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

ANONYMOUS AUTHOR(S)

Upcoming privacy laws regulating the use of AI will soon demand that *"learning shall not be done on protected attributes"*. To date, the most accurate Natural Language Processing (NLP) classifiers are based on deep-learning models that behave like black-boxes. We are unaware whether the words used by the classifiers to make accurate predictions are protected attributes. Thus, we cannot prevent the models from relying on them, making it challenging to detect bias and discrimination. Traditional bias mitigation techniques address the problem of performance imbalance between subgroups on a subset of protected attributes such as gender or race. However, they do not assess to what extent NLP classifiers rely on protected attributes for predictions and mitigate this issue. To tackle this challenge, we propose a novel *mitigation framework* that takes an unlabelled corpus of documents, an existing NLP classifier and its training dataset as input to produce a mitigated training corpus that significantly reduces the learning that takes place on protected attributes without sacrificing classification accuracy. It does so by means of three components: an *Explainer* first detects the most important words used by the classifier to make predictions; an *Identifier* then detects which of these words are protected attributes; finally, a *Moderator* re-trains the classifier to minimize the learning on protected words. To evaluate the effectiveness of the framework, as well as its general applicability, we apply it to three classification tasks: i) identifying toxic language, ii) analyzing sentiments, and iii) classifying occupations. We find that existing NLP classifiers rely heavily on protected attributes (as much as 23% of the most important predictive words) in their decision-making. Our fully automated mitigation framework can produce a mitigated classifier that relies on up to 79% fewer protected attributes than the original one while also increasing the classification F1 score by 1%.

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:
Anonymous Author(s). 2024. NLPGuard: A Framework for Mitigating the Use of Protected Attributes by NLP Classifiers. In . ACM, New York, NY, USA, 36 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent years, there has been exponential growth in the adoption of NLP models based on deep learning. The emergence of new transformer-based models, such as BERT [18], T5 [63], and GPT [8], has made it possible to achieve unthinkable levels of performance on several natural language tasks. However, despite being increasingly accurate, these models remain black-boxes [32]. For an NLP classification task, given an input text, they predict a class label without providing any information on the complex internal decision-making mechanism. The opaque nature of classifiers makes it challenging to identify and mitigate bias or unfair behavior in such models.

Upcoming privacy laws regulating the use of AI will soon demand that learning shall not be done on protected attributes [10, 22, 29] such as race, gender, or sexual orientation, as already identified

CSCW'24, November 9-13, 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. The toxicity probability values P(T) for four sentences produced by a toxicity classifier. All sentences are predicted as toxic. The first three sentences are wrongly classified, while the last is correctly classified.

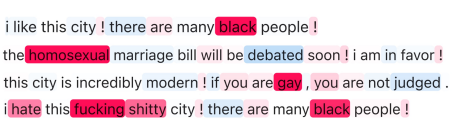

Fig. 2. Words impacting the toxicity classification of the four sentences in Table 1. The more intense the red (blue) color of a word, the more important the word contributes to toxic (non-toxic) classification.

by the *General Data Protection Regulation* (GDPR)[1], the *UK Government*[2], and the anti-discrimination legislation in the United States[3]. However, as we will further report in our analysis, the simple presence of protected attributes in a sentence is used for predicting a class label.

Consider, for example, the task of determining whether a sentence contains toxic language or not. 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 words used by the classifier to make these predictions. As shown, the presence of words such as *'black'*, *'gay'*, or *'homosexual'* are used to distinguish between toxic or non-toxic texts. Yet, the words *'black'*, *'gay'*, and *'homosexual'* are protected attributes and should not be used in such classifications at all. To comply with the new standards demanded by regulators, it is crucial to identify and mitigate this issue [22].

Previously, bias mitigation studies in NLP mainly focused on two challenges: i) mitigating the NLP classifier against performance imbalance on subgroups and selection bias related to gender or race [6, 20, 25, 42, 70, 73]; or ii) tackling the unfairness exhibited in fundamental language model representations (i.e., word embedding) [7, 9, 28, 49, 81]. Instead, this paper tackles the problem from a different perspective and tries to answer the following three research questions (RQs):

- **(RQ1)** *To what extent do existing deep learning NLP models rely on words related to protected attributes to perform classification tasks?*
- **(RQ2)** *How can bias of black-box NLP classifiers be mitigated so that most of the protected attributes are eliminated in the classifiers' decision-making process without sacrificing accuracy?*
- **(RQ3)** *To what extent can such mitigation be generally applied to different kinds of data and tasks?*

In answering these three questions, we make four main contributions:

- We propose a framework, namely NLPGuard, that generates a mitigated training dataset that can then be used to *re-train* an NLP model so to minimize its reliance on protected attributes. The framework consists of three components: 1) an *Explainer* component that determines the most important words used by the NLP classifier to make predictions; 2) an *Identifier* component that determines whether these words are related to protected attributes or not; and 3) a *Moderator* component that modifies the training corpus to re-train the NLP classifier so to minimize learning on such protected attributes.
- We perform an extensive evaluation to assess the sensibility of each component and the effectiveness of our mitigation framework in reducing the reliance on protected attributes

[1] https://ico.org.uk/for-organisations/guide-to-data-protection/key-dp-themes/guidance-on-ai-and-data-protection/what-do-we-need-to-do-to-ensure-lawfulness-fairness-and-transparency-in-ai-systems/
[2] https://www.gov.uk/discrimination-your-rights
[3] https://oag.dc.gov/release/ag-racine-introduces-legislation-stop

by applying it to the case study of classifying toxicity on Wikipedia comments using BERT (§4). For the *Explainer*, we tested two state-of-the-art explainability techniques. For the *Identifier*, we consider both human-in-the-loop and machine-in-the-loop approaches to identify protected attributes. For the *Moderator*, we compare five mitigation strategies (some based on sentence/word removal and some based on replacement). We find that Integrated Gradients is the most effective technique for the *Explainer* component as it demonstrated higher precision in identifying the most crucial words and executes significantly faster. We also find that the machine-in-the-loop Large-Language-Model (LLM) annotation is significantly more accurate than the human-in-the-loop crowd-sourcing approach, suggesting the *Identifier* component (and thus the whole mitigation framework) can be fully automated. Based on this, we found that toxicity predictions of the BERT classifier heavily rely on protected attributes (as much as 23% of its most important predictive words were protected attributes). Finally, the most successful mitigation strategies are the ones that re-train the classifier by means of sentences or words removal strategy: they were able to reduce the usage of protected attributes for prediction by around 60% and also to slightly increase the weighted F1 score by 0.8%, demonstrating the high effectiveness of our framework in reducing protected attributes without sacrificing predictive performance.

- We conducted further experiments to assess the general applicability of our mitigation framework (§5). Firstly, we assessed its effectiveness in mitigating models applied to different data from the ones used for training (out-of-distribution). Specifically, we performed the mitigation of the same toxicity classifier applied to company reviews. Our findings reveal that our framework is also effective in mitigating classifiers used on out-of-distribution data, as it was able to reduce the reliance on protected attributes by 79% while also increasing the weighted F1 score by 1%. Secondly, we evaluated its effectiveness in mitigating a sentiment analysis model. We find that it was able to reduce reliance on protected attributes by 50% without experiencing accuracy loss. Thirdly, we evaluated its effectiveness in mitigating the occupation classification from online biographies [17], a task in which the final prediction maps to a decision in the real world that may have concrete outcomes for an individual. We compared its ability to reduce the reliance on protected attributes with two model-based mitigation techniques: *Iterative Null-space Projection (INLP)* [64], and *Entropy-based Attention Regulation (EAR)* [4]. We found that our framework is the most effective in mitigating the use of such protected attributes by an average 44% decrease without sacrificing prediction accuracy in the occupation classification task.
- We release a practical bias mitigation tool[4] that can be easily applied to any machine learning NLP classification model. We discuss how it can be integrated into existing NLP pipelines, its implications, and current limitations or possible areas of concern (§6).

The above contributions represent significant advances to the literature since our framework is able to extensively mitigate the use of protected attributes without sacrificing predictive performance. Furthermore, our framework covers a broader range of protected attributes (i.e., all attributes defined in GDPR), does not rely on manually pre-defined templates or further dataset annotations, and allows full automation of the entire process.

2 RELATED WORK

We first provide background about current AI regulations and laws in §2.1. Then, we discuss current bias mitigation techniques in §2.2.

[4]The code repository of our framework is available at `anonymous-url-for-blind-review`

2.1 AI Regulations and Laws

In recent years, the growth of artificial intelligence (AI) systems has raised concerns regarding the protection of individual privacy and the potential for discriminatory behaviors. Several privacy laws and regulations are being introduced to regulate the use of AI.

