# STILE: Exploring and Debugging Social Biases in Pre-trained Text Representations

Samia Kabir
Purdue University
West Lafayette, USA
kabirs@purdue.edu

Lixiang Li
Purdue University
West Lafayette, USA
li4256@purdue.edu

Tianyi Zhang
Purdue University
West Lafayette, USA
tianyi@purdue.edu

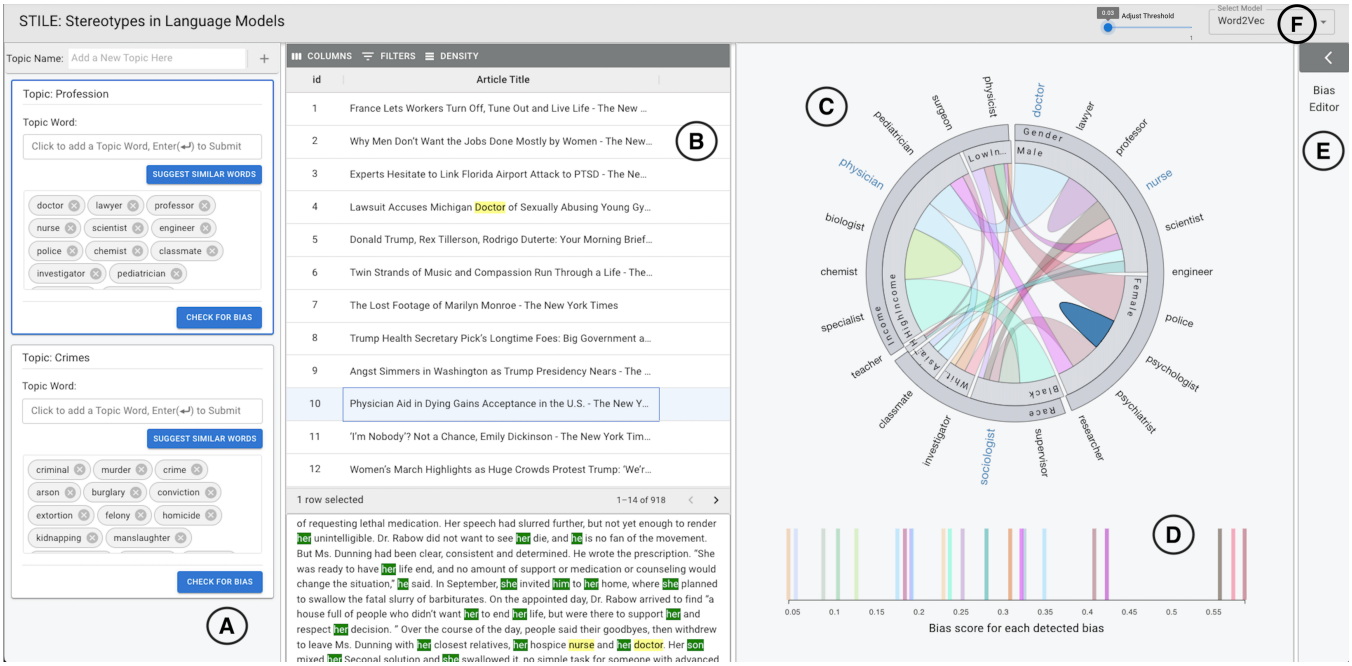



**Figure 1: STILE is an interactive system to explore and debug biases and stereotypes in pre-trained text representations. STILE includes six features: (A) a "Domain Lens" panel to explore a text corpus through user-defined and system-suggested topics, (B) an "Instance View" to explore the development of a bias from the training data, (C) a "Chord Diagram" that provides a bird's-eye view of detected biases, and (D) a "Strip Plot" that ranks detected biases based on their severity, (E) a "Bias Editor" to define and customize bias types, and (F) loading and selecting models.**

## ABSTRACT

The recent success of Natural Language Processing (NLP) relies heavily on pre-trained text representations such as word embeddings. However, pre-trained text representations may exhibit social biases and stereotypes, e.g., disproportionately associating gender with occupations. Though prior work presented various bias detection algorithms, they are limited to pre-defined biases and lack effective interaction support. In this work, we propose STILE, an interactive system that supports mixed-initiative bias discovery and debugging in pre-trained text representations. STILE provides users the flexibility to interactively define and customize biases to detect based on their interests. Furthermore, it provides a bird's-eye view of detected biases in a Chord diagram and allows users to dive into the training data to investigate how a bias was developed. Our lab study and expert review confirm the usefulness and usability of STILE as an effective aid in identifying and understanding biases in pre-trained text representations.



*CHI '24, May 11–16, 2024, Honolulu, HI, USA*

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642111

## CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Computing methodologies** → **Machine learning**; • **Software and its engineering** → **Software testing and debugging**.

## KEYWORDS

AI Fairness, Natural Language Processing, Word Embedding

**ACM Reference Format:**
Samia Kabir, Lixiang Li, and Tianyi Zhang. 2024. Stile: Exploring and Debugging Social Biases in Pre-trained Text Representations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3613904.3642111

## 1 INTRODUCTION

Pre-trained text representations such as word embeddings and language models are the backbone of many NLP models. They are widely used to initialize neural models, which are then fine-tuned for downstream tasks such as sentiment analysis [62] and text generation [19]. Despite the significant strides, there are increasing concerns over the fairness of pre-trained text representations.

Given that word embeddings and language models are trained on real-world text corpora, they run the risk of exhibiting, propagating, or even amplifying the social biases and stereotypes in these data. For instance, Bolukbasi et al. [6] show that a word embedding model trained on Google News disproportionately associates men with STEM-related professions and women with homemakers and caregivers. Furthermore, they demonstrate that these biases can further propagate to a broad range of applications built on top of word embedding models. Identifying biases becomes even harder when there are multiple combinations of biases (i.e., intersectional biases [10, 30, 38]). For instance, "nurse" is associated with "poor" and "female", while "engineer" is associated with "rich" and "male".

Since the seminal work on gender bias in word embeddings [6], many techniques have been proposed to detect social biases in pre-trained text representations [12, 41, 45–47, 51, 55, 63]. However, these techniques are limited to pre-defined biases, e.g., gender, race, occupation, etc. ML developers and practitioners cannot freely explore and examine any biases in their models. Furthermore, many techniques do not account for the fact that whether something is deemed as a bias can largely depend on the context. Thus, they may generate many false positives, requiring extensive human effort to verify. To address these limitations, it is desirable to provide interactive support that incorporates users' domain knowledge and intervention into the process of bias detection and analysis.

Though several interactive tools have been proposed to identify biases in tabular data [2, 11, 25, 27, 58, 61], little attention has been given to pre-trained text representations. To the best of our knowledge, the only related work is WordBias [24], a visual analytics tool for detecting intersectional bias in word embeddings. WordBias allows users to select words to examine and then visualizes detected biases in parallel coordinates. Since WordBias plots social subgroups related to a bias type in an axis of parallel coordinates, they are inherently limited to two social subgroups. For example, for the bias type of race, WordBias can only plot two racial groups. To handle more social subgroups, users have to add a new axis for each pair of subgroups. As subgroups increase, the number of axes will grow exponentially, leading to a severe visual scalability issue.

Moreover, a recent study shows that ML practitioners desire to understand the contexts of identified biases from the text corpus to quickly validate a detected bias [35]. This indicates a major need for bias debugging and validation. However, WordBias only answers "what" biases exist in a model. It does not provide any tool support for reasoning about "why" and "how" those biases are developed from the training data. Hence, the need for an interactive bias exploration and debugging tool persists.

To bridge this gap, we propose an interactive system, Stile, for ML developers and practitioners to explore and debug biases exhibited in pre-trained text representations. In Stile, a user can explore biases in any domain by creating a domain lens (e.g., profession) and supplementing topic words (e.g., doctors, lawyers) to define the domain. They can also freely define or customize the bias types (e.g., gender, race) they want to focus on. Then, Stile computes the association scores between topic words and bias types to identify potential biases in a model. The detected biases are visualized in a Chord diagram to provide users with a bird's-eye view of all detected biases and their prominence. By clicking on each bias in the Chord diagram, Stile filters the training corpus and highlights individual sentences that exhibit the selected bias in an instance view. In this way, users can inspect how exactly a bias is manifested in concrete contexts and perform an in-depth investigation.

The innovations in Stile are three-fold. First, the Chord diagram visualization in Stile provides a compact view of various biases that involve two or more social subgroups. This addresses a key limitation in WordBias—*parallel coordinates can only plot two social subgroups along an axis.* Second, Stile provides a novel debugging mechanism that allows users to trace back to the training data and investigate how a bias is exhibited in different contexts. In this way, users can easily answer questions such as "*is a race bias only exhibited when the surrounding context mentions a specific racial term?* ". By obtaining a deep understanding of how biases are shaped in the training data, users can more effectively develop counter strategies to mitigate the biases. Finally, the mixed-initiative exploration of data through user-defined topics and system-generated suggestions provides flexibility to discover biases in any topic of interest. Stile further enables users to create new bias types or customize the definition of existing biases based on their domain knowledge.

To investigate the usefulness and usability of Stile, we conducted a within-subjects user study with 15 participants who have ML experiences. The results show that Stile helps participants identify significantly more biases with fewer false positives compared to WordBias. Furthermore, when asked how a bias was developed, participants using Stile shared more insights into the reasons and context for each detected bias than those using WordBias. Additionally, we conducted a semi-structured interview with 6 experts who have expertise in ML/AI fairness and visual analytics tools. Our expert evaluation confirms the usability and utility of Stile in identifying and understanding bias, data and model selection, data cleaning and de-biasing, and digital humanities [20]. These results confirm that having such a tool to explore biases and stereotypes in pre-trained text representations as well as investigate the source of bias in data can be beneficial to ML developers and practitioners.

Overall, our work makes the following contributions:

- We propose an interactive system to explore, debug, and understand biases in pre-trained text representations.
- We propose a novel visualization that provides a bird's-eye view of prominent biases identified in a text corpus so that users can quickly and proactively identify and analyze biases.
- Our user study and expert evaluation confirmed the usefulness of our design and led to a set of design implications for

future tool development in the domain of bias detection and debugging in pre-trained text representations.

# 2 PRELIMINARIES, USER NEEDS, AND DESIGN RATIONALE

## 2.1 Background

*2.1.1 Biases and Stereotypes in Pre-trained Text Representation.* According to the Britannica Dictionary [18], stereotype means "*an often unfair and untrue belief that many people have about all people or things with a particular characteristic*". In other words, stereotypes are preconceived notions about a demographic subgroup. In this work, we adopt this definition for defining word bias and stereotype.

In textual data, stereotypes can be either at sentence level or word level, or both. A sentence is considered stereotypical if it makes a preconceived prejudicial remark against a demographic subgroup. These stereotypical sentences consist of some stereotypical words in the context of positive or negative remarks. For example, Asians are good at Math and Asians are bad at driving. Both of these sentences show preconceived notions toward the Asian population. Word stereotype is a boiled-down version of sentence stereotypes. In the example above, the word Asian is associated with Driving and Math. If a text corpus has a significant amount of sentences where Asians and Math are mentioned in the same sentence or same contexts together, a language model trained on that text corpus will show a strong association between the word Asian and Math. Thus, inspecting such word associations from a language model can give deeper insight into the prevailing stereotypes within a text corpus.

*2.1.2 Bias Metrics in Pre-trained Text Representation.* The importance of bias detection and mitigation in NLP models has been established in a large body of previous work [6, 12, 23, 28, 35, 49, 63]. To detect and mitigate biases, it is imperative to quantify the biases in an NLP model. Previous approaches have presented various bias metrics and quantification methods for NLP models [16]. Some widely used bias metrics include word-level metrics such as Word-Embedding Association Test (WEAT) [12] and Relative Norm Difference [23], as well as sentence-level metrics such as Context Association Test (CAT) [49].

WEAT [12] measures the relative bias of two groups of descriptive words (e.g., $\{programmer, engineer\}$ vs. $\{nurse, housekeeper\}$) with respect to two demographic subgroups (e.g., male vs. female). Here, each demographic subgroup is also represented by a set of words (e.g., $\{man, he, his, boy\}$ for male). WEAT computes the relative similarity of each descriptive word to each demographic subgroup based on the cosine similarity of word vectors. Then it runs the Implicit Association Test (IAT) to measure the statistical difference between the two groups of descriptive words in terms of their relative similarity to the two demographic subgroups. A model is considered fair if there is no statistical difference.

