Debiasing Vandalism Detection Models at Wikidata

Stefan Heindorf
Paderborn University
heindorf@uni-paderborn.de

Yan Scholten
Paderborn University
yascho@mail.uni-paderborn.de

Gregor Engels
Paderborn University
engels@uni-paderborn.de

Martin Potthast
Leipzig University
martin.potthast@uni-leipzig.de

ABSTRACT

Crowdsourced knowledge bases like Wikidata suffer from low-quality edits and vandalism, employing machine learning-based approaches to detect both kinds of damage. We reveal that state-of-the-art detection approaches discriminate anonymous and new users: benign edits from these users receive much higher vandalism scores than benign edits from older ones, causing newcomers to abandon the project prematurely. We address this problem for the first time by analyzing and measuring the sources of bias, and by developing a new vandalism detection model that avoids them. Our model FAIR-S reduces the bias ratio of the state-of-the-art vandalism detector WDVD from 310.7 to only 11.9 while maintaining high predictive performance at 0.963 ROC_{AUC} and 0.316 PR_{AUC}.

ACM Reference Format:
Stefan Heindorf, Yan Scholten, Gregor Engels, and Martin Potthast. 2019. Debiasing Vandalism Detection Models at Wikidata. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313507

1 INTRODUCTION

Knowledge bases play an important role in modern information systems. For instance, web search engines use them to enrich search results, conversational agents to answer factual questions, and fake news detectors for fact checking. Collecting knowledge at scale still heavily relies on crowdsourcing: Google acquired the open Freebase project to bootstrap its proprietary "Knowledge Graph" until Freebase was shut down and succeeded by Wikidata, the free knowledge base of Wikimedia. Other prominent open knowledge bases like Yago and DBpedia also depend on crowdsourcing by extracting knowledge from Wikipedia. As crowdsourcing knowledge has a long history, so does the fight against damage caused by vandals and other users, which may propagate to information systems using the knowledge base, potentially reaching a wide audience.

From its humble beginnings with manual review and rule-based detection bots, damage control at Wikipedia has grown into an intricate sociotechnical system, where man and machine work together to review edits and to maintain the integrity of its articles. While Wikipedia's damage control system grew alongside the encyclopedia for more than a decade, Wikidata gained momentum at a much faster pace: Transferring Wikipedia's system to Wikidata without a second thought (see Figure 1 for an illustration), adapting its procedures and the machine learning technologies employed [27, 49], has given rise to discrimination.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313507

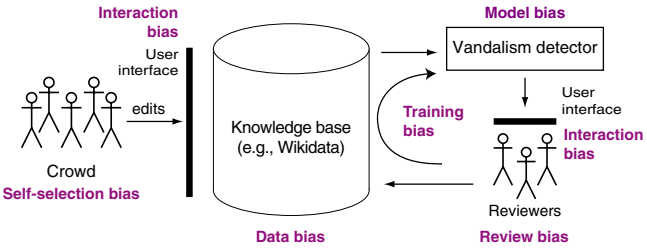


Figure 1: Wikidata's damage control system along potential sources of bias, which may accumulate in a vicious cycle.

In this paper, we reveal for the first time that state-of-the-art vandalism detectors employed at Wikidata are heavily biased against certain groups of contributors. For example, *benign* edits of new users receive vandalism scores over 300 times higher than *benign* edits of older user accounts. Such a widespread discrimination of certain user groups (especially that of anonymous editors) undermines the founding principles on which Wikimedia's projects are built:[1] maintaining a neutral point of view, the ability of anyone to edit articles, and the creation of a welcoming environment. The discrimination of anonymous users by registered users has long been recognized and the problem has been tackled through continuous community outreach.[2] But when discrimination gets encoded into automatic decision-making at Wikimedia, this aggravates the problem. For example, it has been previously found that new contributors whose edits are automatically reverted are much more likely to withdraw from the project [21, 22, 50].