In the European Union (EU), in May 2018, the General Data Protection Regulation (GDPR) [76] was introduced, which regulates data protection and privacy. It demands that organizations ensure that personal data is processed lawfully, fairly, and transparently. The GDPR prohibits processing sensitive personal data, including attributes such as race, ethnicity, religion, and political opinions, unless there is a legitimate basis for using them. Recently, the EU also proposed the EU AI Act [46, 77], which defines rules that establish obligations depending on the level of risk from AI systems. For instance, it defines requirements for several aspects, such as transparency, documentation, and human oversight [57], based on the level of risk of the AI system.

In the United Kingdom (UK), 2010, the UK Equality Act 2010 [40] was proposed. It provides a legal framework for addressing disadvantages and discrimination. It defines nine protected characteristics: age, disability, gender reassignment, marriage and civil partnership, pregnancy and maternity, race, religion or belief, sex, and sexual orientation. It establishes that it is unlawful to discriminate against someone based on any of these protected attributes. The governing body responsible for enforcing the Equality Act 2010 in the UK is the Equality and Human Rights Commission (EHRC) [21].

In the United States (US), the Anti-discrimination Act [12] safeguards individuals from unfair treatment based on protected attributes such as race, gender, religion, and disability, promoting equal opportunities for all. By mitigating discrimination based on protected attributes, the legislation fosters a more inclusive society where individuals can thrive and contribute irrespective of their inherent characteristics. Recently, in the US in late 2022, it has also been passed a blueprint of the AI Bill of Rights [37]. It declares that algorithmics that discriminate or perform unjustified different treatment based on protected attributes may violate legal protections.

Many other regulations associated with AI exist in several other states. These regulations and laws are constantly evolving and are expected to continue in the coming years [39, 54]. However, all these entities seek to reduce the risk of discriminatory outputs based on protected characteristics, raising the issue of reducing the use of protected attributes by AI-based models [10, 22, 29]. Therefore, it is important to assess that NLP classifiers do not extensively rely on protected attributes for predictions and reduce this issue when present.

2.2 Bias mitigation for NLP

In recent years, there has been a significant effort to address biases in automated NLP decision-making processes [6, 20, 50, 70]. These processes have the potential to amplify and perpetuate existing biases, leading to unfair and unequal outcomes.

When quantifying bias, existing works generally highlight disparities between demographic groups. They often use metrics that measure the difference in performance or selection bias on protected attributes such as race, gender, religion, and sexual orientation [25, 33, 42, 73]. For example, instances of individuals (e.g., male and female for gender) are often evaluated using pre-defined templates [58], lists of identity terms [20], or an additional annotation to each sample of the dataset [64]. These terms or additional annotations serve as a proxy for the actual protected attributes (gender) of individuals and allow for a standardized way of comparing and evaluating instances within a demographic group.

Bias in NLP can manifest in many ways, such as biased dialogue generation [19], text classification [20], and machine translation [72]. The biases can arise in the training data used to build

NLP models [25, 73], which can contain biases and stereotypes based on language use and cultural norms. Another source of biases in NLP is pre-trained models, such as word embeddings, which are used to represent the meaning of words in a numerical format [7, 9, 28, 64]. These models are often pre-trained on large text corpora, which can contain biases and stereotypes. As a result, the biases present in the large text corpora can be reflected in the pre-trained models and their outputs, leading to biased and unfair outcomes.

Mitigating biases in NLP models is an active area of research, and the majority of existing works address the problem through data augmentation and modified model training [4, 6, 20, 58, 64, 82]. These techniques aim to address the biases that are present in the training data and the models themselves by adjusting the data and the models to be more fair and unbiased. Most existing works on mitigating biases in NLP models focus on one protected category at a time or specific categories. For example, Badjatiya et al. [6] proposed effective techniques for the identification of protected attributes, such as gender, by creating a manual list of words, measuring skewed occurrence of words across classes, or measuring the skewed predicted class probability distribution of words. Park et al. [58] instead introduced gender swapping as a technique to equalize the number of male and female entities in the training data. Similarly, Dixon et al. [20] proposed effective strategies for mitigating biases in NLP models by augmenting the original dataset. These strategies involve generating new sentences using templates or replacing protected attributes with generic tags, such as part-of-speech or named-entity tags. Another work by Zhang et al. [82] proposed mitigating biases in the training data by assuming a non-discrimination distribution and then reconstructing the distribution using instance weighting. This approach adjusts the distribution of instances in the training data to ensure that it is fair and unbiased. Finally, Ravfogel et al. [64] proposes a method for removing information from neural representations concerning gender or race that can be used for debiasing word embedding for NLP classification. These works demonstrate the potential of data augmentation and model training techniques for mitigating biases in NLP models.

In summary, previous works try to mitigate unintended bias and performance imbalance between subgroups by i) removing implicit bias from word embeddings, ii) performing data augmentation techniques on the training set (i.e., data-based), or iii) intervening directly in the model architecture or objective function (i.e., model-based). Typically, these works employ techniques like data augmentation or modified model training. However, a gap remains in evaluating the extent to which NLP classifiers depend on protected attributes for their predictions and in addressing this issue. In this paper, we aim to fill that gap, and we use the definition of *Fairness through unawareness*: "an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process" [31, 44, 50] adapted to NLP classification.

Apart from having a different objective than ours, previous techniques also exhibit two other main limitations. Firstly, they mainly focus on a subset of protected attributes at a time, usually race and gender. Secondly, the identification of protected attributes is done using manually created pre-defined dictionaries, lists of identity terms, or additional sample annotations, which are static and encompass only a subset of all the potential protected attributes. The only technique addressing these limitations is *Entropy-based Attention Regulation* (EAR) [4]. EAR introduces a regularization term to discourage overfitting to training-specific terms, which could include protected attributes introducing bias into the model. However, the terms mitigated by the technique are automatically identified during training, leaving no flexibility for users to select which categories to mitigate.

Unlike previous techniques, our approach handles all protected categories simultaneously. This makes it possible to simultaneously identify and mitigate biases in multiple protected attributes, which can lead to more fair predictions. Moreover, our framework can be fully automated by exploiting machine-in-the-loop identification of protected attributes, allowing for a dynamic update of the dictionary of protected attributes.

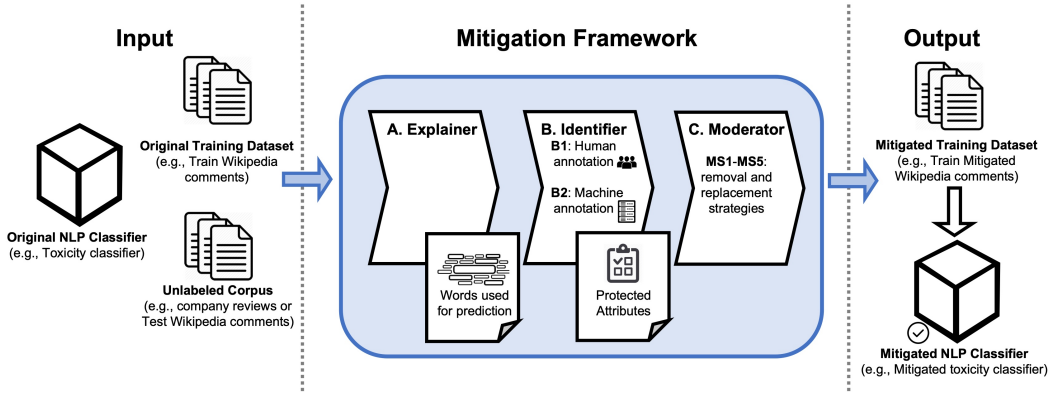

Fig. 3. **Our Mitigation Framework.** It takes the original NLP classifier, the original training dataset, and a new unlabeled corpus as input. The framework then consists of three components: A) an *Explainer* component that identifies the most important words used by the classifiers for its predictions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

3 OUR MITIGATION FRAMEWORK

Our Mitigation Framework (Figure 3), namely NLPGuard, has been designed to be generally applicable to any supervised machine learning-based NLP classification model applied on an unlabelled corpus. More precisely, our framework 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 data (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 aims to identify the most important words used by the model for its predictions by adopting *Explainable Artificial Intelligence* (XAI) techniques. The XAI field has made great strides in making black-box models more transparent, and several techniques exist to explain NLP classifiers [14]. Still, the best one for our purpose should have two qualities: i) quantify the importance of each feature word (feature-based), and ii) 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, 48, 65, 69, 75, 79]. First, the component 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 some of these techniques, which aggregate the words' importance over many sentences to compute the overall importance [78, 79]. 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 establishes which important words are protected attributes. It annotates each important word previously found with a label indicating whether each word falls into one of the following nine protected categories defined by the *Equality and Human Rights Commission (EHRC)*[5]: *age*, *disability*, *gender reassignment*, *marriage and civil partnership*, *pregnancy and maternity*, *race*, *religion or belief*, *sex*, *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 (abbreviated as MTurk) and Prolific [13], have been extensively utilized by the research community to recruit crowdworkers for data labeling purposes. Crowdworkers [66, 74] 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, appropriate mechanisms are adopted to detect random responses from participants and reject them. In §4.3, we will discuss the trap mechanism used in our study.

