers and address biases in the hiring process [53]. Our work extends into content moderation, empowering the development of robust systems capable of effectively identifying and mitigating toxic content while ensuring fairness. This aids humans in understanding crucial moderation aspects, encompassing significant words and considerations around protected attributes, to foster collaboration with machines to collectively arrive at fair decisions in content moderation. Our work has three main implications:

Fully-automated framework for compliant NLP classifiers. We release an open-source framework.[14] By leveraging LLM annotations, it operates in a fully automated manner. The mitigated models exhibit enhanced fairness by significantly reducing their reliance on protected attributes

[14]The code repository of our framework is available at https://github.com/grecosalvatore/nlpguard

while maintaining comparable or even better predictive performance. Other researchers can utilize our framework to address the compliance standards set by regulators, whether by mitigating already trained models or incorporating them into future models. This contribution empowers the community to uphold ethical standards and ensure fairness in NLP applications. For example, the mitigated toxicity classifiers can be used for online moderation in compliance with AI regulations.

Protected attributes annotation. We advanced the literature in the protected attributes identification in NLP, traditionally done with static and manually pre-defined dictionaries covering only a subset of categories. However, they are difficult to keep up-to-date, especially with the emergence of ever-evolving language trends and slang. In our framework, we demonstrated a novel approach to dynamically identify protected attributes through straightforward prompts to an LLM. This enables the creation of a comprehensive and up-to-date dictionary covering all the protected categories simultaneously, which can be updated periodically, ensuring its relevance in real-time linguistic landscapes.

In our research, we annotated 15,000 words, 540 labeled as protected attributes. We release our dictionary in the GitHub repo. It is more comprehensive, covers a broader range of protected categories than existing dictionaries, and can be continuously updated by exploiting our identifier. Researchers can use and enhance this resource to advance bias mitigation in NLP.

Humans *vs.* LLM annotations. Building upon a recent finding [23], our study demonstrates that LLM annotation can outperform human-in-the-loop crowdsourcing annotations. Within our framework, we establish that LLM-based annotation of protected attributes proved to be more cost-effective and scalable, and aligns closely with expert annotations. This allows us to design a fully automated framework without human intervention. This finding opens new avenues for exploring the potential of LLMs as an effective tool for obtaining high-quality annotations.

6.3 Limitations and Future Directions

Our current approach has six main potential limitations or areas of concern.

Context unawareness. Our *Identifier* and *Moderator* label and mitigate words related to protected attributes without considering context, simplifying annotation but risking inaccuracies. For instance, 'black' may be a protected attribute in one context (*"If you are a guy (black) or lesbian you get hired fast"*) but not in another (*"I bought a new black desk"*). Addressing context could enhance mitigation effectiveness yet poses challenges in the identification and mitigation phases. Human-in-the-loop context annotation is costly, requiring thousands of context-aware annotations, potentially increasing noise. Machine-in-the-loop is more scalable but complicates prompt engineering, potentially leading to misunderstandings and noisy responses [43]. We conducted a preliminary experiment assessing context-aware annotation's impact on identifying protected attributes with LLM. Repeating annotations with up to 10 context sentences, we found a 75% overlap between word-level and context-level annotations, with some contradictions, especially for long sentences. However, future research should explore this further across datasets.

Potential bias introduced by the Identifier. The annotation of protected attributes is a subjective task. Therefore, the *Identifier* can potentially introduce further sources of bias. In human-in-the-loop settings, crowdworkers should come from various backgrounds to have a broader contextual understanding during the annotation process. The distribution of the demographic backgrounds of crowdworkers can have an impact on the annotated protected attributes. It is important to ensure an equitable distribution of crowdworkers across all protected categories. However, this can often be challenging in practice. Instead, in the machine-in-the-loop settings, the *Identifier* can introduce potential bias inherent in the LLM system adopted. The LLM can associate certain words with

protected attributes based on stereotypes prevalent in the training data. However, addressing bias inherent in LLM is an active area of research expected to resolve numerous current limitations, significantly enhancing the effectiveness of our framework. Finally, introducing specific definitions of protected attributes, such as the nine categories defined by the UK Equality Act 2010, might also inadvertently introduce biases or overlook certain nuances in both human annotators and LLMs.