Besides raising awareness, we carefully analyze different sources of bias in Wikidata's damage control system. Based on these insights, we develop two new machine learning models and demonstrate that bias can be significantly reduced compared to the state-of-the-art. Our model FAIR-E uses graph embeddings to check the content's correctness without relying on biased user features. Our model FAIR-S systematically selects the best-performing hand-engineered features under the constraint that no user features are used. Furthermore, we experiment with different transformations of the state-of-the-art vandalism detector WDVD: post-processing scores, reweighting training samples, and combining approaches via ensembles. We evaluate our approaches on a subset of the standardized, large-scale Wikidata Vandalism Detection Corpus 2016 [27], comparing our results to others from the literature.

In what follows, after discussing related work in Section 2, we analyze in Section 3 the sources of bias for Wikidata's damage control system. Section 4 introduces our new detection models designed to mitigate biases. Sections 5 and 6 detail the evaluation data and our comparative evaluation, respectively, and Section 7 discusses limitations and practical implications.

[1] https://meta.wikimedia.org/wiki/Founding_principles
[2] https://en.wikipedia.org/wiki/Wikipedia:IPs_are_human_too

2 RELATED WORK

Despite the long-standing insight that machine learning is prone to incur bias, the body of work addressing this problem is surprisingly small. Only recently Baeza-Yates [2] compiled an overview of biases found on the web that may induce discrimination, including an analysis of Wikipedia's editor elite and its gender gap. Romei and Ruggieri [48], Zliobaite [67], and Pedreschi et al. [45] collect measures to quantify bias in machine learning models which we employ in our analysis. Otherwise, only a handful of papers explicitly attempt to mitigate biases in machine learning applications: Zhao et al. [65] and Bolukbasi et al. [5] tackle gender bias, Wang et al. [57], Wilkie and Azzopardi [59] and Yang and Stoyanovich [61] tackle biases in information retrieval systems, Torralba and Efros [53] investigate bias in image classification, and Dixon et al. [13] propose methods to measure and mitigate bias in text classification. Algorithmic attempts at bias mitigation include naive Bayes [7], SVMs [62], decision trees [30], and random forests [47]. The trade-off between accuracy and fairness is explored by Berk et al. [4], Kleinberg et al. [33], Corbett-Davies et al. [10], and Chouldechova [8], while performance measures for imbalanced datasets such as PR_{AUC} and ROC_{AUC} are not considered. Berk et al. [4], Hardt et al. [23], and Dwork et al. [16] discuss different notions of fairness including equality of opportunity and statistical parity. More closely related to our work, Halfaker et al. [21, 22] and Schneider et al. [50] find that newcomer retention at Wikimedia projects is severely affected by overzealous reversion of their edits. Passing part of the blame to automatic vandalism detectors, no remedies are proposed.

Vandalism at Wikipedia is defined as "editing deliberately intended to obstruct or defeat the project's purpose."[3] Though this definition excludes damage caused unintentionally, the task of a vandalism detector, human or other, is to identify both kinds of damage so they can be dealt with according to severity. The literature and practitioners often speak of vandalism detection while actually operationalizing damage detection in general [31]; current vandalism detectors are quite incapable of discerning user intent, which often even challenges human reviewers. Vandalism detection was first proposed for Wikipedia, employing text mining and user reputation features to detect damaging edits to articles [46]. More than a dozen approaches have been proposed since (e.g., [1, 29, 54, 56]), giving rise to a rich set of features, many of which have been transferred into the models currently at work. By comparison, vandalism detection models for Wikidata are still in their infancy [25, 27, 43, 49, 52], as evidenced by the facts that their feature sets are almost entirely the same as those employed for Wikipedia, and that hardly any feature quantifies Wikidata's actual content, i.e., the knowledge encoded as subject-predicate-object triples. Our approach omits the biased user reputation features and introduces novel graph embeddings, drawing from, and advancing related work on fact checking and link prediction.

Fact checking pertains to checking the correctness of a fact and can be divided into approaches using internal and external knowledge. Regarding the former, Ciampaglia et al. [9] compute the shortest path between a fact's subject and its object, considering different weighted (transitive) predicates. Shi and Weninger [51] determine paths between subject and object that help to distinguish correct from incorrect predicates between them, employing a one-hot encoding and logistic regression. We experimented with path features, too, but found a simpler graph embedding to be superior for debiasing. Nishioka and Scherp [43] analyze the evolution of knowledge graphs to verify changes, employing basic features like the age of entities and predicates and the in- and out-degree of entities. Approaches using external knowledge include that of Wu et al. [60], who check the correctness of facts from news articles by generating SQL queries to find contradictory information in a relational database. Lehmann et al. [37] check the correctness of a given subject-predicate-object triple via web search, taking into account the trustworthiness of websites found.