B2. Machine-in-the-loop Annotation. In a recent study [26], it was discovered that Large Language Models (LLMs), including ChatGPT, outperformed crowdworkers in text-based annotation tasks. It also has been shown that LLMs are effective in solving many NLP tasks [62]. These findings suggest that LLMs can serve as a cost-effective and scalable alternative to human-in-the-loop crowdsourcing tasks. Inspired by this potential, we incorporated the capability of annotating protected attributes utilizing LLMs. Specifically, we implemented an annotation process by interacting with ChatGPT and prompting it with queries to determine whether a word is a protected attribute.

A first prompt (Figure 4) provides the protected categories and their definitions. A second prompt (Figure 5) suggests some links which provide more information about the protected categories. Finally, for each word, the LLM is asked to i) classify the word into one of the protected categories or none of them; ii) provide a reliability score in the range $[0, 100]$; and iii) provide an explanation for the classification. A classification request is sent to the LLM for each word (Figure 6).

Figure 7 shows the response provided by GPT-3.5-Turbo for the annotation of the word *'homosexual'*, classified with the category *sexual orientation* with a score of 100/100.

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. Specifically, it takes as input the original pre-trained classifier, the original training dataset, and the list of the most important words enriched with the protected attribute label. Then, it produces a new *mitigated training dataset* by adjusting the dataset used for training based on the identified protected attributes. The mitigated training dataset is used to train a new mitigated classifier. We designed and tested five mitigation strategies (MS).

[5] https://www.equalityhumanrights.com/en/equality-act/protected-characteristics

> **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. 4. **Prompt 1.** It provides the definition of each protected category (i.e., *age*, *disability*, *gender reassignment*, *marriage and civil partnership*, *pregnancy and maternity*, *race*, *religion or belief*, *sex*, *sexual orientation*) to the LLM.

> **USER:** You can learn more about the discrimination along each protected characteristic 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. 5. **Prompt 2.** It suggests some links which provide more information on the protected categories to the LLM.

- *Sentence-level removal (MS1).* Previous works have shown that subsampling can be an easy but effective technique for data balancing [38, 67]. This mitigation strategy eliminates all sentences containing protected attributes from the training set. For example, if a word W_i is identified as a protected attribute, all sentences in the training set that include that word

> **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. 6. **Prompt 3**. For each of the most important words, the LLM is asked to i) classify the word into one of the protected categories or none of them; ii) provide a reliability score in the range $[0, 100]$; and iii) 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. 7. Example of GPT-3.5-TURBO response for the annotation of the word {HOMOSEXUAL}, categorized as *"sexual orientation"* with score 100/100.

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 believe that these protected attributes are crucial for accurately classifying that class. As a result, this mitigation technique reduces the overall number of training examples.

- *Word-level removal (MS2).* 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 (e.g., W_i) 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 attribute words, and rather on the other non-protected words present in the text.

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

- *Word-level replacement with K random synonyms (MS4).* This strategy expands the training set by generating new sentences using synonyms of protected attributes. Instead of simply

replacing the protected attribute in-place with a similar word as in *MS3*, this strategy 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.

- *Word-level replacement with hypernym (MS5).* This strategy replaces each instance of a protected attribute in the training set with a higher-level word that encompasses the meaning of the protected attribute. This higher-level word, known as a hypernym, provides 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].

Advantages of our framework. Compared to prior studies [42, 70, 73] that mitigate biases of models on protected attributes, our framework has three main advantages:

- Our framework can cover a much broader range of protected attributes, compared to prior work [42, 73], which mostly focused on gender [73], age [24], and race [55]. Many protected attributes, such as disability or religious belief, were rarely covered by prior studies.
- The identification process in our framework captures more attributes than just those that are pre-defined with templates [70, 73, 83]. Our identified attributes could also be more specific to the application domain. By using explainability techniques in our framework, we can pinpoint the specific words that the model is using for the classification rather than looking at all possible words in the corpus. This allows us to focus specifically on the important words that the model is using to make its predictions and reduces the complexity of the problem. With this approach, we can more effectively identify and mitigate the use of protected attributes in the model. Additionally, it ensures that the mitigation efforts are directed toward the words that actually impact the model's predictions.
- Our mitigation framework can be fully automated end-to-end by leveraging LLM (Large Language Models) annotation. This automation eliminates the need for human intervention throughout the process.

4 FRAMEWORK EVALUATION: EFFECTIVENESS AND SENSITIVITY

To answer RQ1 and RQ2 outlined in §1, 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). In §5, we will explore their generalizability, including a comparative analysis with two previous techniques.

4.1 Evaluation task

We choose toxicity prediction as the main evaluation task in line with previous research. Toxic language can be defined as a way of communicating that harms or hurts other people. Deep NLP models could be powerful tools for discovering and identifying toxic texts online. Toxicity classifiers are used in different contexts [43, 80], such as Reddit, Twitter, and 4chan, showing competitive performance. However, those classifiers often suffer from different types of biases, as demonstrated in prior literature [15, 16, 35, 68]. It has been found that any Wikipedia comment containing words associated with insults, offense, or profanity, regardless of the tone, the intent, and the context of

that comment, would be classified as toxic. For instance, [15] showed that some toxicity classifiers are more likely to predict toxic language from minority communities, thus suggesting the use of protected attributes by such models. One of the possible reasons is that data annotation did not include such groups in the labeling of toxic comments [30]. Mitigating the use of protected attributes in toxicity classification is important to ensure fairness, equity, and the responsible deployment of AI systems in addressing online toxicity. It helps prevent the perpetuation of biases and discrimination, and promotes a safer and more inclusive online environment for everyone.

In our experiments, we used the "original model" in the widely used detoxify [34] library[6] as a pre-trained toxicity classifier. This model is a BERT-base and uncased model, which means that the model is based on the architecture proposed in the BERT paper [18], and the text was not cased (i.e., upper/lower case distinction was ignored) during the pre-processing. The model was trained on a dataset of publicly available Wikipedia comments.[7] Specifically, 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%. By only considering the *toxicity* label, the classifier achieved 0.82 macro and 0.93 weighted F1 scores (the dataset is imbalanced). These high scores indicate that the model performs well in distinguishing between toxic and non-toxic comments. This classifier is applied to the original test set comprising 153,164 texts.[8] The classifier predicted the toxicity label (i.e., with toxicity probability ≥ 0.5) for 36,148 texts (23.6% of the test set).

4.2 Component evaluation: Explainer

The aim of the *Explainer* is to identify the most crucial words utilized by the classifier for predictions (as described in §3-A), in this case, the most important words for predicting toxicity. The employed XAI technique can influence the words recognized as significant by the model, 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 initially measure the impact on the model's predictive performance (i.e., F1 score) by removing the most important words identified by each XAI technique. The underlying idea behind this evaluation is that an effective and precise explainer should result in a noticeable decrease in the classifier's predictive performance when the identified words are removed. Secondly, to assess the sensitivity of the *Explainer* to different XAI techniques, we measure the overlap of the most important words identified by different XAI techniques. A substantial overlap indicates that the *Explainer* consistently produces similar outputs when employing different techniques. This might suggest that this choice does not significantly affect the output of this component. Lastly, we measure the computation time for generating explanations to assess the efficiency of the *Explainer* based on the XAI technique. Using an efficient *Explainer* to generate explanations faster is preferable 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) [48] as a representative of the permutation-based and with *Integrated Gradients* (IG) [75] of the gradient-based techniques. Both techniques have been shown to demonstrate competitive performance in prior studies [3]. Specifically, for SHAP, we used the *text permutation explainer*[9] with 3,000 as maximum evaluations

[6] https://github.com/unitaryai/detoxify
[7] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data
[8] The ground truth labels are available only for a subset of the test set (42%).
[9] https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/text.html

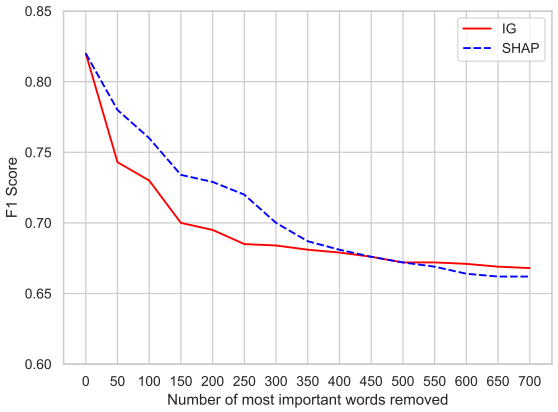

Fig. 8. **Explainer component evaluation.** F1 score decrease by removing the most important words from the test set. The most important words are extracted by the *Explainer* component instantiated with the Integrated Gradients (IG) and SHAP techniques. A greater decrease indicates a higher precision of the technique in identifying the most important words for the predictions.

step parameter. For Integrated Gradients, we exploited the implementation provided by the Ferret [5] library.