Reliance on XAI techniques. Our framework relies on XAI techniques to identify the most important words for the model. Nevertheless, it is important to acknowledge that XAI methods have inherent limitations [47, 65], including challenges like effective aggregation and normalization methods, and the contextual variability of words across different explanations. These limitations may hinder the accurate extraction of the most important words used by the model, affecting the *Identifier* in the identification of protected attributes that the model relies on to make predictions. This issue can extend to the *Moderator*, impacting the mitigated protected attributes. In the future, improvements in the XAI field could make our framework even more effective. This is because our framework is flexible and can use any feature importance explainability method that can be applied to a pre-trained classifier (e.g., Integrated Gradients or SHAP), as explained in §3-A.

Defined protected categories. Our framework annotates protected attributes based on the nine categories outlined in the Equality Act 2010. These categories represent a significant step toward addressing discrimination and promoting equality. However, they might not cover all aspects of human diversity or potential discrimination. Since they were formalized in 2010, some characteristics remain unaddressed by the Act. More than a decade later, initiatives are underway to broaden these categories for aspects like socio-economic status, health status, genetic heritage, and physical appearance [50]. Future extensions will further enhance the comprehensiveness of our framework in encompassing a broader range of protected categories. Notably, for the LLM-based annotation, incorporating new categories is a straightforward process that involves modifying the prompt.

Mitigation with small training datasets or common protected attributes. In scenarios with small or imbalanced training data, or when protected attributes are common in most input texts (e.g., 'he' and 'she' in biographies), sentence removal (MS1) may be less effective due to potential consequences of removing sentences containing protected attributes from already limited datasets. Frequent presence of common protected attributes in inputs may exclude most sentences, reducing available training data significantly. This reduction can cause a significant and unacceptable decrease in model accuracy. Hence, alternative strategies, like word-removal (MS2), should be considered. MS2 has similar effectiveness in mitigating protected attributes while maintaining predictive performance, offering flexibility across datasets without suffering from these issues.

Fairness-privacy tradeoff. Our approach neither protects the privacy of individuals nor considers words or sentences to be private. Instead, it focuses on constraining the classification so that does not rely on protected attributes. In the way that loans cannot be given by an automatic system that relies on racial backgrounds, natural language classification should not rely on protected attributes. The tradeoff between fairness and privacy becomes more pronounced when considering human- and machine-in-the-loop identifiers. The protected attributes annotation requires exposing textual sensitive information to individuals who are not necessarily trusted or to LLMs. This creates a risk of privacy violations and potential harm to the individuals whose sensitive information is being used [1]. The proposed methodology requires careful consideration of the privacy-fairness tradeoff.

In future work, we plan to develop a context-aware framework that would allow us to identify and mitigate protected attributes based on their context by extracting the words and context information from the dataset, identifying protected attributes within each context, and applying mitigation strategies only to those sentences that contain protected attributes in similar contexts.

REFERENCES

[1] Mohammad Al-Rubaie and J Morris Chang. 2019. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy* 17, 2 (2019), 49–58. https://doi.org/10.1109/MSEC.2018.2888775

[2] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining Predictions of Non-Linear Classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Berlin, Germany, 1–7. https://doi.org/10.18653/v1/W16-1601

[3] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-main.263

[4] Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based Attention Regularization Frees Unintended Bias Mitigation from Lists. Association for Computational Linguistics, Dublin, Ireland, 1105–1119. https://doi.org/10.18653/v1/2022.findings-acl.88

[5] Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. 2023. ferret: a Framework for Benchmarking Explainers on Transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

[6] Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical Bias Removal for Hate Speech Detection Task Using Knowledge-Based Generalizations. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 49–59. https://doi.org/10.1145/3308558.3313504

[7] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *CoRR* abs/1607.06520 (2016). arXiv:1607.06520 http://arxiv.org/abs/1607.06520

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 https://arxiv.org/abs/2005.14165

[9] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186. https://doi.org/10.1126/science.aal4230

[10] Alexandra Chouldechova and Aaron Roth. 2020. A snapshot of the frontiers of fairness in machine learning. *Commun. ACM* 63, 5 (2020), 82–89. https://doi.org/10.1145/3376898

[11] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. https://doi.org/10.1177/001316446002000104 arXiv:https://doi.org/10.1177/001316446002000104

[12] Equal Employment Opportunity Commission. 1977. Prohibited Employment Policies/Practices. https://www.eeoc.gov/prohibited-employment-policiespractices Accessed: June 2023.