*Relative Norm Difference* [23] extends *WEAT* to compute the relative bias between any number of descriptive words, not just two groups. It computes the average of the vectors for each word in each of the two demographic subgroups as the group vector. Then, it computes the average Euclidean distance between each group vector and each vector for the given descriptive word(s). The difference between the average distances to the two demographic subgroups indicates the relative bias.

Unlike *WEAT* and *Relative Norm Difference*, *CAT* [49] is a sentence-level bias metric based on pre-defined templates. These templates are fill-in-the-blank sentences and the blanks are accompanied by three options—one stereotypical word, one anti-stereotypical word, and one neutral word. These templates are used as prompts for pre-trained language models such as BERT and GPT-2 to measure the likelihood of each option filling in the blanks. The bias is measured based on the model's preference for the stereotypical option. A limitation of *CAT* is that it overly depends on pre-defined templates, which makes it hard to extend to new, undefined biases and stereotypes users want to investigate.

In this work, we use *Relative Norm Difference* as our bias metric for two reasons. First, it is not limited to any pre-defined templates and can be used to compute the association between any individual words and demographic subgroups. Second, it is computationally efficient, enabling a quick response to user requests. Please refer to Section 3.2 for more details of our implementation of *Relative Norm Difference* and Section 2.2 for a comprehensive overview of existing bias detection and mitigation approaches in NLP.

## 2.2 Related Work

*2.2.1 Need for Bias Detection and Mitigation.* Many recent studies demonstrate the need for helping ML developers and practitioners to build fair systems [15, 35, 56]. A large body of previous work focuses on definitions and algorithms to define, identify, and mitigate unfairness [1, 6, 14, 33, 42, 50, 53]. Some work also analyzes socio-technological aspects of fair algorithms and systems [4, 5, 21, 28, 31, 44, 59]. Holstein et al. [35] conducted semi-structured interviews and surveys with ML practitioners to identify the need for supporting bias detection and mitigation in ML systems. They highlighted the fact that the design of fair ML systems should take into consideration the actual needs in the real world instead of solely focusing on algorithmic methods. Their interviews and surveys showed that developers often struggle to identify and mitigate bias in data due to limited time and resources. Also, teams often struggle with "unknown unknowns". Gu et al. [28] compared the effectiveness of different bias detection and de-biasing tools, e.g., explanatory systems and recommendation systems. They identified the pros and cons of each system and generated a list of requirements for informed design of any future iterative de-biasing tools. Law et al. [43] conducted an in-depth interview study with 11 ML practitioners to investigate the need for human-in-the-loop tools for bias detection. They also came up with design considerations for tools to help ML practitioners in detecting biases in ML systems.

*2.2.2 Bias Detection and Mitigation in NLP Models.* Identifying and mitigating social biases and stereotypes from unstructured text data has gained a lot of attention lately. Bolukbasi et al. [6] presented an association test-based algorithm to detect gender stereotypes from word embeddings trained on Google News articles. Their algorithm found a staggering amount of gender stereotypes, e.g., bias between females and homemakers. They also proposed a bias-mitigating algorithm by directly modifying the embeddings. Caliskan et al. [12] found that NLP models built on top of biased word embeddings can contain similar biases. Jentzsch et al. [36] conducted a similar

experiment where they prompted language models with moral choice questions, e.g., "should I kill people?". Their work showed that social, ethical, and moral biases are imprinted in text corpora and can propagate to downstream ML models.

Although most of the prior works focus on context-free word embeddings, recent work has started moving towards bias detection and mitigation in large language models as well [30, 39, 49, 57]. Tan et al. [57] proposed to use word-level and sentence-level encodings to measure biases while capturing the context of the word and sentences. They suggested that both encodings are important for bias detection and mitigation since they capture the context of input in different ways. Kirk et al. [39] proposed a template-based method to identify occupational biases in pre-trained GPT-2 models. Nadeem et al. [49] presented a large-scale stereotype benchmark called StereoSet. StereoSet can be used to measure stereotypes against gender, race, profession, and religion.

*2.2.3 Interactive Support for Bias Detection and Mitigation.* Many interactive systems have been proposed to detect biases in tabular data [2, 3, 11, 25, 60, 61]. Cabera et al. [11] presented FairVis, a novel visual analytics tool for users to audit fairness and biases in ML models. FairVis allows users to incorporate their own domain knowledge to define new demographic subgroups and compare fairness metrics among groups. FairVis demonstrates the usefulness of interaction support in helping data scientists and the general public to understand and generate fair systems and algorithms, which motivates our work. However, the design of FairVis only works for tabular data, not unstructured text data.

Ahn et al. [2] proposed a fairness framework called FairDM that identifies a set of tasks for each step (understand, identify, measure, and mitigate) of the fair decision-making process. They further proposed a visual analytics tool, FairSight, that demonstrates that integrating FairDM into an interactive tool helps to achieve fairness-aware decision-making. Wang et al. [58] presented DiscriLens, which considers the intersectional property between discriminatory item sets by introducing a novel visualization named RippleSet. DiscriLens provides guided visual interactions to help users understand, analyze, and reduce algorithmic discrimination. Silva [61] helps users explore the source of unfairness in data and ML model through a global casual view. It links causality with quantitative metrics and visualizes causality graphs to answer "what-if" questions when assessing the model's fairness. D-BIAS [25] presents a causality-based interactive network that users can use to identify any bias and modify the graph to generate de-biased tabular data for future use. Their human-in-the-loop bias mitigation system shows increased de-biasing performance on real-world data and at the same time increases user trust and accountability.

Unlike our work, all the aforementioned approaches are designed for structured or tabular data. So far, little attention has been given to providing interactive support for detecting, exploring, and mitigating biases in unstructured text data. To the best of our knowledge, the only interactive system that detects biases in text corpora is WordBias [24]. WordBias uses the relative norm difference metric to detect biases in context-free word embeddings and then visualizes the detected biases in a parallel coordinate view. Compared with Stile, WordBias supports bias types with only two subgroups, e.g., only "white" and "black" for "Race", "male" and "female" for "Gender". Furthermore, WordBias provides no mechanisms for users to trace back to the training data and investigate how a bias is developed from the training data.

## 2.3 User Needs and Design Rationale

To understand the user needs of bias detection in NLP models, we reviewed research papers [2, 11, 24, 25, 61] that have proposed visual analytics tools for detecting and mitigating biases, as well as related need-finding studies [28, 35]. Based on the findings and design implications from these papers, we summarized the design goals for Stile and elaborated the rationale for each goal below.

**G1. Help users explore biases in areas of interest.** A recent study finds that data scientists, ML developers, and ML practitioners wish to interactively explore biases related to a domain or a topic they are interested in, e.g., "Profession", "Crime", etc. [35]. Defining a topic for bias discovery is challenging, since users need to provide a comprehensive set of topic words, such as different kinds of jobs related to the topic of "Profession", for a thorough examination of potential topic-related biases. Inspired by faceted search [34], we design a feature called Domain Lens to allow users to define and customize different groups of topic words for bias detection.

**G2. Help users automatically quantify biases.** Previous bias detection techniques [11, 24, 25] have shown that a tool should accurately quantify and measure bias so that a user can identify the degree and severity of the bias. Stile uses *Relative Norm difference* to detect biases, which accurately quantifies the association of every word with different demographic subgroups (Section 3.2).

**G3. Help users quickly explore detected biases.** Ghai et al. [24] have shown that exploring biases should be quick and intuitive. To support this, Stile provides a high-level overview of all detected biases in a Chord Diagram. This helps users quickly explore biases in association with multiple demographic subgroups (Section 3.3). Furthermore, Stile allows users to identify specific biases related to a descriptive word or a demographic subgroup through selective highlighting (Section 3.3).

**G4. Help users with an effective comparison of biases.** Cabrera et al. [11] have shown that a visual interface should support effective comparison of different biases. In Stile, a user is able to compare the severity of different biases with the help of the Chord Diagram and the Strip Plot. The Chord Diagram utilizes the chord widths to help users compare the severity of detected biases among different words and demographic subgroups. To help users with more effective comparisons, Strip Plot provides a sorted view of all detected biases based on their severity (Section 3.3).

**G5. Help users verify and debug detected biases.** Holstein et al. [35] have shown the necessity of understanding, debugging, and validating detected biases. As shown by Yan et al. [61], an effective way to understand the source of a bias is to investigate the training data. In Stile, we allow users to trace a bias back to relevant sentences in the training corpus by automatically filtering the instance view as users select a specific bias (Section 3.4.1). Furthermore, users can proactively search and filter the training corpus with specific words of interest in the instance view.

**G6. Help users define, modify, and create bias types.** A user should be able to investigate biases against common demographic

**Table 1: Example of topic words suggested by Stile**

| User-provided Words | Recommended Words |
|---|---|
| criminal, crime, murder | arson, burglary, conviction, extortion, felony, homicide, kidnapping, manslaughter, misdemeanor, offense, perjury, prosecution, robbery, crimes, murders, rape, felonies |
| doctor, nurse, engineer | chemist, dentist, police, firefighter, instructor, mechanic, pediatrician, physician, trainer psychiatrist, psychologist, supervisor, surgeon, teacher, technician, janitor, biologist |
| pretty, beautiful, fat | charming, cute, delicious, funny, incredible, lovely, perfect, sexy, sweet, wonderful, weird, scary, cool, nice, boring, pleasant, silly, strange, crazy, creepy |

subgroups (e.g., race, gender, etc.), as well as subgroups based on a user's specific needs (e.g. income, marital status, etc.) [24]. Holstein et al. [35] have shown that a user should also be able to intervene in the bias detection process. In Stile, we provide a set of pre-defined bias types to start with. Users can modify these bias types, add new bias types, and observe the word associations with the changed bias types in real-time (Section 3.4.2).

## 3 STILE: BIAS DETECTION AND DEBUGGING IN PRE-TRAINED TEXT REPRESENTATIONS

Figure 1 shows the four major features in Stile—(1) the *Domain Lens* to define and explore different topics in a text corpus (Section 3.1), (2) the *Chord Diagram* that provides an overview of all detected biases (Section 3.3), (3) the *Instance View* for validating a bias and investigating how it is exhibited in the training corpus (Section 3.4.1), and (4) the *Bias Editor* for defining and customizing the types of biases of interest (Section 3.4.2). These four features help model developers and ML practitioners examine and debug the potential biases in a pre-trained text representation.

### 3.1 Domain Lens: Topic-based Data Exploration

To support G1, Stile allows users to create a domain lens with a group of descriptive words they are interested in. A domain lens can be of any topic, e.g., "Crime", "Profession", "Personality", etc. Then, users need to supplement some initial words to define the domain. For example, a user can add "doctor", "nurse", and "teacher" as a starting point for the lens of "Profession".

To avoid manual efforts of enumerating many words to define a domain, we provide a mechanism that recommends semantically relevant words based on the initial set of user-provided words. To identify semantically relevant words, Stile first computes the word embedding of each user-provided word using Word2Vec [48] and then computes the average of these embeddings. Then, Stile retrieves the top 20 words that have high cosine similarity with the average embedding vector. We use average embedding instead of a synonym generation tool since average embedding represents a specific direction in the continuous vector space of word embeddings. For example, for topic words such as "doctor" and "teacher", this mechanism not only provides some synonyms but also generates words covering various ranges of professions, e.g., "lawyer", "police", etc. Also, the average vector will generate suggestions even if a user enters antonyms as topic words. For instance, if a user enters "ugly" and "pretty" together, the tool will generate physical appearance-specific words. Table 1 shows some examples of Stile's recommendations given different sets of user-provided words.

Users can remove or edit words recommended by Stile, or continue soliciting more words. Once users are satisfied with the set of words to define a domain, they can proceed to detect biases related to this domain. Stile will filter the text corpus and retain instances that contain at least one of the topic words in this domain lens.

### 3.2 Bias Detection

To support G2, we use a popular bias detection algorithm, *Relative Norm Difference* [23] to detect biases in pre-trained text representations. *Relative Norm Difference* is a fast and extensible metric. It computes an association score from a topic word towards a group of words that define a demographic subgroup. If the association score is higher for one subgroup than the others, the topic word is considered biased against that demographic subgroup. Previously, it has been used to successfully identify biases against several demographic subgroups related to gender, race, religion, etc. [7, 23, 24, 46].