4.2.3 Results. Firstly, we produced the explanations within each sentence using both techniques for all toxic texts in the test set. Secondly, we aggregated the individual scores to extract the ordered list of the most toxic words (as described in §3-A).

Figure 8 shows the decrease in the F1 score by removing the most important words identified with SHAP and Integrated Gradients (IG) from the test set. The experiment was repeated for the most important words in the range [50, 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. Integrated Gradients exhibit 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 the classifier for its predictions. We selected the top 400 most toxic words since removing additional words causes a lower decrease in predictive performance, corresponding approximately to the top 10% most toxic words.

Subsequently, we measured the overlap between the 400 most toxic words identified with Integrated Gradients and SHAP. Our findings revealed that 307 out of the 400 words were identical (77%), indicating a substantial agreement between the two techniques regarding the identification of the most toxic words for the model. However, 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 Integrated Gradients 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 Integrated Gradients and 30 for SHAP. The experiments were performed on a single Nvidia RTX A6000 GPU.

In summary, the Integrated Gradients algorithm proves to be the most effective technique for the *Explainer* component. It exhibits higher precision in identifying the most crucial words and executes significantly faster based on the empirical evidence of those experiments. As a result, we

adopt Integrated Gradients as the XAI technique in the *Explainer* for all subsequent experiments. Nevertheless, the framework allows users to select alternative techniques for the *Explainer*, like SHAP.

4.3 Component evaluation: Identifier

The *Identifier* component aims to identify 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.[10] These two annotators are considered 'experts' because they possess a greater depth of experience and knowledge of the definitions of protected categories by AI regulators than participants engaged in the human study. We also add for comparison a pre-defined dictionary of 51 protected attributes from previous works on bias mitigation in toxicity classification [6, 20]. This evaluation allows us to assess the accuracy and performance of the protected attributes identification component using different approaches.

4.3.1 Evaluation metrics. We use the Cohen's kappa inter-annotator agreement [11] on the protected attributes identified by the different approaches as the evaluation metric. It ranges between $[0, 1]$; the higher the score is, the higher the agreement. The level of agreement between the component and expert annotators provides quantitative confidence in the accuracy and reliability of the component's output.

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* component with human-in-the-loop and machine-in-the-loop as follows.

Human-in-the-loop setup. We annotated the most toxic words with an MTurk study following the procedure discussed in §3-B1. We also added a question acting only as a trap to detect random responses and also useful to collect further annotations for other research purposes. Specifically, in addition to the question aimed at labeling protected attributes, we provided 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*.

We added 15 trap words (Table 1) that can be easily classified as *toxic* or *non-toxic*. Each page contained questions concerning 8 original words plus 2 trap words. For the *non-toxic* trap words, we expected MTurk participants would select a score of 1 or 2 on the Likert scale. In contrast, for the *toxic* trap words, we expected a score of 4 or 5. 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%).

[10] The expert annotators are people within our team with a background in human-computer interaction and more than three years of experience in trustworthy and responsible AI. They are located in two different Western countries, with different ethnicities and ages (anonymized for blind review). The experts carefully read the UK Equality Act 2010 and unanimously agreed on the type of annotation to be performed. The annotation was done independently.

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*

Table 1. List of the trap words used in the MTurk study. These words were carefully selected to identify random or unreliable responses. They were chosen for their ability to be easily classified as toxic or non-toxic. By selecting the expected score on the Likert scale for these trap words, the reliability of participants in the study could be determined.

Fig. 9. **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. For instance, A1-GPT is the agreement between the expert annotator 1 and ChatGPT. The agreement score is in the range [0, 1], where a higher score indicates a higher level of agreement.

On average, we obtained five annotations for each word. We defined a word as a protected attribute based on majority voting. Specifically, for each word, we summed the votes for all protected categories (i.e., *age*, *disability*, *gender reassignment*, *marriage and civil partnership*, *pregnancy and maternity*, *race*, *religion and belief*, *sex*, and *sexual orientation*). Then, if this sum was *strictly greater* than the number of *None of the above* received, the word was labeled as a protected attribute.

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. Then, for each candidate word, we prompted GPT-3.5-Turbo asking for a possible protected category label, 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 and 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 (answering *RQ1* in §1).

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.

All the expert annotators (A1 and A2), ChatGPT, and MTurk agree on the annotation of certain words, for example, *'gay'*, *'black'*, *'males'*, *'jew'*, *'homosexual'* and *'lesbian'*. However, some words are labeled as protected attributes by A1, A2, and ChatGPT but not MTurk. Some examples are *'arabian'*, *'nigerian'*, and *'madam'*. Instead, some words labeled as protected attributes by A1, A2, and MTurk, but not ChatGPT, are *'mom'* and *'mommy'*.

Figure 9 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 [45]. The ChatGPT annotations demonstrate substantial agreement (0.67) and moderate agreement (0.56) 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 the ability of the *Identifier* component to correctly classify protected attributes with both human and machine annotations. The machine-in-the-loop LLM-based method outperforms the human-in-the-loop crowdsourcing one in identifying protected attributes. Additionally, the LLM-based method enables full automation of the framework without human intervention. Finally, it is worth noting that pre-defined dictionaries often lack specific categories or require regular updates to stay current, making them less reliable for identifying protected attributes. For the next experiments, we adopt the LLM-based machine-in-the-loop approach for the *Identifier* component to identify protected attributes.

4.4 Component evaluation: Moderator

The *Moderator* aims to create a mitigated training corpus to train a new classifier that reduces its reliance on protected attributes without compromising the predictive performance. To evaluate the effectiveness of each mitigation strategy, we trained a distinct mitigated model for each of them. By evaluating the mitigated models, we can ascertain the efficacy of each strategy in achieving the desired outcome.

4.4.1 Evaluation metrics. In the evaluation of the *Moderator* component, 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*, "an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process" [50], and is quantified by measuring the number of protected attributes each mitigated model relies on in making predictions based on the *Identifier* component. 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, which gives us insight into the model's accuracy in identifying instances of toxicity. 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 mitigation strategies (MS1 and 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 and 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 word embedding trained on Wikipedia and Gigaword [61]. This method was suggested in [6]. For the mitigation strategy that replaces protected attributes with hypernyms (MS5), we utilized the WordNet lexical database [51], which is provided in the NLTK library [6]. Specifically, we replaced each protected attribute with its first-level hypernym extracted from its synset of synonyms.

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. Table 2 shows the differences in the number of training examples after each strategy in the third column. The original training dataset contains 159,571 examples. The strategy that eliminated sentences with protected attributes (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[11] for 3 epochs, with a batch size of 16, and Adam [41] 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* component (instantiated with the Integrated Gradients algorithm) to extract the most important predictive words used by each *mitigated* model for the toxicity predictions in those texts. Finally, we exploited the *Identifier*, instantiated with ChatGPT, 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 2 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 these toxic words already present among the protected attributes used by the original model (M_o) is also indicated in square brackets. Our findings reveal that all evaluated mitigation strategies reduced the number of protected attributes the model relied upon. However, the mitigated models trained with removal-based strategies (MS1 and 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.

We can qualitatively see the fairness improvement of the mitigated models in Figure 10. It shows the prediction on the same texts discussed before (see Figures 1 and 2 in §1) made by one of the mitigated models (M_2^*). The first three sentences are not predicted as toxic anymore (they were wrongly predicted as toxic by the original classifier). This is probably because the model is not extensively using the words *'black'*, *'homosexual'*, and *'gay'* for toxicity predictions anymore. The

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

Model ID	Mitigation Strategy	Δ Train Examples	*Predictive Performance* ↑ F1 Macro	F1 Weight	AUC	*Fairness* ↓ % 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^*	*MS4 - 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]

Table 2. **Mitigating toxicity prediction on in-distribution data.** Summary of the results in mitigating the toxicity classifier applied to the in-distribution data (i.e., test set). The original model is highlighted in grey (M_o). For each mitigated model are reported: i) the model identifier, ii) the mitigation strategy applied, iii) the decrease or increase of training examples after the mitigation strategy, iv) the F1 macro and weighted scores for the toxicity class on the original test set, v) the *AUC* score for all toxicity-related labels on the original test set, vi) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in brackets. *Fairness* is defined as the model's reliance on protected attributes. The metrics indicated with ↑ (↓) represent higher (lower) values being better. The best performing for each metric is in bold.