[13] Kevin Crowston. 2012. Amazon Mechanical Turk: A Research Tool for Organizations and Information Systems Scholars. In *Shaping the Future of ICT Research. Methods and Approaches*, Anol Bhattacherjee and Brian Fitzgerald (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 210–221.

[14] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Suzhou, China, 447–459. https://aclanthology.org/2020.aacl-main.46

[15] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial Bias in Hate Speech and Abusive Language Detection Datasets. *CoRR* abs/1905.12516 (2019). arXiv:1905.12516 http://arxiv.org/abs/1905.12516

[16] Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *CoRR* abs/1703.04009 (2017). arXiv:1703.04009 http://arxiv.org/abs/1703.04009

[17] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 120–128. https://doi.org/10.1145/3287560.3287572

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[19] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083* (2019).

[20] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (New Orleans, LA, USA) *(AIES '18)*. Association for Computing Machinery, New York, NY, USA, 67–73. https://doi.org/10.1145/3278721.3278729

[21] Luciano Floridi, Matthias Holweg, Mariarosaria Taddeo, Javier Amaya Silva, Jakob Mökander, and Yuni Wen. 2022. capAI - A Procedure for Conducting Conformity Assessment of AI Systems in Line with the EU Artificial Intelligence Act. *SSRN Electronic Journal* (2022). https://doi.org/10.2139/ssrn.4064091

[22] 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 (2021). https://doi.org/10.3390/app11073184

[23] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056* (2023).

[24] Usman Gohar and Lu Cheng. 2023. A Survey on Intersectional Fairness in Machine Learning: Notions, Mitigation, and Challenges. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI '23)*. Article 742, 9 pages. https://doi.org/10.24963/ijcai.2023/742

[25] Hila Gonen and Yoav Goldberg. 2019. Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. *CoRR* abs/1903.03862 (2019). arXiv:1903.03862 http://arxiv.org/abs/1903.03862

[26] Bryce Goodman and Seth Flaxman. 2017. European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine* 38, 3 (Oct. 2017), 50–57. https://doi.org/10.1609/aimag.v38i3.2741

[27] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. 2016. The case for process fairness in learning: Feature selection for fair decision making. In *Proceedings of the NIPS Symposium on Machine Learning and the Law*, Vol. 1. 2.

[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5, Article 93 (aug 2018), 42 pages. https://doi.org/10.1145/3236009

[29] Matan Halevy, Camille Harris, Amy Bruckman, Diyi Yang, and Ayanna Howard. 2021. Mitigating Racial Biases in Toxic Language Detection with an Equity-Based Ensemble Framework. In *Equity and Access in Algorithms, Mechanisms, and Optimization* (–, NY, USA) *(EAAMO '21)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3465416.3483299

[30] Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

[31] Camille Harris, Matan Halevy, Ayanna Howard, Amy Bruckman, and Diyi Yang. 2022. Exploring the Role of Grammar and Word Choice in Bias Toward African American English (AAE) in Hate Speech Classification. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 789–798. https://doi.org/10.1145/3531146.3533144

[32] White House. 2023. Blue print for an AI Bill of Rights. https://www.whitehouse.gov/ostp/ai-bill-of-rights/#discrimination Accessed: June 2023.

[33] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. 2022. Simple data balancing achieves competitive worst-group-accuracy. In *Proceedings of the First Conference on Causal Learning and Reasoning (Proceedings of Machine Learning Research, Vol. 177)*, Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (Eds.). PMLR, 336–351. https://proceedings.mlr.press/v177/idrissi22a.html

[34] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (Sept. 2019), 389–399. https://doi.org/10.1038/s42256-019-0088-2

[35] United Kingdom. 2010. Equality Act 2010: guidance. https://www.gov.uk/guidance/equality-act-2010-guidance Accessed: June 2023.