Since the original *Relative Norm Difference* algorithm only considers biases between two demographic subgroups, we extend it to account for any number of subgroups. Our extended algorithm computes the average word embedding vector of the words that define a demographic subgroup. For each subgroup, it computes the cosine similarity between the average vector of that group and the vector of a given topic word. This cosine similarity is used as the association score for computing bias. If the association score of one subgroup is higher than the scores of any other subgroup by a threshold, the given topic word is considered biased against that subgroup. For this tool, we selected the default threshold to be 0.03. We selected this threshold with a trial and error basis to see at which threshold the tool shows a reasonable amount of biases without overwhelming the users. Users are allowed to adjust this threshold based on their own preferences.

To verify that our extension to the *Relative Norm Difference* [23] algorithm does not change the bias calculation process, we reproduced their results following the same process of some of their experiments. We extracted occupational biases using Stile from Google News Embedding and compared them with racial and gender biases that were identified by the *Relative Norm Difference* algorithm. The result shows that 80% of biases detected by Stile are similar to those identified by the original algorithm.

Compared to prior work [24], Stile not only supports bias detection in context-free word embeddings such as Word2Vec [48] and GloVe [52] but also supports bias detection in contextualized word

**Table 2: Bias glossary of Stile**

| Bias Type | Subgroups | Words to define the subgroups |
|---|---|---|
| Gender | Male | he, his, him, male, man, men, boy, boys, son, guys, himself, Mr., males |
| | Female | she, her, hers, female, woman, women, girl, girls, herself, Mrs., Miss, females |
| Race | Black | black, colored, blacks, african_american, dark_skinned, afro, african |
| | White | white, whites, caucasian, caucasians, caucasoid, light_skinned, european |
| | Hispanic | hispanic, hispanics, latino, latina, spanish, mexican |
| | Asian | asian, asians, chinese, japanese, indian, indians, korean, China, Japan, Korea |
| Income Level | High | rich, wealthy, affluent, richer, richest, affluence, advantaged, privileged, millionaire, billionaire, exquisite, extravagant, lavish, moneyed |
| | Low | poor, poors, poorer, poorest, poverty, needy, penniless, moneyless, underprivileged, homeless, bankrupt, insolvent |
| | Mid | middleclass, workingclass, middleincome, whitecollarclass, bourgeois, bourgeoisie |
| Age | Young | young, youth, teenager, teen, children |
| | Old | old, elderly, aged |

embeddings such as BERT [17] and GPT-3 [9]. Unlike context-free embeddings, contextualized embeddings take into consideration the surrounding context of a word. Therefore, language models require a sentence, rather than a single word, as input to compute contextualized embedding vectors for words. To handle this unique requirement, for a given word, Stile retrieves its word embedding vectors in various contexts from a pre-trained language model by searching for sentences that mention this word in the training corpus. Then, it computes the average vector and uses it to compute its association with a subgroup. Since this process is computationally intensive, we pre-computed and cached the contextualized word embeddings for all words in the vocabulary of a language model to improve the responsiveness of our interactive system.

We repeat the same process for all topic words in a domain lens and all bias types defined by a user. Since relative norm difference is fast to compute, this bias detection process is done in real-time. Every time a user adds a new topic word or a new bias type, a new set of biases will be computed immediately at run time. By default, Stile provides a pre-defined set of bias types, as shown in Table 2. When using Stile, users can add, remove, and edit bias types using the bias editor (detailed in Section 3.4.2).

## 3.3 Bias Visualization

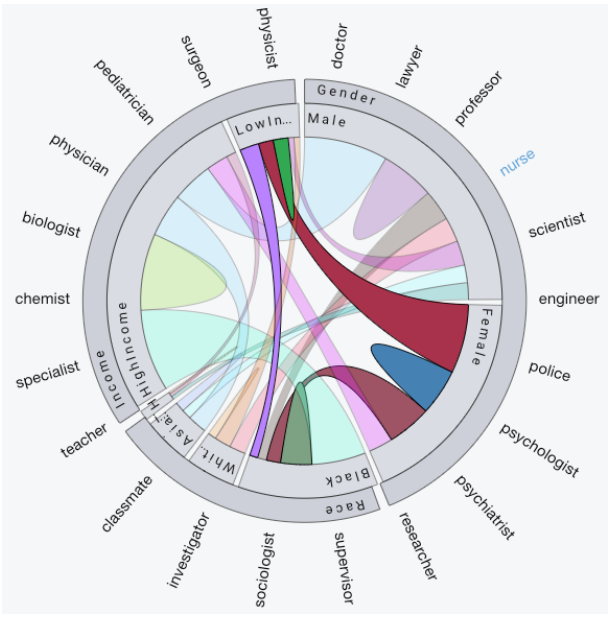
To support G3 and G4, Stile uses a chord diagram to provide a quick overview of detected biases. It also uses a strip plot to render detected biases in a sorted order based on their severity.

*3.3.1 Chord Diagram.* Our goal is to render all biases in a scalable and space-efficient way, especially accounting for intersectional biases and bias types with more than two demographic subgroups. In the initial design, we experimented with several graphs and network visualizations that can effectively visualize inter-entity relationships. One of our initial design considerations consisted of the Sankey Diagram, which is a version of a Parallel Axis plot. Each axis is divided into multiple parts in a Sankey Diagram. However, with the growing number of bias types added as new axes, the visualization is stretched horizontally. Furthermore, when new bias dimensions are added for a bias type, the number of nodes for each axis increases and expands the visualization vertically. This consumes a significant amount of screen space in the interface. Then again, we need a design that does not expand the visualization horizontally or vertically with an increasing number of bias types or bias dimensions. Hence a concentric design is best suited for this purpose. We eventually chose the chord diagram, since it is concentric and can visualize a large number of biases without consuming a lot of space in the interface.

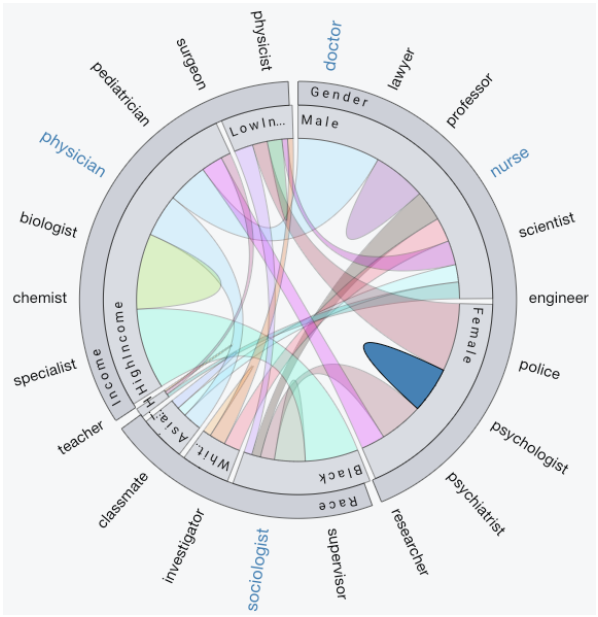
We adapt the original design of chord diagrams to render the relationships between topic words and biases. Our chord diagram consists of two concentric circles instead of one. The outer circle renders all topic words in a domain lens. The arcs in the inner circle represent different demographic subgroups. The chords that lay inside the inner circle represent biases. Stile automatically assigns a unique color to each detected bias. Chords that start and end at the same arc represent individual biases. Otherwise, they represent intersectional biases. Two chords connecting three arcs represent intersectional bias between three subgroups. For example, a chord from "Female" to "Low Income" and a chord from "Female" to "Black" represent an intersectional bias between "Female", "Black", and "Low Income".

The width of chords towards arcs represents the weight of identified biases. This weight is the accumulation of bias scores of all the words that are associated with a bias. For example, in Figure 2(a), the chord from "Female" to "Low Income" is wider towards "Female" and narrower towards "Low Income". This means the words associated with these biases have more association with "Female" and less association with "Low Income". The reason for visualizing all topic words and biases in and around the circle is to help users focus their attention in the same area. Placing the topic words away from the circle would require users to move their eye gaze from the chord diagram every time they inspect a new word. As per our design goal [G3], we designed these two concentric circles for better usability and quick exploration.

The chord diagram of Stile is made highly interactive to help users quickly explore different biases. When a user hovers over a word in the outer circle, all biases associated with that word are highlighted (Figure 2(a)). When a user hovers over a chord, all words

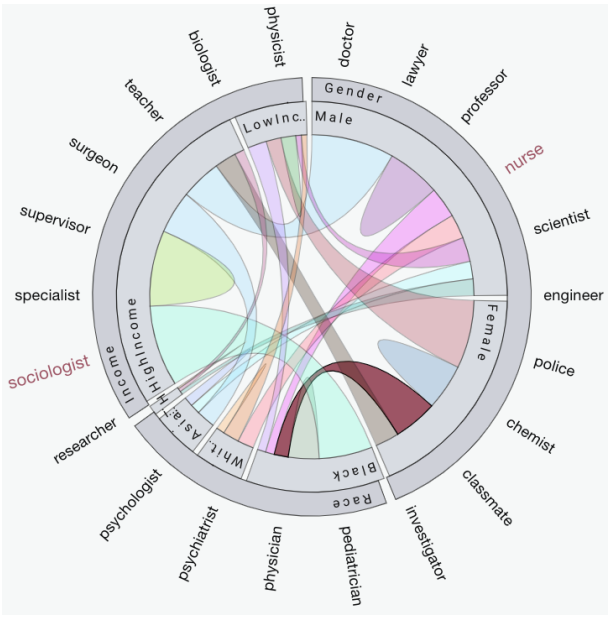



(a)

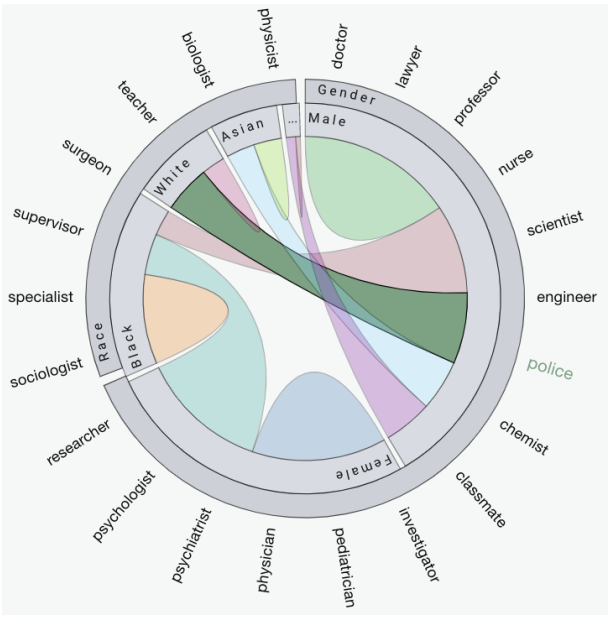



(b)

**Figure 2: The chord diagram of biases detected from Word2Vec. (a) All biases associated with the word "nurse" are highlighted. (b) All words biased against the "Female" subgroup are highlighted in the same color as the bias.**




(a)




(b)

**Figure 3: The simplification of a Chord Diagram. (a) Before simplification. (b) After disabling the bias type of "Income".**

associated with that bias are highlighted in the same color (Figure 2(b)). In this way, users can easily observe the biases associated with a word, or words associated with a specific bias.

*3.3.2 Strip Plot.* The strip plot ranks and renders all detected biases based on their bias score (Figure 1 ①). This helps users to prioritize which bias to investigate first. Each strip in the strip plot corresponds to a bias in the chord diagram. When a user hovers over a strip, it shows a tooltip with information about the corresponding bias, e.g., its bias score, and also highlights the corresponding chord in the chord diagram along with the associated words. The closer a strip is to the right end of the plot, the higher the bias score is. Hovering over chords inside the chord diagram highlights the

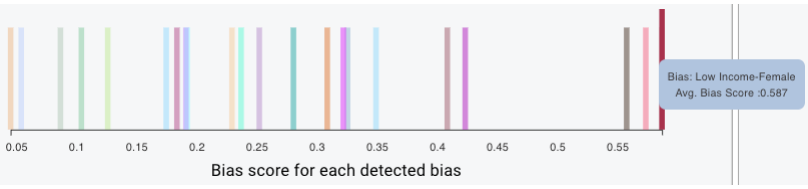



**Figure 4: The strip plot of biases detected from Word2Vec. The most severe bias is against "Female"-"Low Income".**

corresponding strip in the strip plot to guide user attention to the severity of the corresponding bias. Appendix B shows an example of the interaction between the chord diagram and the strip plot. Figure 4 shows the strip plot for the biases in the chord diagram of Figure 2. The most severe bias for this scenario is the bias against the intersectional group—"Female" and "Low income".