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. 10. 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.

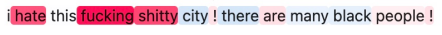

Fig. 11. Words impacting the toxicity classification of the fourth sentence in Table 10. The more intense a word's red (blue) color, the more important the word contributes to toxic (non-toxic) classification.

fourth sentence is still correctly predicted as toxic. However, as shown in Figure 11, the prediction is influenced by words such as *'hate'*, *'fucking'*, and *'shitty'* and not by *'black'* anymore.

These results provide compelling evidence that the removal-based mitigation strategies (MS1 and MS2) are highly effective in reducing the usage of protected attributes in classification. Remarkably, this achievement was obtained after just one round of mitigation.

Predictive performance. In Table 2, columns 4 and 5 report the macro and weighted F1 scores on the toxicity label for both the original and mitigated models. Additionally, column 6 also presents the mean *AUC* scores across all the toxicity labels (i.e., *toxicity*, *severe toxicity*, *obscene*, *threat*, *insult*, and *identitiy attack*). The results show that all the models present similar F1 scores compared to the original model, except for the model trained with MS4 which exhibits a greater decrease. Interestingly, the mitigated models trained on the removal-based mitigation strategies (MS1 and MS2) achieve better F1 scores. Specifically, the word removal strategy (MS2) increases the macro and weighted F1 scores by 1.3% and 1.2%, respectively. Indeed, we observed that the removal-based mitigation strategies reduced the number of false positives in the toxicity predictions (i.e., non-toxic texts wrongly predicted as toxic). Finally, 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. The largest decrease in *AUC* score

is observed in the M_1^* model, with a decrease of 0.005 (from 0.986 to 0.981). Despite this minor reduction in predictive performance, all the mitigated models still exhibit competitive performance in the classification task, making the tradeoff for a fairer classifier well worth it.

Based on these findings, we can conclude that our framework effectively reduces the model's reliance on protected attributes without compromising the predictive performance, answering RQ2 in §1. Indeed, all the 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 mitigated models trained on removal-based strategies also increased their predictive performance after the mitigation.

5 FRAMEWORK EVALUATION: GENERALIZABILITY

To answer RQ3 in §1, we evaluate the generalizability of our framework to toxicity prediction on out-of-distribution data (§5.2) and different tasks, i.e., sentiment analysis (§5.3) and occupation classification (§5.4). In the occupation classification, where an additional protected attribute label (gender) is available for each individual, we also conduct a comparison with two baselines.

5.1 Framework and evaluation settings

For this evaluation, we instantiated the *Explainer* with the Integrated Gradients algorithm, 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 follow a similar approach to evaluate the mitigated models by measuring their *fairness* and *predictive performance*. *Fairness* is defined as *fairness through unawareness*, and is evaluated by quantifying the number of protected attributes that each mitigated model relies on, indicating the extent to which the model's predictions are influenced by these attributes. For *predictive performance*, we employ quantitative evaluation metrics on the test set. In the case of the toxicity classifier, we measure the F1 score specifically for the toxicity label, which allows us to gauge the model's accuracy in detecting instances of toxicity. Additionally, we calculate 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. Regarding the sentiment and the occupation classifiers, we solely measure the F1 score as the relevant evaluation metric, as it provides a comprehensive assessment of the model's accuracy in sentiment classification.

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, after training, classifiers are normally applied to datasets from other domains where the word distribution is significantly different from that of the training data. Toxicity classification on company reviews simulates a possible real application scenario to out-of-distribution data. We aim to determine whether our mitigation framework remains effective.

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. The platform, for example, allows only registered users to write reviews. Also, reviews are published anonymously. On the one hand, this

Source	Text	P(T)
pros	*If you are a guy (black) or lesbian you get hired fast.*	0.82
cons	*They discriminate against gays. I was bullied and harassed daily til I quit and got a lawyer*	0.85
cons	*Poor female manager. Coworkers do not support new staff*	0.65

Table 3. Examples of toxicity predictions performed by the original classifier on some company reviews. *Source* shows if the text comes from *pros* or *cons* of the review. *Text* is the sentence used as input to the toxicity prediction. $P(T)$ is the toxicity probability predicted by the *original* model (M_o).

Fig. 12. Toxic words within each individual sentence. The explanations were produced with the SHAP framework [48]. The more intense the red (blue) color of a word, the stronger its association with the toxic (non-toxic) class. Protected attributes such as *'black'*, *'lesbian'*, *'gay'*, and *'female'* are among the most important words in toxicity predictions.

promotes user privacy. On the other hand, however, 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. These reviews were written from 2008 to 2020 and belonged to 11 industries classified according to the Global Industry Classification Standard (GICS) ('Industrials', 'Consumer Staples', 'Health Care', 'Financials', 'Energy', 'Materials', 'Communication Services', 'Utilities', 'Real Estate', 'Consumer Discretionary', 'Information Technology'). Each review contains a *pros* part (positive comments in the review) and a *cons* part (negative comments). The mean number of tokens[12] (words) in a review is 23 (pros) and 37 (cons). 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 contain inappropriate content, we found 1.6% of 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*. However, interestingly, some people tend to be so angry and frustrated by the work experience that they let off steam even in the *pros* fields.

Table 3 shows examples of toxicity predictions for some *pros* and *cons*. We can observe that those examples contain words belonging to protected attributes referring to race, gender, or sexual

[12]We leveraged the tokenizer from the NLTK library https://www.nltk.org/api/nltk.tokenize.html

orientation. These words must not be used by the model for its predictions. However, its black-box nature hides such a problematic use of protected attributes.

Figure 12 shows the most toxic words within each sentence in Table 3 produced by the *Explainer*. The more intense a word's red (blue) color, the stronger its association with the toxic (non-toxic) class. It confirms that the model also uses protected attributes in its predictions when applied to out-of-distribution data.

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 (original model M_o) in Table 4. Based on this evaluation, 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. This finding confirms the answer presented for *RQ1* in §1. The analysis of the original classifier's behavior across diverse datasets provides evidence that protected attributes strongly influence its predictions in the context of toxicity classification.

5.2.4 Training the mitigated models. We applied the removal-based mitigation strategies (MS1 and MS2) to the original Wikipedia comments training dataset based on the protected attributes identified on the toxic company reviews. Table 4 shows the differences in the number of training examples after each strategy in the third column. The original training dataset contained 159,571 examples. The strategy that eliminates sentences with protected attributes (MS1) resulted in a decrease of 6k examples. Instead, the word-removal strategy did not change the number of training examples. 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 4 show the percentage and number of the most toxic words labeled as protected attributes. The number of these toxic words already present among the protected attributes used by the original model (M_o) is also indicated in square brackets. The results confirm that removal-based mitigation strategies reduce the number of protected attributes the model relied upon. The sentence removal (MS1) and the word removal (MS2) reduce the percentage of protected attributes from 19% to 4% and 5% (16 and 19 protected attributes out of 400 important words), respectively. This corresponds to a decrease of 79% and 75%. They also mostly reduce the protected attributes the original classifier relies on from 76 to 8 and 11, respectively.

Predictive performance. Columns 4 and 5 in Table 4 show the macro and weighted F1 scores achieved by the original and mitigated models on the test set. Also in this case, 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

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]

Table 4. **Mitigating toxicity prediction on out-of-distribution data.** Summary of the results in mitigating the toxicity classifier applied to the out-of-distribution data (i.e., company reviews). The original model is highlighted in grey (M_o). For each mitigated model are reported: i) the model identifier, ii) the mitigation strategy applied, iii) the decrease or increase of training examples after the mitigation strategy, iv) the F1 macro and weighted scores for the toxicity class on the original test set, v) the *AUC* score for all toxicity-related labels on the original test set, vi) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in brackets. *Fairness* is defined as the model's reliance on protected attributes. The metrics indicated with ↑ (↓) represent higher (lower) values being better. The best performing for each metric is in bold.

only 0.003. Also in this case, we can conclude that all the mitigated models still exhibit competitive performance in the classification task (even better in distinguishing toxic texts) while increasing the fairness of the classifier.