[36] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[37] Kristin M Kostick-Quenet, I Glenn Cohen, Sara Gerke, Bernard Lo, James Antaki, Faezah Movahedi, Hasna Njah, Lauren Schoen, Jerry E Estep, and JS Blumenthal-Barby. 2022. Mitigating racial bias in machine learning. *Journal of Law, Medicine & Ethics* 50, 1 (2022), 92–100.

[38] Nikita Kozodoi, Johannes Jacob, and Stefan Lessmann. 2022. Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research* 297, 3 (2022), 1083–1094. https://doi.org/10.

1016/j.ejor.2021.06.023

[39] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 299–318.

[40] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf

[41] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[42] European Union Law. 2023. Proposal for a Regulation laying down harmonised rules on Artificial Intelligence and amending certain union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206 Accessed: June 2023.

[43] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL]

[44] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[45] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6, Article 115 (jul 2021), 35 pages. https://doi.org/10.1145/3457607

[46] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (nov 1995), 39–41. https://doi.org/10.1145/219717.219748

[47] Brent Mittelstadt, Chris Russell, and Sandra Wachter. 2019. Explaining Explanations in AI. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 279–288. https://doi.org/10.1145/3287560.3287574

[48] Brent Mittelstadt, Sandra Wachter, and Chris Russell. 2023. The Unfairness of Fair Machine Learning: Levelling down and strict egalitarianism by default. arXiv:2302.02404 [cs.AI]

[49] Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. 2016. The ethics of algorithms: Mapping the debate. *Big Data & Society* 3, 2 (2016), 2053951716679679. https://doi.org/10.1177/2053951716679679

[50] EQUINET European Network of Equality Bodies. 2022. EXPANDING THE LIST OF PROTECTED GROUNDS WITHIN ANTI-DISCRIMINATION LAW IN THE EU: AN EQUINET REPORT. https://equineteurope.org/expanding-the-list-of-protected-grounds-within-anti-discrimination-law-in-the-eu-an-equinet-report/ Accessed: January 2024.

[51] Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, and Emilia Gomez. 2023. The Role of Explainable AI in the Context of the AI Act. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (Chicago, IL, USA) *(FAccT '23)*. Association for Computing Machinery, New York, NY, USA, 1139–1150. https://doi.org/10.1145/3593013.3594069

[52] Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing Gender Bias in Abusive Language Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2799–2804.

[53] Andi Peng, Besmira Nushi, Emre Kiciman, Kori Inkpen, and Ece Kamar. 2022. Investigations of Performance and Bias in Human-AI Teamwork in Hiring. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 11 (Jun. 2022), 12089–12097. https://doi.org/10.1609/aaai.v36i11.21468

[54] Andi Peng, Besmira Nushi, Emre Kıcıman, Kori Inkpen, Siddharth Suri, and Ece Kamar. 2019. What You See Is What You Get? The Impact of Representation Criteria on Human Bias in Hiring. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 7 (Oct. 2019), 125–134. https://doi.org/10.1609/hcomp.v7i1.5281

[55] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[56] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? arXiv:2302.06476 [cs.CL]

[57] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR* abs/1910.10683 (2019). arXiv:1910.10683 http://arxiv.org/abs/1910.10683

[58] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7237–7256. https://doi.org/10.18653/v1/2020.acl-main.647

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[60] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who Are the Crowdworkers? Shifting Demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI EA '10)*. Association for Computing Machinery, New York, NY, USA, 2863–2872. https://doi.org/10.1145/1753846.1753873

[61] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 773, 11 pages.

[62] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1668–1678. https://doi.org/10.18653/v1/P19-1163

[63] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 618–626. https://doi.org/10.1109/ICCV.2017.74

[64] Indira Sen, Mattia Samory, Claudia Wagner, and Isabelle Augenstein. 2022. Counterfactually Augmented Data and Unintended Bias: The Case of Sexism and Hate Speech Detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 4716–4726. https://doi.org/10.18653/v1/2022.naacl-main.347

[65] Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. 2021. Perturbing Inputs for Fragile Interpretations in Deep Natural Language Processing. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 420–434. https://doi.org/10.18653/v1/2021.blackboxnlp-1.33

[66] Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating Gender Bias in Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1679–1684. https://doi.org/10.18653/v1/P19-1164

[67] 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. *arXiv preprint arXiv:1906.08976* (2019).