## 3.4 Debugging and Customizing Biases

*3.4.1 Instance View.* To support G5, Stile uses the instance view. The instance view allows users to trace back to concrete texts that exhibit a bias in the training corpus of a pre-trained text representation model. If users want to click on a bias in the chord diagram or strip plot, the instance view will automatically be filtered to only show sentences that manifest that bias.

The instance view has two parts. The top part renders the title of each document or the first sentence in the document if no title exists (Figure 5ⓐ). When users click on the title, the article is expanded at the bottom part (Figure 5ⓑ). Furthermore, Stile highlights the topic words and the subgroup-related words in the document. This highlighting feature makes it easier for users to quickly glance over a lengthy document and locate where the bias is manifested.

The instance view also allows users to search from training data. We implement an efficient regular expression matching mechanism, so the search can be done in real-time over a large corpus. Additional support, e.g., filtering, and sorting of instances are provided to make the bias explanation experience more effective.

***Scalability.*** At this point, our data explorer can scale to 100k documents. For bigger corpora, scalability can be addressed by utilizing advanced indexing & ranking functions like BM25 [54], which has been shown scalable to the entire Yelp Corpus [13].

*3.4.2 Bias Editor.* To support G6, Stile provides a bias editor to define and customize bias types. If a user wants to know how a demographic subgroup is defined, they can expand the bias type and examine the words used to define this subgroup (Figure 6(a)). Users can also edit those words (Figure 6(b)). When a user adds a new bias type, a color with a high hue is automatically assigned to it and the newly added bias type is highlighted with that color in the instance view (Figure 6(c)). Furthermore, a user can disable any existing biases. This helps users focus on the bias types of interest without getting overwhelmed by many potential biases.

# 4 USAGE SCENARIO

This section describes how Stile can be used to facilitate three use cases—(1) discovering, validating, and debugging biases in pre-trained text representations, (2) comparing the fairness of different

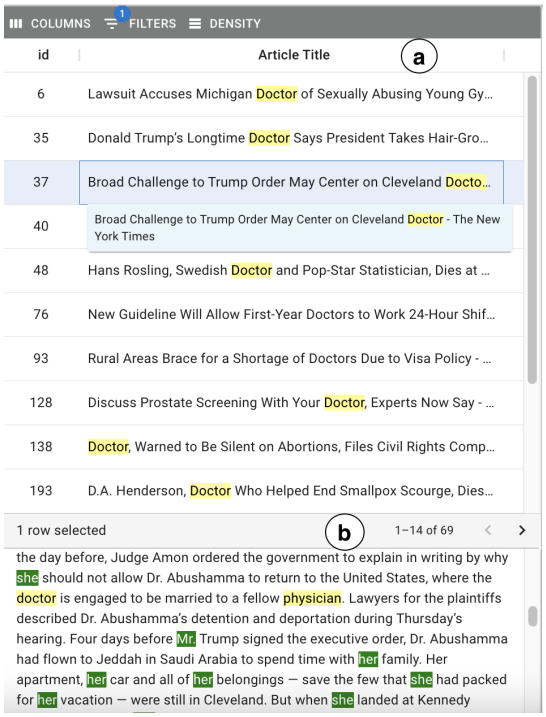


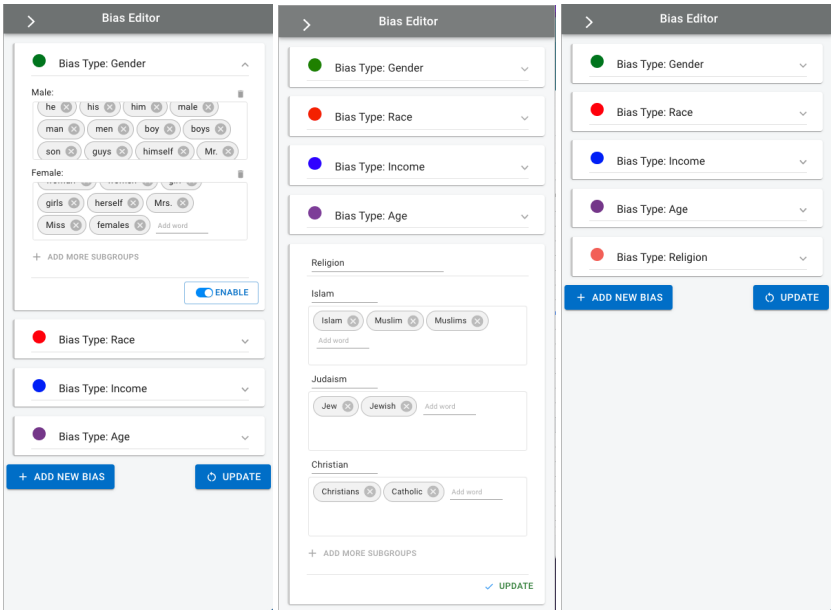

**Figure 5: The "Instance View" with an expanded instance that explains the bias between "doctor", "physician" and "female".**

word embedding models, and (3) cleaning and de-biasing of the training data for NLP models.

## 4.1 Usage Scenario 1

Suppose Alice is a machine learning developer and she needs to use a word embedding model called Word2Vec to encode text data in her application. Alice is aware that word embedding models can exhibit biases and stereotypes, so she uses Stile to examine the underlying biases in the Word2Vec model.

After opening the tool, Alice sees the "Domain Lens" panel on the left (Figure 1Ⓐ). This panel allows Alice to explore biases in any topic she is interested in. Alice is curious to know if there are any social stereotypes between different demographic subgroups and professions. So, she creates a new topic lens named "Profession". Then Alice starts adding different profession-specific words to define the topic. She enters topic words such as "doctor", "engineer", "nurse", etc. Since Alice does not want to enumerate all professions manually, she asks the system to generate similar profession-specific




(a) (b) (c)

**Figure 6: The "Bias Editor" with (a) Expanded bias type "Gender" (b) A new bias type "Religion" is being added, (c) New bias type added with assigned color for explanation and highlight.**

words for her by clicking on the "SUGGEST SIMILAR WORDS" button. This generates a set of new profession-specific words, such as "lawyer" and "teacher". Alice examines the system-generated words, makes a few edits, and moves on to the next step.

To detect biases related to professions, Alice clicks on the "CHECK FOR BIAS" button at the bottom of the "Profession" lens. Stile then runs the bias detection algorithm to identify topic words that are disproportionately correlated with different demographic subgroups. The detected biases are then rendered in a chord diagram (Figure 1Ⓒ). Alice notices that the topic words, such as "doctor" and "teacher", are rendered in the outer circle of the chord diagram. The biases associated with these words are rendered as chords inside the inner circle of the chord diagram.

When Alice hovers over a chord, the topic words associated with the corresponding bias are highlighted in the outer circle (Figure 2(b)). For example, when hovering over the bias chord "Female"-"Black", Alice can see that words such as "nurse" and "sociologist" (Figure 3(a)) are highlighted in the same color as that bias. Also, the chord width tells Alice that these words are more biased towards the "Female" subgroup and less biased towards the "Black" subgroup. She further examines some topic words by hovering over each of the words, and then the tool highlights the associated subgroups inside the chord diagram. For instance, when she hovers over the word "nurse", she can see that the word "nurse" is biased towards "Black", "Female", and "Low-Income" subgroups.

Alice wonders which biases are the most prominent ones that deserve more attention to mitigate. She turns to the strip plot below (Figure 1Ⓓ). The strip plot renders the detected biases in strips with the same color as in the chord diagram. Alice checks the rightmost strip in the plot and sees that the bias "Female"-"Low Income" is the most severe bias with the highest bias score (Figure 4).

Alice is curious about how this bias was developed from the training corpus. So she clicks on the "Female"-"Low Income" bias in the chord diagram. Then, Stile automatically filters the instances that contain this bias in the instance view (Figure 1Ⓑ). To help Alice quickly recognize the bias-related terms in the relevant instances, Stile highlights those terms in each sentence. For example, Alice clicks on some instances and sees that words such as "she", "her", "girl" are highlighted. After inspecting several instances, Alice gets the idea that the model learns a *superficial correlation* between "nurse" and female pronouns, which leads to this bias.

Alice is also curious to know how the "Male" and "Female" demographic subgroups are defined. So, she expands the bias editor (Figure 6). She examines the representative words that define the "Female" subgroup. She deletes some words, e.g., "miss" from the list, and adds a new word, "feminine", to the list. Then she clicks the "UPDATE" button to observe if there are any changes in the identified biases or bias scores. Alice finds the chord diagram contains many different types of biases in it, some of which she does not care about. So Alice disables some biases in the bias editor to only inspect specific biases she cares about. She disables the bias type "Income Level" and continues with only racial and gender biases (Figure 3(b)). She further uses the "ADD NEW BIAS" feature to create a bias type of "Religion" and observes newly detected biases (Figure 6(b)(c)). Alice then continues to create more topics or more iterations based on her needs. Table 3 shows a list of severe social stereotypes found by Alice in this Word2Vec model.

**Table 3: Stereotypes identified in Word2Vec trained on news data from CNN, NY Times, Breitbart, and Business Insider**

| Bias | Domain | Associated Words |
|---|---|---|
| "Female"-"Black"-"Low-Income" | Crime | extortion, offence, rape, criminality |
| "Male"-"Black"-"Low-Income" | Crime | criminals, killing |
| "Black"-"Young" | Crime | kidnapping, murder, rape, violence, homicide, burglary, incarceration |
| "Female"-"Low- Income" | Profession | nurse, seamstress, homemaker, teacher, educator, nanny, librarian |
| "Male"-"High Income" | Profession | engineer, dentist, scientist, captain, entrepreneur, admiral |
| "Male" | Sports | football, baseball, boxing, chess |
| "Female" | Sports | soccer, volleyball, gymnastics |
| "Low-Income" | Transportation | train, bus, cab, ferry, convoy, boat |
| "High-Income"- "Male" | Transportation | cabin, jet, ship |
| "Black"- "Male" | Personality | loud, dumb, silly, angry, condescending, funny |
| "White"- "Male" | Personality | paranoid, dismissive, polite |

## 4.2 Usage Scenario 2

Suppose Alice finds two models, Word2Vec and GloVe, trained on the same dataset. She wants to integrate the model with fewer biases and stereotypes in her application. To compare the biases and stereotypes in these two models, she opens Stile and loads both of them. She follows the same process as Usage Scenario 1 (Section 4.1) to create a few sets of topics to examine. Next, by using the "Select Model" feature (Figure 1Ⓕ) at the top-right corner of Stile, she toggles between Word2Vec and GloVe. By examining the number of chords in the chord diagram (Figure 1Ⓒ) and the number of strips in the strip plot (Figure 1Ⓓ), she notices that GloVe is portraying more biases than Word2Vec in "Profession" topic (Figure A.1, Appendix A). She keeps comparing the number of biases made by both models for each of the other topics by switching between these two models. She finds out that for the majority of topics, GloVe is making more biases and stereotypes. To understand how these biases are exhibited in the dataset, she uses the instance view (Figure 1Ⓑ) and notices GloVe is capturing more superficial relationships from the training data. Hence, she decides to use Word2Vec for this specific dataset.

## 4.3 Usage Scenario 3

Suppose Alice is a data scientist who is trying to de-bias a dataset so that the models trained on this dataset would not make gender and racial biases. Alice is using a de-biasing method called Gender Swapping [64] to clean the data. The intuition behind Gender Swapping is to create a new training corpus by replacing all male pronouns with female pronouns and vice versa and combining the original and modified corpora to train a model. Alice is trying to extend this method to substitute all gender and race-specific words in the training data, rather than just male and female pronouns. However, Alice does not know which gender and race-specific words she should alter. So she decides to use Stile and examines a Word2Vec model trained on the original corpus. Alice uses the bias editor (Figure 6) to see which words define different demographic subgroups such as "Male", "Female", "Black", "White", "Asian", etc. Furthermore, she uses the instance view (Figure 1Ⓑ) to examine a few samples from the training data to see how gender and race-specific words are used together with other words in the corpus. While inspecting sample training data, she picks up a few more race-specific words that are not present in the bias editor. She takes notes of all the gender and race-specific words and moves on to de-bias her training corpus and word embeddings. Finally, she trains her Word2Vec model again with the newly de-biased corpus and uses Stile as a checkpoint to verify the progress of the de-biasing process.