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 the adaptability and robustness of our framework 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 because we want to test our framework in mitigating the use of protected attributes for tasks where we might expect a lower reliance on them.

5.3.1 Training the original sentiment classifier. We fine-tuned a BERT-base and uncased model for sentiment classification with a dataset of 163K tweets and 37K Reddit comments[13] in English. The dataset expresses people's opinions towards the general elections held in India in 2019. The task consists of a multi-class classification problem with 3 classes: *negative*, *neutral*, and *positive*. We split the dataset with 80% for training (160,000) and 20% for testing (40,000). 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

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

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]

Table 5. **Mitigating sentiment analysis.** Summary of the results in mitigating the sentiment classifier. The original model is highlighted in grey (M_o). For each mitigated model are reported: i) the model identifier, ii) the mitigation strategy applied, iii) the decrease or increase of training examples after the mitigation strategy for the *negative* and *positive* classes, iv) the F1 macro score on the original test set, vi) the percentage and ratio of relied-upon protected attributes, with the number of those present in the original model in brackets, for both classes. *Fairness* is defined as the model's reliance on protected attributes. The metrics indicated with ↑ (↓) represent higher (lower) values being better. The best performing for each metric is in bold.

words. Our findings reveal that out of 200 negative words, 16 (8%) were identified as protected attributes by the *Identifier*. Similarly, out of 200 positive words, 11 (6%) were annotated as protected attributes. These results suggest that the sentiment model exhibits a moderate reliance on its predictions concerning protected attributes.

5.3.3 Training the mitigated models. We applied the two removal-based mitigation strategies (MS1 and MS2). Notice that, in this case, we analyzed two class labels. Therefore, 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). In this case, the MS1 mitigation decreases the training set by 5k *negative* and 8k *positive* training examples, as shown in the third and fourth columns in Table 5. Also in this case, the models were fine-tuned for 3 epochs.

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

5.3.4 Results.

Fairness. The last four columns in Table 5 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. The mitigation strategy 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* class). Interestingly, the number of protected attributes the original classifier relied on is almost completely mitigated, except for 2 protected attributes for the *negative* and 1 for the *positive* classes. The mitigated model produced by the 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. We can conclude that MS1 is the most effective in this case.

Predictive performance. The fifth column in Table 5 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.

Our experimental 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. This suggests that our framework can generally mitigate the usage of protected attributes without sacrificing accuracy also when applied to out-of-distribution data or different NLP classification tasks, answering RQ3 in §1.

5.4 Mitigating occupation classification

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

To achieve these goals, we selected the task of predicting occupations from online biographies. We used a dataset of online biographies annotated by gender and occupation from previous works [17, 60, 64]. 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"). As a further pre-processing step, we removed all the first names because the model can use them to infer the gender of the biography. The dataset also contains an additional label about the gender of each biography.

We compare our framework with two model-based mitigation techniques: *Iterative Null-space Projection (INLP)* [64] and *Entropy-based Attention Regulation (EAR)* [4] (whose results will be reported in the rows in Table 6, columns 5 to 9). The *INLP* technique requires an additional annotation to each sample for the mitigated category only present for gender in this dataset. Therefore, we conduct two distinct analyses: i) focusing solely on gender-related protected attributes, and ii) examining all protected categories together. For the gender-related protected attributes, we compare both baselines with our word-removal mitigation strategy (MS2). We use only *EAR* as a baseline to mitigate all the protected categories instead. We chose our word-removal (MS2) for this comparison because it has shown similar performance in the trade-off between reducing the reliance on protected attributes while maintaining competitive predictive performance, but it has higher flexibility across datasets than the sentence-removal (MS1) (see §6.3).

5.4.1 Training the original occupation classifiers. The dataset contains 393,423 biographies for 28 occupations. We kept the same splitting, resulting in 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). Therefore, each task is highly imbalanced. 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.[14] The second column in Table 6 shows the macro F1 score on the test set for each original model trained for the five occupations (i.e., without applying any of the mitigation techniques). Those models are highly effective in classifying occupations. The model trained to classify the *journalist* occupation has the lowest performance, achieving a 0.89 macro F1 score. All the other models achieve a macro F1 score higher than 0.93. Still, such high classification performance could be achieved by heavily using protected attributes in predictions.

5.4.2 Identifying protected attributes in occupation classification. We used each fine-tuned original model to predict each occupation label over the entire test set and analyzed those texts using the *Explainer* component (instantiated with Integrated Gradients) to extract the most important words in predicting each occupation. Then, for each occupation, we annotated with GPT-3.5-Turbo the

[14] Inversely proportional class weights to the number of samples were used in the loss, as the training dataset is highly imbalanced.

top 400 words to identify protected attributes (corresponding to approximately 10% of the most important words of each occupation). We measured the models' reliance on protected attributes related to i) gender only, and ii) all categories, as shown in the third and fourth columns in Table 6. All the models moderately rely on protected attributes in the classifications. The *nurse* occupation is the most influenced by protected attributes, with a high reliance also on gender-related words, such as pronouns (e.g., *'she'*, *'her'*). Surprisingly, the *professor* occupation is the only one that does not rely on gender-related protected attributes.

5.4.3 Training the mitigated models. With our framework, we applied the word removal mitigation strategy (MS2) for each occupation on i) gender-related protected attributes only, and ii) all the categories simultaneously. Therefore, we trained two different mitigated models for each occupation class analyzed.

We trained one mitigated model for each occupation with the *INLP* methodology [64]. This technique mitigates the gender-related protected attributes only, as the original dataset contains the additional annotation only for gender. For all the other protected attribute categories, it is not applicable. Specifically, we used the original pre-trained BERT weights as the encoder. Then, the model multiplies the embedding representation of the [CLS] token from the last hidden layer extracted from 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 [64].

For the *EAR* technique [4], we used the same BERT architecture. We only added the regularization term to the loss function, with 0.001 as regularization strength. Notice that this technique does not allow the selection of which protected categories or words 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.

As in the previous experiments, we used all the mitigated models to predict the occupation labels over the entire test set. We analyzed those texts with the *Explainer* component to extract the most important 400 words for each occupation separately. Finally, we annotated those words with the *Identifier* component to evaluate if they exhibit a reduced reliance on protected attributes.

5.4.4 Results.

Fairness. Columns 7 and 9 in Table 6 show the number and percentage of protected attributes on which the mitigated models rely for the gender only and all categories separately. The number of these words already present among the protected attributes used by the original model is also indicated in square 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 effective or as effective as other baseline techniques in mitigating the use of such protected attributes. For instance, in the case of the *nurse* occupation, which represents a 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. The *INLP* technique obtained a similar mitigation effect, while the *EAR* technique 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%.

Occupation	*Original Model* F1 ↑	*Original Model* *Gender only* Ratio (%) PA ↓	*Original Model* *All Categories* Ratio (%) PA ↓	Mitigation Technique	*Mitigated Models* F1 ↑	*Mitigated Models* *Gender only* Ratio (%) PA ↓	*Mitigated Models* F1 ↑	*Mitigated Models* *All Categories* 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%)
Professor	0.938	0/400 (0%)	8/400 (2%)	*Our - MS2*	*N.A.*	*N.A.*	**0.941**	**4/400 [0] (1%)**
				INLP	*N.A.*	*N.A.*	*N.A.*	*N.A.*
				EAR	*N.A.*	*N.A.*	0.940	9/400 [3] (2%)

Table 6. **Mitigating occupation classification.** Summary of the results in mitigating the occupation classifiers. For each analyzed occupation, a different classifier is trained in one-vs-all settings. The original models for each occupation are highlighted in grey. The *Original Model* section shows the macro F1 (predictive performance) and the reliance on protected attributes (PA) for gender only and for all categories of the models trained without bias mitigation strategies. For each occupation, we applied our framework with the word-removal mitigation strategy on gender-related only and all categories of protected attributes training two different *mitigated* models. For the *EAR* [4] technique, a single model was trained but evaluated on different protected categories. *INLP* [64] is only applicable to gender-related protected attributes in this dataset. For each mitigated model are reported: i) the mitigation technique applied, ii) the macro F1 score for the occupation classification on the original test set (*predictive performance*), iii) the ratio and percentage of relied-upon protected attributes, with the number of those present in the original model in brackets (*fairness*). The objective of each *mitigated* model is to reduce the reliance on protected attributes while maintaining the original performance compared to the respective original model. The metrics indicated with ↑ (↓) represent higher (lower) values being better. The best performing for each metric is in bold.

The results are similar when considering all categories of 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 44%.

Predictive performance. Columns 6 and 8 in Table 6 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 model for each occupation (column 2). The mitigated models produced by our framework and the *EAR* technique achieve similar or sometimes even better performance than the original model on the test set (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 this technique 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 [53].