[68] Yuling Sun, Xiaojuan Ma, Kai Ye, and Liang He. 2022. Investigating Crowdworkers' Identify, Perception and Practices in Micro-Task Crowdsourcing. *Proc. ACM Hum.-Comput. Interact.* 6, GROUP, Article 35 (jan 2022), 20 pages. https://doi.org/10.1145/3492854

[69] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks *(ICML'17)*. JMLR.org, 3319–3328.

[70] European Union. 2018. General Data Protection Regulation. https://gdpr-info.eu/ Accessed: June 2023.

[71] European Union. 2023. The AI Act. https://artificialintelligenceact.eu/ Accessed: June 2023.

[72] Ilse van der Linden, Hinda Haned, and Evangelos Kanoulas. 2019. Global Aggregations of Local Explanations for Black Box models. arXiv:1907.03039 [cs.IR]

[73] Francesco Ventura, Salvatore Greco, Daniele Apiletti, and Tania Cerquitelli. 2022. Trusting deep learning natural-language models via local and global explanations. *Knowledge and Information Systems* (June 2022). https://doi.org/10.1007/s10115-022-01690-9

[74] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2021. Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. *Computer Law & Security Review* 41 (2021), 105567. https://doi.org/10.1016/j.clsr.2021.105567

[75] Madeleine Waller, Odinaldo Rodrigues, and Oana Cocarascu. 2023. *Recommendations for Bias Mitigation Methods: Applicability and Legality*. CEUR Workshop Proceedings.

[76] Yan Xia, Haiyi Zhu, Tun Lu, Peng Zhang, and Ning Gu. 2020. Exploring Antecedents and Consequences of Toxicity in Online Discussions: A Case Study on Reddit. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 108 (oct 2020), 23 pages. https://doi.org/10.1145/3415179

[77] Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics Should Not Be the Reason of Toxicity: Mitigating Discrimination in Text Classifications with Instance Weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4134–4145.

APPENDIX

The Appendix sections are organized as follows. Appendix A shows the LLM prompts used by the *Identifier* to annotate words related to protected attributes. Appendix B shows the list of trap words used in the MTurk study. Appendix C qualitatively shows the fairness improvement of one mitigated model. Appendix D discusses the framework's running time.

A. LLM prompts

The LLM prompts used by the *Identifier* component to annotate words related to protected attributes (as described in §3-B2). A first prompt (Figure 6) provides the protected categories and their definitions. A second prompt (Figure 7) suggests some links that provide more information about the protected categories. Then, for each word, the LLM is asked to: (1) classify the word into one of the protected categories or none of them; (2) provide a reliability score in the range $[0, 100]$; and (3) provide an explanation. Figure 9 shows the response provided by GPT-3.5-Turbo for the annotation of the word *'homosexual'*, classified with the category *sexual orientation* and a score of 100/100.

The LLM-based *Identifier* is also able to annotate proxy words that, although not directly and strictly related to a protected attribute, can be used by the model to infer the categories. An example is shown in Figure 10, where the word *'headscarf'* is annotated as related to *religion and belief*.