# 5 USER STUDY

To evaluate the usefulness and usability of Stile, we conducted a within-subjects user study with 15 participants with various levels of AI/ML expertise. These participants examined biases in three models—Word2Vec [48], GloVe [52], and BERT [17]. To the best of our knowledge, WordBias [24] is the only interactive system for detecting and exploring biases in word embeddings. Thus, we used WordBias as the comparison baseline. We used the WordBias implementation from its GitHub repository.[1] Since the original implementation of WordBias only supports Word2Vec and GloVe, we extended it to also support BERT. Furthermore, since WordBias has no built-in support for tracing a bias back to the training data, we provided users with the training data in a large spreadsheet for search, filtering, and examination.

## 5.1 Participants

For the user study, we recruited 15 students (5 female, 10 male) from the Department of Computer Science at Purdue University using the department graduate mailing list. Regarding their AI/ML experience, 2 participants had less than 1 year of experience, 9 had 2-5 years, and 4 had more than 5 years. Specifically, 9 of 15 participants had used word embedding models before. 4 participants had research experience in AI fairness, 10 participants had some basic knowledge about AI fairness, and 1 participant said she had only heard about it. As compensation for their participation, each participant received a $25 Amazon gift card.

## 5.2 Models

Stile is designed for both context-free word embeddings and contextualized word embeddings (i.e., language models). For context-free word embeddings, we selected Word2Vec [48] and GloVe [52], since they are commonly used models in NLP applications. We

[1] https://github.com/bhavyaghai/WordBias

trained both Word2Vec and GloVe with a Kaggle dataset [37] of 50,000 news articles from CNN, NY Times, Breitbart, Business Insider, etc. The vector dimension is 300 for each of the embedding models. For context-aware embeddings, we used BERT [17] since it is a popular language model used in various NLP applications. We used the pre-trained bert-base-uncased model from HuggingFace [22]. The BERT word embeddings have 768 dimensions.

## 5.3 Protocol

Each user study started with consent collection and an introduction to AI fairness. During the study, each participant used one tool to identify biases in one of the three models and used another tool to identify biases in another model. To mitigate the learning effect, both the model assignment order and the tool assignment order are counterbalanced across participants. Overall, each model was examined by four participants using each tool. Before each task, the participants first watched a tutorial video about how to use the assigned tool. Then, they were given 5 minutes to get familiar with the assigned tool. After this step, they were given the following task description to identify and verify biases in the assigned model using the assigned tool. During the user study, we did not disclose which tool was developed by our team and which tool was the baseline to ensure a fair comparison.

> Suppose you are a Google developer and you want to use a pre-trained [Word2Vec/GloVe/BERT] model in your product. Your colleague told you that this model has some biases towards certain demographic groups of people. Now you want to figure out what kinds of biases this model exhibit and whether you should use it in your product. In this task, you will use 20 minutes to *identify* and *verify* as many biases as possible using the assigned tool. Every time you find a bias, please verify it with the training data and document it below. Please also explain the reason or the evidence behind each bias you identify.

We put a time limit of 20 minutes for the task and encouraged participants to identify and verify as many biases as possible given this time limit. After completing each task, participants were asked to fill out a post-task survey to give feedback about their experience with the assigned tool. The post-task survey asked participants what they liked or disliked about the assigned tool and what new feature or information they wished to have when performing the task. The survey also included a set of Likert-scale questions to ask users to rate the key features of the tool. After all the tasks were completed, participants filled out a final survey, where they directly compared the two tools. We recorded each user study with the permission of the participants. Each study took about 80 minutes on average.

# 6 USER STUDY RESULTS

This section describes both the quantitative and qualitative results from the within-subjects user study.

## 6.1 User Performance on Bias Detection

All participants used up the given 20 minutes to detect and verify biases. To measure the performance of each participant, we counted how many biases they reported in their responses and how many

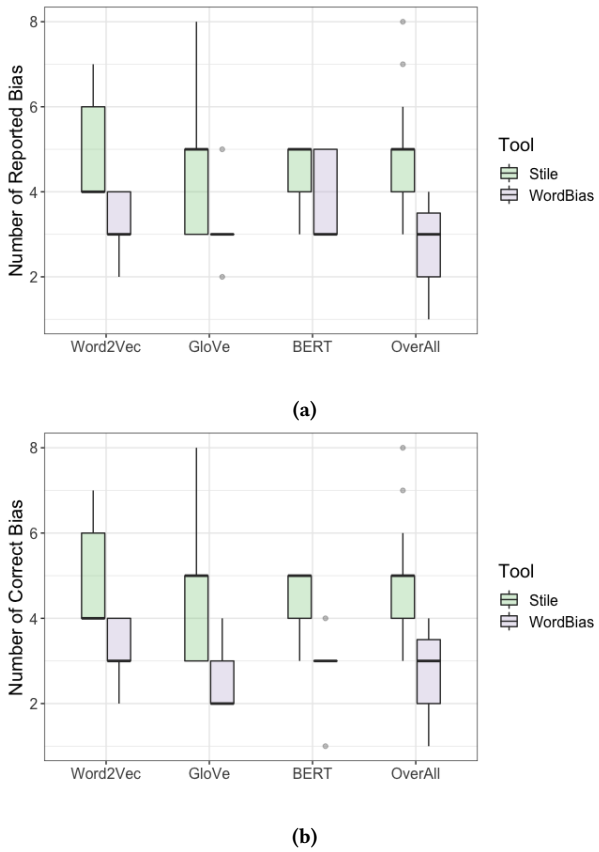



**Figure 7: The number of user-reported biases and the number of correct ones when using Stile and WordBias**

of these biases were real based on the detection by the tool. During the study, we noticed that participants may misidentify a bias for two main reasons. First, some participants reported the stereotypes in their own minds after inspecting a couple of training instances, while these stereotypes were not manifested in the pre-trained text representation. Second, some participants misinterpreted the bias detection results from the assigned tool. For example, the tool identified a word associated with "Male" but the participant reported that the word is associated with "Female". We marked these types of misinterpreted biases as *Incorrect Biases*. We measure the performance of Stile using the following metrics.

***Number of Biases.*** Figure 7(a) shows the number of reported biases per participant across individual models. Overall, within the 20-minute time limit, participants found 4.73 biases ($\sigma = 1.44$) when using Stile and 3.40 biases ($\sigma = 0.98$) when using WordBias. The mean difference of 1.33 bias is statistically significant (Welch's unpaired t-test, $p = 0.0033$).

***Number of Correct Biases.*** Figure 7(b) shows the number of correct biases found by each participant across different models. Overall, all reported biases are correct when using Stile, whereas only 2.87 of 3.40 reported biases are correct when using WordBias. The mean difference of 1.86 is statistically significant (Welch's unpaired t-test,

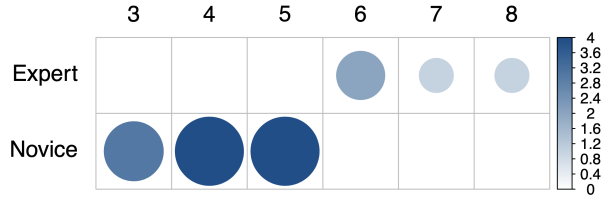



**Figure 8: Correlation between detected biases when using Stile and participants' AI fairness expertise. The numbers in the X-axis indicate the number of detected biases, and the size and color of the circles are determined by the number of participants who detected a certain number of biases.**

$p = 0.0001$). This user performance improvement of Stile over WordBias is consistent for all three models.

Among 15 participants, 13 participants performed better using Stile, while 2 participants performed better using WordBias. We analyzed whether participants' backgrounds played a role here. We observed that all participants who did well using Stile had more than 2 years of ML experience, while the two participants who did not perform well using Stile had less than 2 years of experience in ML. Furthermore, we measured whether participants' knowledge of AI fairness influenced their performance. We categorized the 4 participants who had research experience in AI fairness as *experts*, while the other participants as *novices*. Pearson's Chi-square test shows that there is a statistically significant effect of AI fairness expertise level on the number of detected biases when using Stile ($X^2 = 15$, $df = 5$, $p = 0.0103$). The correlation diagram (Figure 8) shows that all expert participants found 6 or 7 biases, while all novice participants found 5 or fewer biases.

In the post-task survey, participants who performed well with Stile also found it beneficial to trace back to the training data when exploring biases. P2 wrote, "*The visual representation is very clear and at the same time does not feel cluttered. The functionality of showing the samples that it determined each bias from is very helpful.*" Four participants (P5, P7, P8, P9) also mentioned that the filtering and highlighting features in the instance view help them get a better understanding of the identified biases. Participants who did well with Stile also appreciated that the chord diagram presents a large amount of bias information. P10 wrote, "*This tool is fantastic in the sense that it has an instance view showing instances associated with a topic and bias. This saves a LOT of time when you want to verify the bias. The graph, once you get used to it, is also very nice and gives you plenty of information, while being highly interactive.*"

## 6.2 User Performance on Bias Understanding

Apart from detecting more biases, Stile users also showed a deeper insight into the source of a bias in their responses. We measured the **length of user descriptions of biases** to evaluate user understanding and insights. On average, Stile users used 47.6 words to describe one bias and the source of that bias. In contrast, WordBias users used 29.5 words to describe a bias. The mean difference of 18.1 words is statistically significant (Welch's t-test: p-value<0.01). This shows Stile's performance in not only identifying more biases but also conveying more information about the bias. For instance, P4 wrote, "*There is a bias associated with the profession 'soldier' and the gender 'male' as well as the race 'white.' In most of the articles associated with the occupation 'soldier,' masculine pronouns are used to discuss soldiers and their attributes. Most articles are centered around western white-dominant countries' military activities and efforts as well, hence the association with the white race.*"

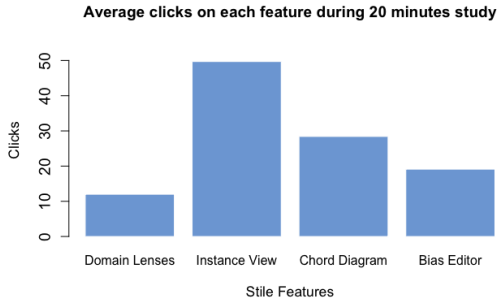



**Figure 9: Average clicks of participants on each feature of Stile during the study**

By tracing back to relevant texts that exhibit a bias in the training data, Stile also helps users to discern some false alarms detected by the bias detection algorithm. P9 wrote, *"There is a bias between white and males... given results do not exactly show bias in the text. For example, there is an article on dogs that also describes them as strong, masculine, and having white color on their face. This is no indication of the relationship between white and male groups of humans*". Indeed, since the relative norm difference algorithm in Stile only computes the association score between a single word and a bias type, it does not consider the surrounding context of a word. Therefore, false alarms may occur. Stile addresses this limitation by allowing users to inspect relevant text in the training data.

When using WordBias, most users struggled to find the source of biases directly from the training data. Furthermore, it was hard for them to know what exactly to look for. P1 mentioned, "*It is difficult to search for a particular instance of bias in the raw data. Regular text search gives hundreds of matches. But they don't point to the particular instance where it happened.*" Some participants also mentioned that they used their personal knowledge and intuition to look for evidence since they had no guidance. P12 wrote, "*I look for a word I know something about (scientist, novice) ..., then look at what biases the AI has. Then I look for articles in the news data file. ... the articles often include people's names, so I could use my own bias to determine race, e.g., Trump = white.*"

## 6.3 Utility Rate and User Ratings

We collected the event log while participants used Stile during the study. The event log captured the number of times each of the main 4 features was clicked. Figure 9 shows the average clicks by a user on each of the features within the 20-minute time limit. On average, a user clicked the Domain Lens feature 12 times, the Instance View feature 49.7 times, the Chord Diagram feature 28.42 times, and the Bias Editor feature 19.13 times. Our analysis also reveals a sequence of back-and-forth clicks between the Instance view and the Chord Diagram feature, indicating a significant amount of engagement involving these two features.