Link prediction is the task of predicting missing predicates between pairs of subjects and objects. Nickel et al. [41] survey corresponding approaches and distinguish explicit and latent features. The former category includes path rankings used to complete existing knowledge bases [36, 40, 51] and validation knowledge extracted from web sources [14, 15, 28]. Gardner and Mitchell [20] simplified and sped up path ranking. We experimented with their best features, but found a simpler graph embedding to be superior for debiasing. Other approaches do not try to complete a knowledge base but classify the local completeness of entities within [11, 18]. Link prediction approaches with latent features are based on matrix factorization [42], neural embeddings [14], and translational embeddings [6, 38, 55, 58]. Special kinds of link prediction include type prediction of the type of an entity [19, 39, 44] and predicting obligatory attributes of entities of a given type [35].

Altogether, fact checking and link prediction are complementary in that the former's goal is to identify incorrect knowledge, whereas the latter's goal is to introduce correct knowledge that is missing. Both are often evaluated on artificial data, whereas we apply them to vandalism detection at scale on real-world data from Wikidata. Moreover, both tasks typically assume a static knowledge base, while vandalism detection presumes a constant stream of edits. We adapt the methods borrowed from both tasks accordingly.

3 BIAS ANALYSIS

Wiktionary defines 'discrimination' as "*treatment* of an individual or group to their disadvantage" and 'bias' as "*inclination* towards something; predisposition, partiality, prejudice [...]." If a given system acts discriminatorily, we call it biased, and we ask which of its components cause its bias and how it can be controlled to render its behavior non-discriminatory. In this section, we carry out the first bias analysis of Wikidata's damage control system.

Several potential sources of bias can be identified (see Figure 1): *model bias* refers to the vandalism detection model used, and may result from its selection of features and its learning algorithm. The user interface of monitoring tools may introduce *interaction bias* by directing reviewer attention to high-scoring edits first, enclosing them in a "filter bubble" and reinforcing inherent *reviewer bias*, i.e., the prejudices that humans sometimes hold. Analyzing the reviewers' decisions, models are adjusted directly, by creating rules and features, or indirectly, by using their decisions as labeled training data, causing *training bias* and *data bias*. Furthermore, biased reviewer decisions may lead to *self-selection bias* among Wikidata's volunteers, ousting non-conformists.

[3] https://en.wikipedia.org/wiki/Wikipedia:Vandalism

The discriminatory nature of *Wikipedia's* damage control system has been previously shown [21, 22, 50]: the rise of (semi-)automatic reviewing tools caused more newcomer contributions to be considered damaging, severely affecting retention. Although policies have been adjusted to prevent such discrimination, the vandalism detection models have not been redesigned. Our analysis shows that, at least for *Wikidata*, this is insufficient. In what follows, we go beyond previous analyses by shedding light on the main sources of bias, the detection models and the human reviewers, and by taking into account anonymous users, the antecedents of newcomers.

3.1 Measuring Bias

Every edit of an item at Wikidata results in a new revision i of that item, and the task of a vandalism detector c is to compute a vandalism score for each new revision as soon as it arrives, so that $c(i) \approx Pr(i = \text{vandalism} \mid x_i)$, where x_i is i's feature vector. The scores are then used for two modes of operation. In fully automatic mode, the revisions exceeding a score threshold are automatically reverted without any human intervention. In semi-automatic mode, revisions are ranked by vandalism score and manually reviewed.