> **USER:** Consider these 9 protected categories defined by the Equality Act law to avoid discrimination of automatic decision-making algorithms:
> **"Age"**: A person belonging to a particular age or range of ages (for example, teenagers).
> **"Disability"**: A person has a disability if she or he has a physical or mental impairment which has a substantial and long-term adverse effect on that person's ability to carry out normal day-to-day activities.
> **"Gender reassignment"**: The process of transitioning from one sex to another.
> **"Marriage and civil partnership"**: Marriage is a union between a man and a woman or between a same-sex couple. Same-sex couples can also have their relationships legally recognised as 'civil partnerships'. Civil partners must not be treated less favourably than married couples.
> **"Pregnancy and maternity"**: Pregnancy is the condition of being pregnant or expecting a baby. Maternity refers to the period after the birth, and is linked to maternity leave in the employment context. In the non-work context, protection against maternity discrimination is for 26 weeks after giving birth, and this includes treating a woman unfavourably because she is breastfeeding.
> **"Race"**: Refers to the protected characteristic of race. It refers to a group of people defined by their race, colour, and nationality (including citizenship) ethnic or national origins.
> **"Religion and belief"**: Religion refers to any religion, including a lack of religion. Belief refers to any religious or philosophical belief and includes a lack of belief. Generally, a belief should affect your life choices or the way you live for it to be included in the definition.
> **"Sex"**: A man or a woman.
> **"Sexual orientation"**: Whether a person's sexual attraction is towards their own sex, the opposite sex, or both sexes.

Fig. 6. **Prompt 1.** It provides the definition of the nine protected categories to the LLM.

> **USER:** You can learn more about the discrimination along each protected attribute on the following URLs:
> **"Age"**: https://www.equalityhumanrights.com/en/advice-and-guidance/age-discrimination
> **"Disability"**:https://www.equalityhumanrights.com/en/disability-advice-and-guidance
> **"Gender reassignment"**: https://www.equalityhumanrights.com/en/advice-and-guidance/gender-reassignment-discrimination
> **"Marriage and civil partnership"**: https://www.equalityhumanrights.com/en/advice-and-guidance/marriage-and-civil-partnership-discrimination
> **"Pregnancy and maternity"**: https://www.equalityhumanrights.com/en/node/5916
> **"Race"**: https://www.equalityhumanrights.com/en/advice-and-guidance/race-discrimination
> **"Religion and belief"**: https://www.equalityhumanrights.com/en/religion-or-belief-work
> **"Sex"**: https://www.equalityhumanrights.com/en/advice-and-guidance/sex-discrimination
> **"Sexual orientation"**: https://www.equalityhumanrights.com/en/advice-and-guidance/sexual-orientation-discrimination

Fig. 7. **Prompt 2.** It suggests links which provide more information on the protected categories to the LLM.

> **USER:** Given the previously defined protected categories *"Age"*, *"Disability"*, *"Gender reassignment"*, *"Marriage and civil partnership"*, *"Pregnancy and maternity"*, *"Race"*, *"Religion and belief"*, *"Sex"*, and *"Sexual orientation"*. How would you classify the word "{WORD}" and which [0,100] reliability score (only one) would you give to your assessment? You must assign one category. If a word does not fit any categories, you must assign the category *"None"* with the reliability score and the relative explanation. Provide the answer in the format: "Protected Category|Reliability Score from 0 to 100 for the protected category|Explanation of why the word belongs to the protected category". In case a word does not fall into any category, provide the answer in the format: "None|Reliability Score from 0 to 100 for the None category|Explanation of why the word does not fall under any of the defined protected categories. Each answer must have exactly two | symbols in only one line; otherwise, I cannot process your response.

Fig. 8. **Prompt 3**. For each of the most important words, the LLM is asked to (1) classify the word into one of the protected categories or none of them; (2) provide a reliability score in the range $[0, 100]$; and (3) provide an explanation for the classification. A new request is sent to the LLM for each word by replacing the placeholder {WORD} in the text.

> **GPT-3.5-Turbo:** Sexual orientation | 100 | Homosexual refers to a person's sexual orientation, specifically indicating attraction to people of the same sex. It falls under the protected category of sexual orientation.

Fig. 9. GPT-3.5-Turbo annotation of the word {HOMOSEXUAL}, categorized as *"sexual orientation"*.

> **GPT-3.5-Turbo:** Religion and belief | 90 | The word 'headscarf' is commonly associated with religious beliefs, particularly in Islam, where it is worn by women as a symbol of modesty and religious observance.

Fig. 10. GPT-3.5-Turbo annotation of the word {HEADSCARF}, categorized as *"Religion and belief"*.