Furthermore, in the post-task survey, participants rated the key features of Stile in 7-point Likert scale questions. Figure 10 shows the distribution of individual feature ratings. The Instance View feature was the most popular choice among users, while the Chord Diagram and Domain Lens were the second most popular choices. P7 wrote, "*The visualization, filtering, and coloring texts are very convenient to look up.*" P3 wrote, "*The visualization on the Chord Diagram was very helpful and showed up very fast. It helps figure out biases very fast.*" P6 said, "*This [domain lens] can be really helpful when the user thinks there might be some bias with a specific word that is not present in the tool.*"

The Strip Plot was the least favorite feature among all features. After manually analyzing the screen recordings, we found that this feature was not frequently used. Some users preferred to have a feature for individual terms or words instead of the strip plot. P9 wrote, "*It doesn't capture the essence of the word.*" P5 suggested, "*Providing less information at times by making more features (like strip chart) optional may help*".

## 6.4 Bias Quality

To understand how useful and novel the identified biases are, the first author manually labeled the correct user answers with one label from each of the following sets—*{Harmful, Neutral}*, and *{Anticipated, Surprising}*. The two categories represent usefulness and novelty respectively. For example, a bias between "terrorist" and "black", "Muslim", is considered to be *Harmful*. On the other hand, a bias between "mother" and "female" is considered to be *Neutral*. For *{Anticipated, Surprising}*, a bias is labeled as *Anticipated* if it aligns with known and existing stereotypes in society, and as *Surprising* otherwise. For example, a bias between "nurse" and "female" is *Anticipated* but a bias between "painter" and "old" is *Surprising*. We found that among the correct biases that participants reported, 65.57% of the biases are *Harmful*, 66.65% are *Surprising*, 36.06% are both *Harmful* and *Surprising*, and 29.51% are *Harmful* but *Anticipated*. This suggests that users were able to find a significant amount of biases that they were not aware of and were harmful.

## 6.5 User Preference and Feedback

9 out of 15 participants reported that they preferred to use Stile over WordBias to identify biases in pre-trained text representations. Figure 11 shows the distribution of user preferences. To understand user preferences for Stile, we coded participants' responses to the questions about what they liked and disliked about Stile and what other features they wished to have. We identified three reasons for the preference towards Stile. First, participants liked the interactivity and responsiveness of Stile. P10 wrote, "*Overall Stile is more interactive and much more informative. I would pick Stile all day because the interface is that much better.*" Second, participants liked the capability of tracing a detected bias back to the training data. Third, users found the visualization to be visually appealing and easy to manipulate. P8 wrote, "*Visually Appealing... I can enable and disable biases to remove clutter.*"

The 6 participants who preferred WordBias over Stile mentioned information overload and overwhelmingness of the chord diagram as the main reasons. P5 wrote, "*Stile counters the pain points of WordBias but also results in information overload in the process.*" Among them, 3 participants actually detected more biases when using Stile than when using WordBias.

# 7 EXPERT REVIEW

To further understand the utility and potential of Stile, we conducted semi-structured interviews with six experts who have expertise in AI/ML fairness and Visualization. To recruit the experts, we emailed authors of several related papers to this work. Five experts are faculty and postdoctoral researchers in R1 universities in the U.S. and one expert is a researcher in the industry. All six experts have at least one peer-reviewed research paper in top-tier conferences on the topic of AI/ML fairness and Visualization.

## 7.1 Protocol

Before starting the expert review, we asked for permission to record the audio of the interview. At the beginning of the expert review, each expert was briefed about the background, motivation, problem definition, and existing solutions. After the brief background, we demonstrated Stile with a walkthrough video and live demonstration of all the features and use cases described in Section 4. The experts were encouraged to give their feedback and point out any limitations, confusion, or suggestions during the demonstration.

After the demonstration, we asked experts to assess Stile in terms of utility, usability, limitations, and improvement opportunities. Specifically, we first asked experts what features they liked or did not like about Stile. Then, we asked experts about their opinion on how effective Stile is in identifying and understanding the source of biases, how effective the tool is in the usage scenario, and what other real-world use cases they find Stile helpful with. We also asked experts how easy, difficult, or overwhelming they think the tool is. Finally, we asked them to point out the limitations of Stile and suggest future improvement opportunities. Each interview session took about 30-40 minutes.

The first author transcribed the audio recording of the interviews and conducted deductive thematic analysis [8, 29] to identify common feedback from the experts. The first author first labeled the transcripts with the open coding methodology [32] and grouped the insightful comments to the following themes—features they like, features they do not like, effectiveness and usefulness, ease of use, and suggestions for further improvement.

## 7.2 Results

Our thematic analysis reveals six features the experts liked and found intuitive and useful. Five out of six experts (E1, E2, E3, E4, E6) liked the ability to go back and forth between identified biases and actual sources in the training data. E2 mentioned, "*So I like the fact that how you can go from having, you know, the keywords and the visualizations to the actual documents that may have been used to train them. So for a common dataset where you just have tons and tons of web content, you know, finding individual examples that contribute to how a model reads things can be important...it may even show you how you can go back to your corpus and figure out maybe something that you shouldn't include.*" Four out of six experts (E1, E3, E5, E6) mentioned they liked that Stile lets users inspect and modify the sensitive attributes. E5 mentioned, "*I liked that sensitive groups like gender and religion can be freely modified.*" Four out of

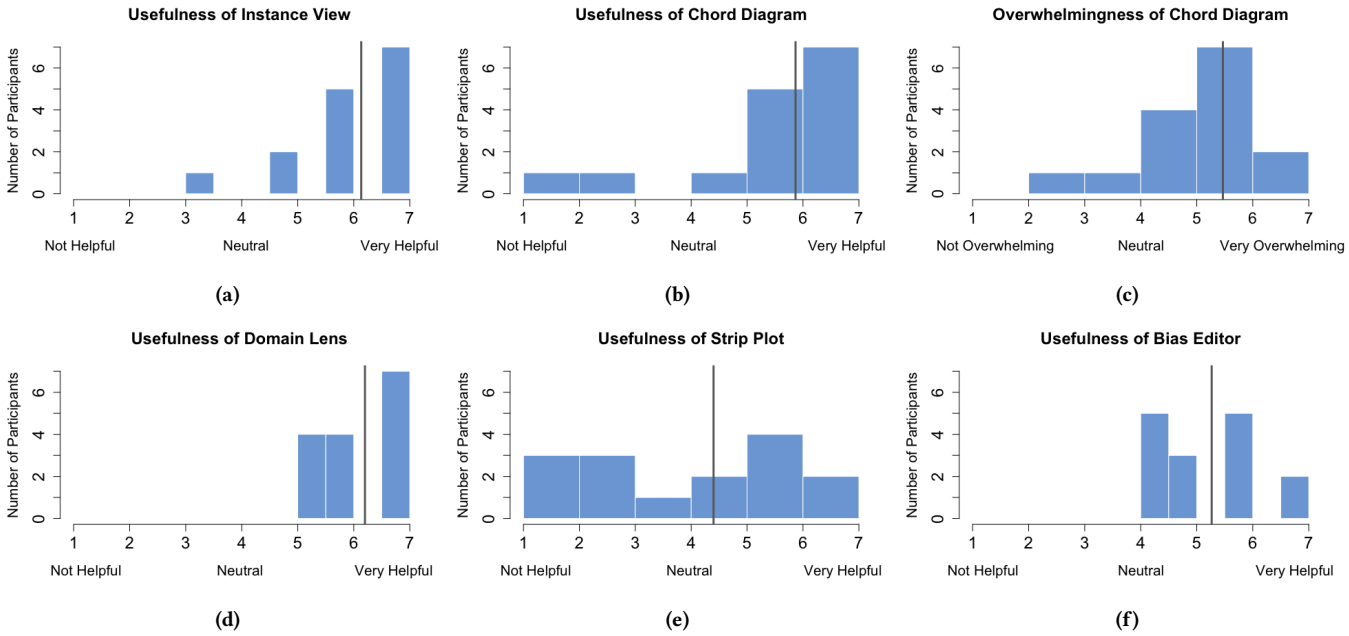



**Figure 10: Individual feature ratings each key feature in Stile (The gray lines indicate the means).**

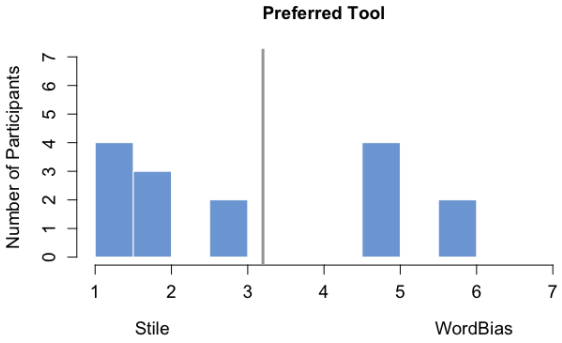



**Figure 11: User Preferences for using Stile or WordBias**

six experts (E1, E2, E5, E4) mentioned model comparison, and three out of six experts(E2, E3, E4) mentioned ranking and comparing biases to be useful features. E1, E4, and E6 mentioned the design and usage of topic words to be very helpful. E6 mentioned, "*You try to write down a couple of words, you actually have this capability of suggesting similar words, which is actually a very nice feature.*" E4 also stated, "*on the left panel you can select which word you want to view. So I think that gives users a lot of flexibility which I like pretty much a lot.*" E1 and E2 mentioned that Stile has a "*very focused workflow*" and "*the visual dashboard gives a lot of useful information that is intuitive enough and doesn't crowd the dashboard.*"

When answering our question about how useful and effective Stile is in identifying biases, all six experts found Stile to be very useful, effective, and sufficient in identifying, understanding, and comparing biases. E1 commented, "*Very effective due to visual summary that lets you look at the whole patterns, also lets you how the biases are inherited from the text.*" E2 mentioned, "*It's very effective and useful, it's nice to compare between biases and also compare between models, good potential for ranking biases.*" E5 stated, "*I definitely believe it can show something that is hiding in the data.*"

We also asked experts, apart from the demonstrated use cases, what other real-world use cases they can think of where Stile will be useful. The experts mentioned **four new use cases** where the experts thought Stile would be useful. First, E1, E2, E3, and E4 thought Stile can be an effective aid in informing the data collection process. E1 mentioned, "*organizations that collect data can use stile to do initial checking to kind of flag the potential biases and their sources*", and E4 mentioned "*use this as a checkpoint to validate cleaned data.*" E2 stated, "*you might find like police reports from Florida in your training corpora that mentions a lot of like race explicitly or something that shouldn't be in there, stile can be helpful for cases like this.*" Second, E2 expressed Stile can be an interesting and useful tool in "Digital Humanities" [20]. E2 mentioned Stile can be easily used in theater or by writers to compare scripts of different genres such as drama, comedy, romance, etc. This can help in informed decisions such as which creative aspect to put more focus on, how to define tropes, etc. Third, E5 believed Stile can serve as an educational tool. E5 mentioned, "*If someone has heard something about the bias in large language models, I think it can serve something as an educational tool for that. Because you also show them, like the original text to provide the context with understanding.*" Lastly, E4 mentioned that a lot of companies have cleaning or data-devising tools automated in their ML pipeline. However, for more sophisticated data, such as data with privacy

concerns, can be checked for bias using tools such as Stile without integrating it into the pipeline and compromising privacy.

Regarding utility, all six experts found Stile to be easy to use, straightforward, intuitive, and not overwhelming. However, E3 and E5 mentioned that the chord diagram can be hard to interpret at first and can seem overwhelming to beginner users. We asked these two experts to share their suggestions on how to reduce the initial overwhelmingness of the chord diagram. E5 suggested that preparing detailed onboarding tutorials for Stile can help reduce the initial information load. She stated, "*Some onboarding like walks through tutorials, to help people understand how to read the visualization can help.*" E3 suggested more strategic use of color and highlighting, such as gradient to represent individual bias scores, more neutral color for all biases, and only adding color when hovering over, these can potentially reduce limitations to interpretation. Moreover, E4 and E6 mentioned that the chord diagram seems hard to interpret initially but becomes clear once we explain the functionalities. E6 mentioned, "*After your explanation, I think I kind of get what the visualization means. But initially, when I just watched the video, I didn't quite get what the visualization really means.*"

Regarding the limitations and less likable features of Stile, experts pointed out four main limitations. Four out of six experts stated that the initial interpretation is hard and not interpretable without explanation. E1 and E2 stated that Stile is very open-ended as it lets users define new sensitive attributes which "*can inject users' own biases*" into the system. E3 stated that Stile requires manual validation after identifying biases, which can become difficult at times. He mentioned, "*Previous work shows relative bias like man is to doctor and women is to nurse, but since we are using bias score, it will require validating how real the bias is.*" And lastly, E6 mentioned that the instance view panel could be more organized and more synchronized with the visualization. For example, clicking on a specific word in the chord diagram should automatically filter all text samples related to that word.