To measure the bias of a detector producing continuous scores, roughly following Kleinberg et al. [33] and Zemel et al. [63], we compare the average scores of *benign* edits by two disjoint user groups. The more the difference deviates from 0 or the ratio deviates from 1, the more biased the classifier is. By convention, we call one group "protected" (i.e., to be protected from discrimination). Let I denote the set of all revisions, then given the ground truth whether a revision i is vandalism or benign, we get the following contingency table, where $A, \ldots, H$ denote the corresponding subsets of I, from which we derive the difference (Diff.) and the Ratio between the protected group and the remainder as bias measures:

		Truth benign	Truth vand.	⋃
Protected	yes	A	B	C
	no	D	E	F
⋃		G	H	I

$$\text{Diff.} = \frac{1}{|A|}\sum_{i\in A} c(i) - \frac{1}{|D|}\sum_{i\in D} c(i)$$

$$\text{Ratio} = \frac{1}{|A|}\sum_{i\in A} c(i) \;/\; \frac{1}{|D|}\sum_{i\in D} c(i)$$

Table 1a shows examples, comparing the vandalism score of the Wikidata Vandalism Detector (WDVD) for the creation of a given subject-predicate-object triple by a registered user with that of creating the same triple anonymously. Regarding the first example, at a score difference of 0.0907, an anonymous user receives a score that is 900.2 times higher than that of a registered user.

To measure the bias of semi-automatic detection, which produces binary scores, we utilize the odds ratio, which is invariant to changes in class distribution in the dataset. Let $p_1 = \sum_{i\in A} c(i)/|A|$ be the proportion of benign revisions from protected users that are considered vandalism and let $p_2 = \sum_{i\in D} c(i)/|D|$ be the proportion of benign revisions from other users that are considered vandalism:

$$\text{Odds ratio} = \frac{p_1/(1-p_1)}{p_2/(1-p_2)}.$$

Further measures have been proposed, including the closely related risk difference and risk ratio [7, 48, 67], but also measures based on individuals instead of groups [16], the latter demanding that similar edits receive similar scores. For lack of a task-specific similarity measure, we resort to the aforementioned measures.

3.2 Biases of Vandalism Models

There are two kinds of vandalism detectors for Wikidata: (1) the rule-based Wikidata Abuse Filter (FILTER) [27], which tags revisions as per user-defined rules,[1] and (2) machine learning-based detectors. The latter include the state-of-the-art approach WDVD [27] and variants thereof developed during the WSDM Cup 2017 [24], the "Objective Revision Evaluation Service" (ORES) [49] deployed at Wikimedia, and our new approach FAIR proposed in this paper.[2] The WDVD approach outperformed all other approaches in the recent WSDM Cup 2017 in terms of PR_{AUC} [25]. Table 1b shows their distribution of features across feature categories. Apparently, all previous approaches rely on contextual, and especially user-related features, such as account age and whether an edit was made by a registered or an anonymous user. Moreover, hardly any content features capture Wikidata's statements, but only textual portions of an item. This already hints at a high potential for bias.

Analyzing the vandalism scores of all approaches applied on the Wikidata Vandalism Corpus 2016 [25] reveals that all except FAIR exhibit significant biases. Table 1c exemplifies the average vandalism scores obtained for anonymous and registered users from which we compute the bias measures found in Table 1d (top row pair). Similarly, the bias against newcomers and country of origin is computed (remainder of the table). Being anonymous or a newbie raises one's average vandalism score of benign edits by a factor of up to 311 under WDVD, and by a factor of 40-133 under ORES and FILTER. With FAIR, we reduce the bias ratio to 11.9. There is comparably small bias against country of origin, and even a bias in favor of Japanese users, which can be explained by their low vandalism prevalence [31]. These values must be taken with a grain of salt, since geolocation is available only for anonymous users, who are a priori discriminated because of this fact. In what follows, we hence focus on mitigating the bias against anonymous users.

Upon close inspection, the many user-related features employed by WDVD, ORES, and FILTER turned out to be the causes for their biases. For example, the feature `isRegisteredUser`, that is employed by both WDVD and ORES, is a simple feature with high predictive performance. But it causes *benign* edits by anonymous users to have much higher scores than *benign* edits by registered users: 9.00% of edits by anonymous users constitute vandalism and only 0.03% of edits by registered users (in the training and validation set of WDVC-2016-Links). The feature is not able to discriminate *benign* and *vandalizing* edits based on the actual content of an edit, thus assigning roughly 300 times higher scores to *all* edits by anonymous users. Similarly, ORES includes the age of a user (`userAge`), and WDVD includes the numbers of revisions and items edited by a user (`userFrequency`, `cumUserUniqueItems`), as well as geolocation of IP addresses (e.g., `userCountry`, `userCity`). Regarding FILTER's rules, Table 1e lists the top-most rules fired which exploit user information: "new user changing sth."; "new user removing sth."; and "possible vandalism." Compared to the top ones not doing so, these rules act highly biased against anonymous users. Edits to which these rules apply receive an average vandalism score of 28.8%, which is a high number given the large class imbalance; the majority of wrong decisions affect anonymous users (69.1%), again

[1] We convert tags to continuous scores by computing the empirical vandalism probability of all past revisions with a given tag.