In summary, these results confirm the effectiveness of our framework in mitigating the use of protected attributes without sacrificing predictive performance in a different task where the protected attributes are strictly related to individuals (answering RQ3 in §1). Moreover, 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 is particularly useful when it is wanted to mitigate only a subset of protected categories because 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 in the classification task. Preventing models from using protected attributes will be a fundamental requirement for upcoming generations of predictive models to adhere to the standards imposed by regulators. The results of our experiments demonstrate that all the mitigation techniques, while reducing the number of protected attributes used in the predictions, maintain competitive performance in the model's accuracy. Among all the mitigation strategies, the removal-based approaches (MS1, MS2) have been shown to be the most effective in reducing the reliance on protected attributes while also increasing the predictive performance after the mitigation. We also evaluated the crowd-sourcing and LLMs-based protected attribute identification. We observed that the machine-in-the-loop LLM-based method performed better than the human-in-the-loop approach, allowing the automation of the entire mitigation process, and a dynamic updating of the dictionary of protected attributes.

6.1 Framework integration, versatility and complexity

Integration into existing NLP pipelines. Our framework can be integrated into existing NLP pipelines for two main purposes. First, the *Explainer* and *Identifier* can be used to measure and evaluate existing NLP classifiers' reliance on protected attributes. As a result, different models can be quantitatively compared not only through predictive performance and traditional fairness metrics (e.g., group performance disparities) but also through the use of protected attributes in predictions. Secondly, if the model exhibits extensive use of protected attributes, the entire framework can be used to facilitate training a new mitigated model with reduced reliance on protected attributes and competitive performance.

Enhancement of existing Bias techniques. Our framework can also be integrated to complement previous bias mitigation techniques 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. Instead, the *Identifier* can annotate protected attributes covering a broader range of categories. Indeed, many protected attributes, such as disability or religious belief, were rarely covered by prior studies. In this way, our *Explainer* and *Identifier* can be used as a starting point to improve other bias mitigation techniques acting in both the model and data spaces (i.e., replacing the moderator with their mitigation technique).

Versatility. Our framework is specifically designed to achieve *fairness through unawareness* by mitigating the model's reliance on protected attributes in predictions. It can address multiple protected attribute categories simultaneously. Therefore, it can potentially address intersectional bias, i.e., that encompasses multiple sensitive attributes together [27]. Moreover, our framework provides users the flexibility to choose which categories to mitigate, thanks to the fine-grained

annotation performed by the *Identifier* component. Such flexibility in the choice of categories 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 of our framework to produce the mitigated training dataset depends on many factors. Given a fixed model complexity, the execution times increase linearly with the growth in size of three key factors: i) the unlabeled corpus, ii) the number of most important words annotated, and iii) the size of the training dataset. Increasing the complexity of the model results in a slight increase in the execution time of all components.

We report here the execution time for the mitigation of the BERT model for the *nurse* occupation classification 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 to classify each input text of the unlabeled corpus. With a batch size of 512, it takes 846 seconds to classify each text in the unlabeled corpus (with an average of 0.01 seconds for each text). Then, the *Explainer* generates the explanations within each sentence (local-explanations) for all the texts predicted with the *nurse* occupation, in this case 4,071. Subsequently, it aggregates the scores to compile the overall list of the most important words. The entire 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 by applying the word removal mitigation strategy (MS2) of the *Moderator* on the 255.7k examples in the training set requires 29 seconds. Therefore, the total execution time is 2,134 seconds (35 minutes).

The output of our framework is a mitigated training corpus. However, an additional training phase is required to produce the mitigated model. The (re-)training time depends on the model's complexity and the original training data size. Some mitigation techniques do not change the dimensionality of the training dataset (MS2, MS3, MS5), so the training time for the mitigated model is the same as for the original model. In contrast, other techniques either decrease (MS1) or increase (MS4) the size of the training dataset, leading to a corresponding decrease or increase in training time. For the previously mentioned example, training the mitigated BERT model over 3 epochs takes approximately 1.5 hours.

6.2 Implications

Our research significantly contributes to the CSCW community by delving into the intricate landscape of human-AI collaboration, particularly within decision-making contexts. For example, our developed tool could assist humans in comprehending the hiring decisions made by NLP classifiers and address potential biases in the hiring process [59]. Moreover, 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. Such understanding fosters collaboration with machines to collectively arrive at fair decisions in content moderation. Our work has four main implications:

> **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. 13. Example of GPT-3.5-TURBO response for the annotation of the word {HEADSCARF}, categorized as *"Religion and belief"* with score 90/100.

Fully-automated framework. We release an open-source framework[15] that mitigates the use of protected attributes by NLP classifiers. By leveraging LLM annotations, the framework operates in a fully automated manner. As shown by the experimental results, it could be exploited to train a new classifier that mitigates the use of protected attributes without sacrificing performance in the classification task.

Other researchers can utilize this 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.

Compliant NLP classifiers. Our framework advances the state-of-the-art in promoting fairness in NLP applications by reducing reliance on protected attributes. The trained mitigated models exhibit enhanced fairness by significantly reducing their reliance on protected attributes while maintaining comparable and, in some cases, even better predictive performance. All mitigated models are shared on the HuggingFace platform.[16] Those models can be used as starting models for further fine-tuning on other datasets.

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 as moderation models for online platforms in compliance with AI regulations.

Protected attributes annotation. We advanced the state-of-the-art in the identification of protected attributes in NLP. Traditionally, protected attributes have been identified by static and manually pre-defined dictionaries covering only a subset of protected attributes, such as race or gender. 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 where protected attributes can be dynamically identified through straightforward prompts to an LLM model. This methodology enables the creation of a comprehensive and up-to-date dictionary covering all the protected categories simultaneously, which can be easily and rapidly updated periodically, ensuring its relevance in real-time linguistic landscapes.

Moreover, the LLM-based *Identifier* is able to annotate also proxies 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 13, where the word *'headscarf'* is annotated as related to *religion and belief*. However, we acknowledge that proxy words might not be exhaustive, and the accuracy of their identification should be better investigated in future work.

Throughout our research, we annotated approximately 15,000 words, 540 of which were labeled as protected attributes by the LLM. We release the dictionary[17] containing the full list of

[15]The code repository of our framework is available at anonymous-url-for-blind-review
[16]The fine-tuned mitigated models can be downloaded at anonymous-url-for-blind-review
[17]The dictionary is available in the GitHub repository.

protected attributes with the associated protected category. Our dictionary is more comprehensive with respect to other dictionaries available in the literature, which usually encompass just a subset of protected categories such as race and gender. Notably, our dictionary is designed to be dynamic, allowing continuous updates and enrichment over time by exploiting our protected attributes identifier. Other researchers can leverage this valuable resource to further advance the bias mitigation efforts in NLP models, promoting fairness in language processing. The dissemination and use by other researchers can improve the dictionary by making it more precise and accurate.

Humans vs. LLM annotations. Building upon a recent finding [26], our study demonstrates that LLMs 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 areas of concern

Our current approach has seven main potential limitations or areas of concern that should be considered when utilizing the framework, which could be addressed in future work.

Context unawareness. In our framework, the *Identifier* component exploits human-in-the-loop and machine-in-the-loop approaches to label individual words as protected attributes without considering the context in which they appear. In line with previous works in bias mitigation, we perform word-level identification to simplify the annotation task. However, this can also lead to inaccuracies if the same word has different meanings depending on the context. For example, the word *'black'* may be a protected attribute in one context (such as *"If you are a guy (black) or lesbian you get hired fast"*), but not in another context (such as *"I bought a new black desk"*). Also, the *Moderator* applies the mitigation strategies to protected attributes without considering the context of the sentence. The framework can more effectively mitigate protected attributes if it differentiates between words based on their context. However, this is a complex challenge that requires the consideration of context both in the identification and mitigation phases.

Performing human-in-the-loop annotation of the words within contexts is much more costly. Consider annotating 10 context sentences for each of the 400 most important words for the toxicity classifier as an example. It would require 4,000 context-aware annotations. The task would also be more complex, thus probably increasing the noise of the annotation.

The machine-in-the-loop approach is much more scalable and efficient and opens new possibilities to explore context-aware annotation. However, this makes prompt engineering more complex. Each context sentence will replace a new placeholder in the prompt. As a result, the prompt text can become really long (the context sentence is potentially very long). This can cause the LLM not to understand the task properly and, consequently, produce noisy responses [47].