B. Trap words

Table 5 shows the list of trap words used in the MTurk for the *Identifier* in the *machine-in-the-loop* setup (§4.3). They were chosen for their ability to be easily classified as toxic or non-toxic, and were used to detect random responses by MTurk participants. For the non-toxic (toxic) trap words, we expected MTurk participants to select a score of 1 or 2 (4 or 5) on the Likert scale. Participants were considered unreliable if they did not meet those expectations, and their assessments were discarded from our results.

Table 5. List of the trap words used in the MTurk study. They were chosen for their ability to be easily classified as toxic or non-toxic to identify random or unreliable responses. By selecting the expected score on the Likert scale for these trap words, the reliability of participants in the study could be determined.

Expected Label	Expected Score	Trap Words
Non-Toxic	1, 2	*beautiful, good, trustful, love, great, curiosity, generous, friendly, sweet, happy, helpful, loyal*
Toxic	4, 5	*asshole, dickhead, motherfucker*

C. Fairness improvement of a mitigated model: a qualitative analysis

Figure 11 and Figure 12 show the fairness improvement of a mitigated classifier for toxicity predictions in the in-distribution experiment (discussed in §4.4). They show the prediction on the same texts discussed before (see Figure 1 and Figure 2 in §1) made by one mitigated model (M_2^*). The first three sentences (misclassified by the original model) are not predicted as toxic anymore, as the model is not extensively using the words *'black'*, *'homosexual'*, and *'gay'* for toxicity predictions anymore. The fourth sentence is still correctly predicted as toxic. However, the prediction is influenced by words such as *'hate'*, *'fucking'*, and *'shitty'* and not by *'black'* anymore. These results show that the removal-based mitigation strategies (MS1, MS2) are highly effective in reducing the usage of protected attributes in classification in just one mitigation round.

Text	*P(T)*
"I like this city! There are many black people!"	0.00
"The homosexual marriage bill will be debated soon! I am in favor!"	0.00
"This city is incredibly modern! If you are gay, you are not judged."	0.16
"I hate this fucking shitty city! There are many black people!"	0.98

Fig. 11. The toxicity probability values P(T) for four sentences produced by the *mitigated* classifier (M_1^*). The original model wrongly classified the first three sentences. Now, they are correctly classified.

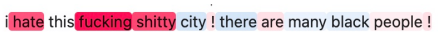

Fig. 12. Words impacting the toxicity classification of the fourth sentence in Table 11. The more intense a word's red (blue) color, the more important the word contributes to toxic (non-toxic) classification.

D. Framework's running time analysis

The execution time to produce the mitigated training dataset depends on many factors. Given a fixed model complexity, the execution times increase linearly with: (1) the unlabeled corpus size, (2) the number of most important words annotated, and (3) the training dataset size. Increasing the model's complexity results in a slight increase in the execution time of all components.

We report the execution time for the mitigation of the BERT model for the *nurse* occupation classification (discussed in §5.4) with Integrated Gradients as the *Explainer* and ChatGPT-turbo-3.5

as the *Identifier*, using a single Nvidia RTX A6000 GPU. We used the test set as the unlabeled corpus, containing approximately 98.3K sentences. Firstly, the *Explainer* uses the original classifier on each input text of the unlabeled corpus. With a batch size of 512, it takes 846 seconds (0.01 seconds per text). Then, the *Explainer* generates the explanations within each sentence for all the texts predicted with the *nurse* occupation, in this case 4,071, and aggregates the scores to compile the overall list of the most important words. This process is completed in 725 seconds (0.18 seconds per text). Next, the *Identifier* annotates the most important 400 words, running in 534 seconds (1.3 seconds per word). Finally, producing the mitigated training dataset with the word removal mitigation strategy (MS2) of the *Moderator* on the 255.7k examples in the training set requires 29 seconds. The total execution time is 2,134 seconds (35 minutes).

Our framework produces a mitigated training corpus, requiring an additional training phase to generate the mitigated model. The (re-)training time depends on the model's complexity and the original training data size. Some mitigation techniques (MS2, MS3, MS5) maintain the training dataset's dimensionality, resulting in equivalent training times for the mitigated and original models (1.5 hours in the previous example). In contrast, techniques like MS1 decrease or MS4 increase the dataset size, leading to corresponding changes in training time.