For the last question, we asked the experts to suggest future improvement opportunities. E1 stated it would be nice to have some default topic so that users are not the ones solely initiating the interaction with Stile. E2, E5, and E6 mentioned that a side-by-side comparison of models or data can be a nice added feature. E5 also suggested having a feature to download or save the current progress in Stile can be useful for facilitating the comparison tasks. E3 suggested encoding individual bias scores into the visualization for better interpretability. Lastly, E4 stated that having the option to work with phrases or sub-phrases (e.g., a man from Asia, doctors are always, etc. ) instead of just topic words can be more intuitive.

# 8 DISCUSSION

## 8.1 Design Implications

The user study results suggest that interaction support for bias exploration and debugging can significantly improve the performance of users. In the past, most research effort has been put into the algorithmic part of bias detection. Little attention has been paid to designing interactive support for bias detection. Our work confirms that interaction support is necessary for users to explore, debug, and make sense of biases detected by those algorithms. Therefore, we should treat bias detection systems and users as a team and focus on improving their collaborative performance.

Stile contributes to this effort by providing (1) a domain-based data exploration mechanism, (2) a bias overview visualization for fast and effective bias identification, (3) explaining the source of each identified bias in the training data, and (4) complete transparency and interaction to debug and intervene the bias detection process. Given the same amount of time, participants using Stile identifies more biases, with no incorrect identification. This demonstrates the effectiveness of the interaction design in Stile.

Moreover, our results suggest that the interactive support in Stile also helps users find biases and stereotypes that they were not aware of. In the post-study survey, 10 out of 15 users mentioned Stile helped them identify unknown biases. 12 out of 15 users intervened in the bias detection process by either disabling or modifying existing bias types or by creating new bias types. We believe this type of interaction support will benefit ML developers in identifying and understanding biases within limited time and resources. Results of our expert review also suggest that Stile is an effective aid in identifying and understanding biases and stereotypes in word embeddings and can be deemed useful in comparing models or training corpus as well.

An important lesson we learned from this work is that rendering all detected biases in a compact view such as a chord diagram can be deemed as overwhelming to some users. While we intend to provide an overview of all detected biases for users to explore, participants in our user study had not seen such a visualization before. Therefore, some of them found the chord diagram has a steeper learning curve compared with the parallel coordinates visualization in WordBias, which resembles line charts. The main reason we chose the chord diagram visualization is to render biases related to multiple social or demographic subgroups at scale, which cannot be handled by parallel coordinates. Expert users—those who had research experience in AI fairness—appreciated this visualization and its capability to render intersectional biases in a compact yet highly interactive manner. Our expert review also suggests that expert reviewers do not find the visualization hard to interpret once an explanation is given. However, some experts pointed out how the visualization can seem overwhelming to novice users. One solution to this can be allowing users to switch between different visualizations, e.g., a ranked list and a chord diagram, based on their own needs. Also, as suggested by experts E3 and E5, providing detailed onboarding walkthrough tutorials, or limiting the use of color and highlighting can reduce the overwhelmingness. Furthermore, to make the experience of using our visualization better, we provided the enable-disable option to de-clutter the visualization.

In the meantime, our results suggest that simple visualization like WordBias can lead to incorrect bias identification. We observed a trade-off between correctness and overwhelmingness. Since WordBias visualizes two different subgroups in the same axis, users need to rely on their own cognitive ability to find and compare the relative position of different words in an axis. This confuses users and they unintentionally make mistakes. For example, P1 mentioned that *The word "Dumb" is biased against "Female" and the word "Genius" is biased against "Male".* But in reality, both of the words were associated with "Male" by the tool. The reason for this type of confusion is that both "Male" and "Female" are represented

on the same axis, the upper half for "Female" and the lower half for "Male". So even though the lines from these two words were in the "Male" half of the axis, due to the fact that one was almost towards the middle and one was almost towards the edge, that was easily misinterpreted as being connected to two different halves of the axis. A similar pattern was observed for other incorrect biases. Thus, although WordBias was less overwhelming, it came with the cost of incorrect identification. On the other hand, in Stile, only those subgroups that are associated with a word are highlighted to guide user attention toward relevant biases. So there is no room for such mistakes, which is also backed up by our results.

## 8.2 Identifying and Debugging Biases

As presented in the result section, users were satisfied with the key features of our tool. The main three features that were most liked by users were the "Domain Lens", "Chord Diagram", and "Instance View". As these three features are the major contribution of this work, we believe this tool has a lot of potential in aiding ML developers and practitioners to find and understand biases with limited time and resources. Users mentioned about Stile that the the visualization, "*was very helpful*", "*helps figure out biases very fast*", "*easy to use*", "*visually appealing*".

The usability of the features in Stile is also backed up by our event-log analysis. In the short span of 20 minutes, clicks on each feature, mainly Instance View and Chord Diagram were very high. A combined 80 clicks on these two features in 20 minutes means on average there were at least 4 clicks on these two features per minute. This is a very promising result for Stile in terms of usability. The relatively high amount of clicks on the other two features indicates that users were largely engaged with the aforementioned features, where occasionally using other features as well. The user engagement and pattern of usage confirm that users not only liked the features but also found them useful.

As shown by a previous study [35], ML developers and practitioners have limited time and resources to invest in identifying biases in a model and its data. Hence our tool seems very promising as it presents a fast, effective, and highly interactive visualization to identify biases. Furthermore, our explanation of biases by showing related instances and highlighting topic words and demographic words in the instances was liked by all the users irrespective of their performance in the user study. As mentioned by P5, the instance view "*automates the process of bias search to a large degree*". Users felt that Stile "*provides a good interface to study and analyze biases*". Also, our result shows that, without any aid for understanding the source of biases from the data, users often rely on their own biases to understand why a model develops such a bias. When looking for evidence, depending on such personal biases can lead to consideration of attributes that are not considered by the model when detecting biases. For example, a user might think certain names have contributed to violent activities being associated with a certain race. But in reality, the model might have no association with those names regarding that bias. This can cause major misconceptions about the data, model, and bias detection process in general. Stile provides a clear and transparent way of explaining the source of biases to users, which in turn helps them to get a better understanding of the model and the bias detection process.

Findings from the expert review also confirm this observation as all experts found Stile to be useful in identifying, understanding, and comparing biases. The experts in our expert review appreciated the feature of tracing back the biases in the training corpus from the visualization and the flexibility to explore different domains. They believe Stile can effectively aid in the data cleaning or de-biasing process since it gives a deeper understanding of the training corpus and gives the flexibility to compare and contrast between biases, models, and training corpus.

## 8.3 Target User Groups and Use Cases

Our result shows that users who have more expertise in AI/ML and fairness significantly outperformed other users both in terms of performance and data insights. As our intended users for Stile are ML model and application developers who wish to have interaction support for bias exploring and debugging, the result is promising.

Our expert review also highlighted some innovative use cases such as digital humanities or education settings where Stile can be applicable. Although Stile is not designed for end-users with limited ML expertise, user performance results show that users with basic knowledge of ML can also benefit from using Stile. Many of our novice users performed better using Stile than using WordBias, finding more biases and developing good data insights. We believe that Stile can be extended to further support non-expert users. Some extensions include providing an ample amount of pre-defined topic words and bias types covering different areas of interest. Since the Chord Diagram visualization is deemed complex for some users who are not familiar with data visualization, we need to simplify the Chord Diagram visualization to reduce the requirement for graphic literacy for end-users.

We also believe Stile can be potentially used for AI fairness education. In this era of chatbots and large language models, tools such as Stile can be utilized to raise awareness about social risks associated with these models and in turn, raise AI fairness in AI end-users. One such use case of Stile for AI fairness education can be analyzing texts generated by well-known language models by plugging in their text representation to Stile. Educators can directly demonstrate the stereotypes that exist in excerpts of generated texts and discuss the social risks associated with them. Moreover, as pointed out by expert E5, one challenge with growing LLM popularity is educating end-users on the potential risks of AI. A lightweight tool such as STILE takes away a fair share of the workload by allowing users to freely explore potential biases in LLMs. By tracing back to some training examples, users can quickly understand how a bias is exhibited in the original data. This makes users aware that AI-generated content is not free from biases, nudging users to more responsive use of LLMs. Additionally, Stile can be integrated into the pipeline of popular state-of-the-art LLMs such as ChatGPT. Collecting synthetic datasets (e.g., news, social media content, educational Q&A, etc.) from ChatGPT and using Stile to detect biases in those datasets can help users effortlessly identify biases and stereotypes generated by the LLM.

Furthermore, our expert review also pointed out further use cases including digital humanities, script writing, informed data collection, etc. Therefore, we believe a lightweight tool such as Stile has the potential to be integrated into several other systems to ensure responsible use of AI.

### 8.4 Limitations and Future Work

In addition to the limitation of information overload, it is imperative to deliberate upon some other limitations and challenges we faced.

In this work, we have only used one bias detection metric, *Relative Norm Difference*. This metric has an inherent limitation—the bias score largely depends on the bias-definitive words used to represent a bias. We designed the "Bias Editor" feature to address this limitation by allowing users to add or delete the words used to define a demographic subgroup. However, one can also consider devising new metrics that do not solely depend on bias definition to fundamentally overcome this limitation. Furthermore, we have considered only cosine similarity for measuring the distance between word vectors while calculating relative norm difference. In the future, it is worthwhile to experiment with other similarity metrics such as Minkowski or Euclidean distance. Nevertheless, since the main goal of this study is to demonstrate the usefulness of visualization and interaction support rather than algorithmic advancement, we selected the highly used distance measure as our method to calculate bias detection metrics.

Currently, Stile only supports bias detection and exploration in English text corpora. In the future, we plan to extend Stile to support bias detection in other languages.

Moreover, the intersectional bias is limited by the physical limitation of a chord having two edges only. Since a chord can connect at most two demographic subgroups, it requires two chords to visualize an intersectional bias between three subgroups. Future work can focus on designing novel visualization that can connect more than two subgroups with a single connection.

Finally, like any explainable AI that suffers from over-trust and under-trust depending on the level of explanations [26, 40], our tool also suffers from the same limitations. When users found irrelevant articles that were used by the model to make some association, their trust in the system started to decrease. For example, P4 wrote, "*The bias between "doctor" and "female" is interesting and new to me. Upon examining some of the articles used to extract this bias, it appears that this bias may not be entirely accurate...For example, an article about a doctor sexually assaulting young women contains numerous feminine pronouns but the doctor is a male. Seeing a handful of such articles makes me wonder whether the model/algorithm used in this tool to extract biases is accurate.*" This kind of false alarm on model biases often affects user trust. Currently, the bias detection algorithm we use does not understand the context of a word or take it into account when detecting bias. This plays a major role in user trust in the system. In future work, we aim to focus on user trust and the required level of explainability for bias debugging tools like Stile.

## 9 CONCLUSION

This paper presents Stile, an interactive system to identify and debug biases and stereotypes in pre-trained text representations. With ML being used for making everyday decisions, it is necessary for ML developers to know the biases in a model and understand the sources of biases in the training data. Stile provides a solution to this by supporting the exploration and debugging of biases in a highly interactive way. Specifically, Stile provides a bird's-eye view of all detected biases using the chord diagram visualization and also allows users to trace back to text documents that exhibit a bias in the training data. Our user study with 15 participants shows that users detected more biases and developed deeper insights when using Stile compared to using WordBias. Our expert review confirms the utility and usability of Stile and reveals promising usage of Stile in real-world scenarios. We believe that our intended users will highly benefit from using this tool and will apply their knowledge in developing fair models and applications.

## REFERENCES


[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[2] Yongsu Ahn and Yu-Ru Lin. 2019. Fairsight: Visual analytics for fairness in decision making. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1086–1095.