We conducted a preliminary experiment to assess the impact of context-aware annotation on identifying protected attributes with LLM. We repeated the annotations of the most important words of the toxicity classifier by extracting up to 10 context sentences from the test set. Then, we performed the annotation with LLM, where each word was labeled as a protected attribute or not based on the context. Considering a word a protected attribute, if labeled as a protected attribute in at least one context (i.e., 10%), the overlap between word-level and context-level annotations is around 75%. However, we found that some of the context-level annotations are contradictory, especially for long context sentences. This finding reveals that, for this task and dataset, the context

does not seem to impact the annotation of protected attributes too much. However, future research should investigate whether this outcome is consistent across different datasets. Additionally, this necessitates further research into prompt engineering to enhance the effectiveness of context-level LLM annotation.

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. Thus, it is important to ensure an equitable distribution of crowdworkers across all protected categories and their respective subcategories. 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. For instance, 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. This progress will significantly enhance the effectiveness of our framework. Finally, introducing specific definitions of protected attributes, such as the nine protected 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 [52, 71]. These include challenges like the need for effective aggregation and normalization methods, and the contextual variability of words across different explanations. Such limitations can affect our framework. They may hinder the accurate extraction of the most important words used by the model, thus affecting the identification of protected attributes that the model relies on to make predictions. This issue can extend to the *Moderator* component, impacting the mitigated protected attributes. However, future developments in the XAI field could enhance our framework's effectiveness, given that our approach is adaptable and does not impose strict requirements on the choice of XAI methods used.

Defined protected categories. Our framework annotates protected attributes based on the nine protected categories outlined in the Equality Act 2010 in the United Kingdom. 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 areas. Since these categories were formalized in 2010, there have been characteristics that remain unaddressed by the Act. More than a decade later, initiatives are underway to broaden these categories to include aspects like socio-economic status, health status, genetic heritage, and physical appearance [56]. Future extension of this list 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 simply involves modifying the LLM prompt.

Synonyms. The removal-based mitigation strategies (MS1 and MS2) work by removing the sentences or words related to protected attributes from the training set. This can be an effective method for reducing the model's reliance on these attributes. However, there is a potential issue in that these strategies do not consider the synonyms of protected attributes. This could mean that a model might still rely on similar words that express the same meaning as the protected

attribute, undermining the intended mitigation. To address this issue, three potential solutions can be considered. One option is to use static synonym dictionaries, such as the Wordnet corpus in the Natural Language Toolkit (NLTK). Another option is to propose an embedding-based methodology that considers the semantic relationships between words and synonyms in the mitigation process. Finally, multiple mitigation rounds of our framework can be performed.

Mitigation with small training datasets or common protected attributes. In scenarios where the original training dataset is small or highly imbalanced, or the protected attributes are common words present in almost all input texts (e.g., *'he'* and *'she'* in biographies), the mitigation strategy based on sentence removal (MS1) may exhibit limited effectiveness. This is primarily due to the potential consequences of removing sentences that contain protected attributes from a dataset that is already limited or imbalanced. Additionally, the frequent presence of common protected attributes in most inputs may lead to the exclusion of nearly all sentences, significantly diminishing the quantity of training data available. Such a reduction can lead to a significant decrease in the model's accuracy. The decrease in accuracy can be too high and unacceptable in certain cases, and other mitigation strategies may need to be considered to mitigate the model's reliance on protected attributes while avoiding a significant reduction in performance. For example, the mitigation strategy based on word-removal (MS2) demonstrated similar performance in mitigating protected attributes while maintaining predictive performance, but has greater flexibility across datasets as it does not suffer from these issues. It is essential to strike a balance between mitigating the reliance on protected attributes and preserving the model's accuracy.

Fairness-privacy tradeoff. Our approach neither protects the privacy of individuals nor considers words or sentences to be private. By contrast, our approach focuses on constraining the classification to 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 both the human-in-the-loop or the machine-in-the-loop identification phase of the *Identifier* component. The annotation of protected attributes requires exposing the sensitive information contained in the text to individuals who are not necessarily trusted or to large language models. This creates a risk of privacy violations and could potentially lead to harm to the individuals whose sensitive information is being used, as reported by [1]. The privacy tradeoff must be carefully considered in implementing the proposed methodology, and appropriate measures should be taken to minimize the risk of privacy violations.

6.4 Future directions

Our framework has achieved promising results in reducing the reliance of the original model on protected attributes without sacrificing predictive performance. We evaluated its sensitivity to each component and its effectiveness in mitigating a state-of-the-art 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). However, there is still room for improvement in our approach. 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. This would require extracting the words and context information from the dataset, validating the reference of the words to protected attributes within each context, and applying mitigation strategies only to those sentences that contain protected attributes in similar contexts.

6.5 Ethical considerations

The data collected and used in this research project was done for research purposes only and in a responsible and ethical manner. To protect the privacy of individuals in the collected company reviews, the names of users were removed, and the analysis was only conducted at the company level, not at the individual level. Additionally, steps were taken to further decrease the risk of de-anonymization by paraphrasing sentences when presenting the results of the qualitative analysis.

All the people involved in the MTurk human study (crowdworkers) have been paid for their valuable contributions and time devoted to this research.

6.6 Author Positionality Statement

Acknowledging the impact of our experiences and backgrounds on our positionality is indeed of crucial importance. These factors can potentially shape our research process in multiple ways, particularly by introducing subjectivity. Our positionality can influence how we frame our research questions, design the methodology, conduct the study, and interpret and analyze the data [23, 36].

In this paper, we position ourselves as researchers in both academic and industry fields in a Western country[18] during the 21st century. Our team comprises a diverse group of n[19] females and n[20] males representing various ethnic and religious backgrounds. We bring expertise from different sectors, including Human-Computer Interaction (HCI), Artificial Intelligence, Responsible AI, and Natural Language Processing.

We recognize that our backgrounds and experiences have shaped our positionality. As HCI researchers affiliated with a predominantly Western organization, we acknowledge the need to broaden the understanding of the research questions and methodology presented in this paper.

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,

[18] Anonymized for blind review
[19] Anonymized for blind review
[20] Anonymized for blind review

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] Equality and Human Rights Commission (EHRC). 2018. Equality Act 2010. https://www.equalityhumanrights.com/en/equality-act/equality-act-2010 Accessed: June 2023.

[22] 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

[23] Hana Frluckaj, Laura Dabbish, David Gray Widder, Huilian Sophie Qiu, and James D. Herbsleb. 2022. Gender and Participation in Open Source Software Development. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2 (2022). https://doi.org/10.1145/3555190

[24] Adriana Solange Garcia de Alford, Steven K Hayden, Nicole Wittlin, and Amy Atwood. 2020. Reducing Age Bias in Machine Learning: An Algorithmic Approach. *SMU Data Science Review* 3, 2 (2020), 11.

[25] 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

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

[27] 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

[28] 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

[29] 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

[30] Nitesh Goyal, Ian D. Kivlichan, Rachel Rosen, and Lucy Vasserman. 2022. Is Your Toxicity My Toxicity? Exploring the Impact of Rater Identity on Toxicity Annotation. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 363 (nov 2022), 28 pages. https://doi.org/10.1145/3555088

[31] 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.

[32] 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

[33] 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

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

[35] 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

[36] Lucy Havens, Melissa Terras, Benjamin Bach, and Beatrice Alex. 2020. Situated Data, Situated Systems: A Methodology to Engage with Power Relations in Natural Language Processing Research. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, Barcelona, Spain (Online), 107–124. https://aclanthology.org/2020.gebnlp-1.10

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

[38] 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

[39] 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

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

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

[42] 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.

[43] 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.

[44] 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

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

[46] 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.

[47] 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]

[48] 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

[49] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. *CoRR* abs/1903.10561 (2019). arXiv:1903.10561 http://arxiv.org/abs/1903.10561

[50] 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

[51] 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

[52] 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

[53] 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]

[54] 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

[55] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *PloS one* 15, 8 (2020), e0237861.

[56] 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.

[57] 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

[58] 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.

[59] 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

[60] 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

[61] 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

[62] 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]

[63] 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

[64] 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

[65] 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

[66] 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

[67] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An Investigation of Why Overparameterization Exacerbates Spurious Correlations. arXiv:2005.04345 [cs.LG]

[68] 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

[69] 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

[70] Deven Shah, H Andrew Schwartz, and Dirk Hovy. 2019. Predictive biases in natural language processing models: A conceptual framework and overview. *arXiv preprint arXiv:1912.11078* (2019).

[71] 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

[72] 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

[73] 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).

[74] 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

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

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

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

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

[79] 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

[80] 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

[81] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating Unwanted Biases with Adversarial Learning. 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, 335–340. https://doi.org/10.1145/3278721.3278779

[82] 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.

[83] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 15–20.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009