[3] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).

[4] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2021. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* 50, 1 (2021), 3–44.

[5] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'It's Reducing a Human Being to a Percentage' Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 Chi conference on human factors in computing systems*. 1–14.

[6] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems* 29 (2016).

[7] Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4758–4781.

[8] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[10] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*. PMLR, 77–91.

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

[12] 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.

[13] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. 2019. SearchLens: Composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 498–509.

[14] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[15] Kate Crawford. 2017. Artificial intelligence with very real biases. *The Wall Street Journal* (2017).

[16] Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihiro Nishi, Nanyun Peng, et al. 2022. On Measures of Biases and Harms in NLP. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*. 246–267.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[18] The Britannica Dictionary. 2022. The Britannica Dictionary. https://www.britannica.com/dictionary.

[19] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems* 32 (2019).

[20] Johanna Drucker. 2016. Intro to digital humanities: introduction. *UCLA Center for Digital Humanities//Center for Digital Humanities* (2016).

[21] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[22] Hugging Face. 2022. Hugging Face – The AI community building the future. https://huggingface.co/.

[23] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences* 115, 16 (2018), E3635–E3644.

[24] Bhavya Ghai, Md Naimul Hoque, and Klaus Mueller. 2021. WordBias: An Interactive Visual Tool for Discovering Intersectional Biases Encoded in Word Embeddings. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[25] Bhavya Ghai and Klaus Mueller. 2022. D-BIAS: A Causality-Based Human-in-the-Loop System for Tackling Algorithmic Bias. *IEEE Transactions on Visualization and Computer Graphics* (2022).

[26] Ella Glikson and Anita Williams Woolley. 2020. Human trust in artificial intelligence: Review of empirical research. *Academy of Management Annals* 14, 2 (2020), 627–660.

[27] Google. 2017. FACETS KNOW YOUR DATA. https://pair-code.github.io/facets/.

[28] Ziwei Gu, Jing Nathan Yan, and Jeffrey M Rzeszotarski. 2021. Understanding User Sensemaking in Machine Learning Fairness Assessment Systems. In *Proceedings of the Web Conference 2021*. 658–668.

[29] Greg Guest, Kathleen M MacQueen, and Emily E Namey. 2011. *Applied thematic analysis*. sage publications.

[30] Wei Guo and Aylin Caliskan. 2021. Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 122–133.

[31] Foad Hamidi, Morgan Klaus Scheuerman, and Stacy M Branham. 2018. Gender recognition or gender reductionism? The social implications of embedded gender recognition systems. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–13.

[32] Beverley Hancock, Elizabeth Ockleford, and Kate Windridge. 2001. *An introduction to qualitative research*. Trent focus group London.

[33] M Hardt, E Price, N Srebro, et al. 2016. Equality of opportunity in supervised learning, in 'Advances in neural information processing systems'. (2016).

[34] Marti Hearst. 2006. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR workshop on faceted search*. Seattle, WA, 1–5.

[35] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–16.

[36] Sophie Jentzsch, Patrick Schramowski, Constantin Rothkopf, and Kristian Kersting. 2019. Semantics derived automatically from language corpora contain human-like moral choices. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 37–44.

[37] Kaggle. 2017. All the news. "https://www.kaggle.com/datasets/snapcrack/all-the-news".

[38] Jae Yeon Kim, Carlos Ortiz, Sarah Nam, Sarah Santiago, and Vivek Datta. 2020. Intersectional Bias in Hate Speech and Abusive Language Datasets. In *Proceedings of the Fourteenth International Conference on Web and Social Media (ICWSM), Data Challenge Workshop*. AAAI Organization.

[39] Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *Advances in neural information processing systems* 34 (2021), 2611–2624.

[40] René F Kizilcec. 2016. How much information? Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2390–2395.

[41] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring Bias in Contextualized Word Representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. 166–172.

[42] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *arXiv preprint arXiv:1703.06856* (2017).

[43] Po-Ming Law, Sana Malik, Fan Du, and Moumita Sinha. 2020. Designing Tools for Semi-Automated Detection of Machine Learning Biases: An Interview Study. *arXiv preprint arXiv:2003.07680* (2020).

[44] Min Kyung Lee. 2018. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society* 5, 1 (2018), 2053951718756684.

[45] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*. PMLR, 6565–6576.

[46] Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. 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)*. 615–621.

[47] Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. 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)*. 622–628.

[48] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[49] Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456* (2020).

[50] Arvind Narayanan. 21. Fairness definitions and their politics. In *Tutorial presented at the Conf. on Fairness, Accountability, and Transparency*.

[51] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. 2020. Bias in word embeddings. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 446–457.

[52] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[53] Amirarsalan Rajabi and Ozlem Ozmen Garibay. 2022. Tabfairgan: Fair tabular data generation with generative adversarial networks. *Machine Learning and Knowledge Extraction* 4, 2 (2022), 488–501.

[54] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[55] Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics* 9 (2021), 1408–1424.

[56] Aaron Springer, Jean Garcia-Gathright, and Henriette Cramer. 2018. Assessing and Addressing Algorithmic Bias-But Before We Get There.... In *AAAI Spring Symposia*.

[57] Yi Chern Tan and L Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. *Advances in Neural Information Processing Systems* 32 (2019).

[58] Qianwen Wang, Zhenhua Xu, Zhutian Chen, Yong Wang, Shixia Liu, and Huamin Qu. 2020. Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1470–1480.

[59] Allison Woodruff, Sarah E Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. 2018. A qualitative exploration of perceptions of algorithmic fairness. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–14.

[60] Tiankai Xie, Yuxin Ma, Jian Kang, Hanghang Tong, and Ross Maciejewski. 2021. Fairrankvis: A visual analytics framework for exploring algorithmic fairness in graph mining models. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 368–377.

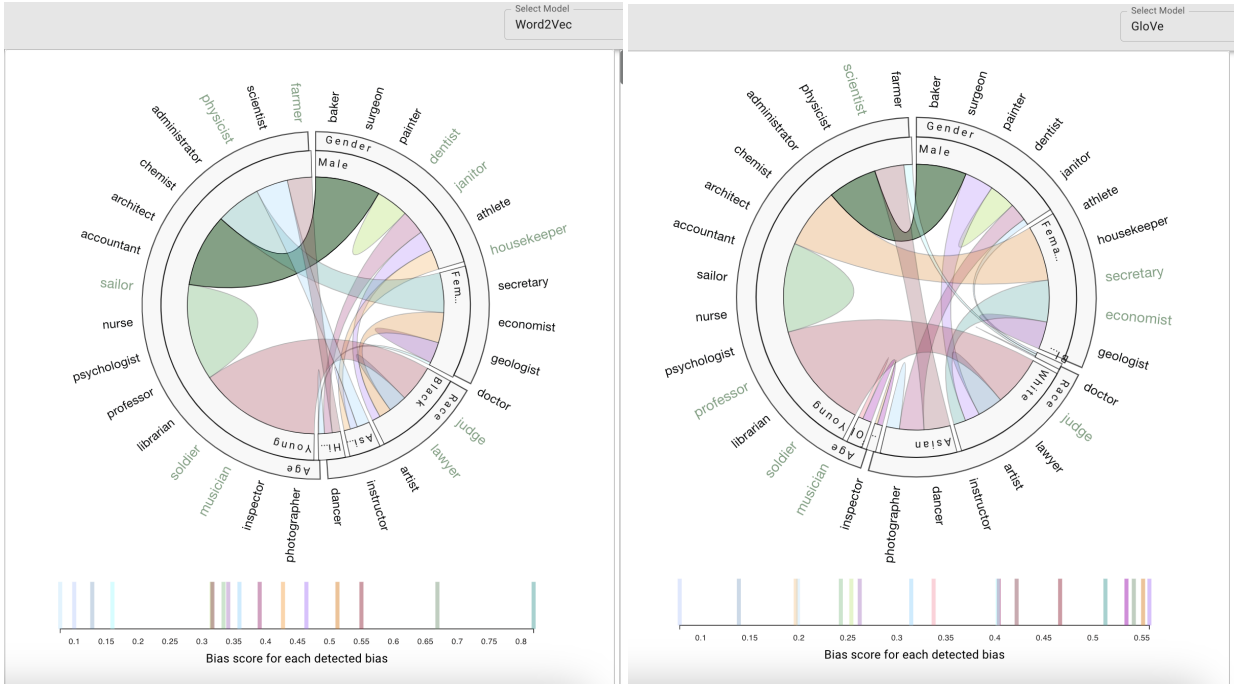
[61] Jing Nathan Yan, Ziwei Gu, Hubert Lin, and Jeffrey M Rzeszotarski. 2020. Silva: Interactively assessing machine learning fairness using causality. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[62] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*. 534–539.

[63] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender Bias in Contextualized Word Embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1.

[64] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876* (2018).

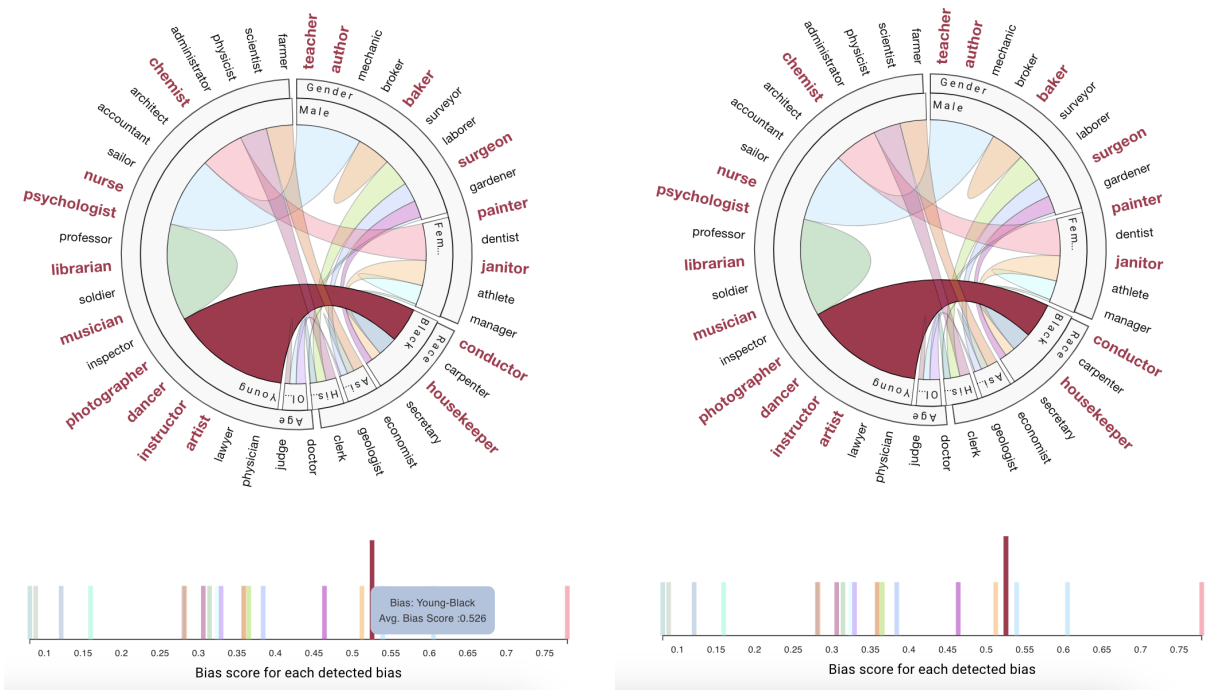
# A EXAMPLE OF USAGE SCENARIO 2




**(a)** **(b)**

**Figure A.1: An example comparison of detected biases for the same set of "Professions" between two word embedding models (a) Word2Vec, b) GloVe. Both models are trained on the same training corpus. For the same set of professions and the same training data, STILE captured 15 intersectional biases for Word2Vec and 18 intersectional biases for GloVe.**

# B EXAMPLE OF INTERACTION BETWEEN STRIP PLOT AND CHORD DIAGRAM




(a) (b)

**Figure B.1: An example of cross interaction between the Strip Plot and the Chord Diagram (a) Hovering over a strip in the Strip plot highlights the corresponding chord and all words associated with that bias in the Chord Diagram, (b) Hovering over a bias or chord inside the Chord Diagram highlights the corresponding bias strip in the strip plot to indicate the severity of that bias.**