[2] For brevity, we report only on the best variant, FAIR-S, in this bias analysis.

Table 1: Bias analyis: (a) Examples for model bias. (b) Overview of features by model. (c) Average vandalism scores for anonymous and registered users.[4] (d) Bias measurements against protected users as score difference (top rows) and score ratio (bottom rows). (e) Rules of the Wikidata Abuse Filter (FILTER) and their biases.[5] (f) Biases by Wikidata reviewers. Among 1,100 benign edits, 145 were incorrectly reverted by human reviewers with disproportionately many affecting anonymous editors.

(a)

Revision	User	Score	Diff.	Ratio
⟨Guido Westerwelle, place of death, Cologne⟩				
313453592	Anonymous	0.0908	0.0907	900.2
313455460	Registered	0.0001		
⟨Alejandro Cuello, occupation, actor⟩				
325717121	Anonymous	0.2912	0.2763	19.6
318143388	Registered	0.0149		

(b)

Feature	WDVD	ORES	FILTER	FAIR-S
Content	27	5	0	4
Character	11	0	0	0
Word	9	3	0	0
Sentence	4	1	0	0
Statement	3	1	0	4
Context	20	8	1	10
User	10½	2	½	0
Item	2	2	0	6
Revision	7½	4	½	4

(c)

Users	WDVD	ORES	FILTER	FAIR-S
Anonymous	0.1215	0.1144	0.0978	0.0337
Registered	0.0004	0.0009	0.0014	0.0028

(d)

Protected	WDVD	ORES	FILTER	FAIR-S
Benign edits by *all* users				
Anonymous	0.121	0.114	0.096	0.031
	310.7	133.1	69.2	11.9
Newcomer (1h since 1st edit)	0.138	0.109	0.109	0.037
	172.7	72.4	63.7	13.5
Newcomer (1d since 1st edit)	0.101	0.085	0.085	0.026
	170.9	68.3	58.2	10.2
Newcomer (7d since 1st edit)	0.060	0.053	0.053	0.015
	118.8	48.5	40.7	6.1
Benign edits by *anonymous* users				
Origin USA	0.053	0.091	0.053	0.021
	1.5	1.9	1.6	1.6
Origin Mexico	0.243	0.132	0.109	0.059
	3.1	2.2	2.1	2.8
Origin Spain	0.167	0.097	0.085	0.049
	2.5	1.9	1.9	2.5
Origin Japan	-0.080	-0.053	-0.001	-0.028
	0.3	0.5	1.0	0.2

(e)

FILTER Rules	Performance (*all* edits) Total	Vand.	Prob.	Bias (*benign* edits) Total	Anon.	Prob.
w/ user information	21,225	6,113	28.8%	15,112	10,445	69.1%
New user changing sth.	14,452	4,313	29.8%	10,139	6,525	64.4%
New user removing sth.	5,609	1,031	18.4%	4,578	3,560	77.8%
Possible vandalism	1,164	769	66.1%	395	360	91.1%
w/o user information	12,612	849	6.7%	11,763	739	6.3%
Self-referencing	4,030	141	3.5%	3,889	236	6.1%
Removal of gender	3,408	83	2.4%	3,325	95	2.9%
Unexpected gender	3,381	612	18.1%	2,769	292	10.5%
Miscellaneous	1,793	13	0.7%	1,780	116	6.5%
No Rules	6,970,017	15,688	0.2%	6,954,329	140,580	2.0%

(f)

Users	**Manually reviewed benign edits** (n=1,103) Observed behavior: Reverted (incorrect)	Observed behavior: Non-reverted (correct)	Expected behavior: Reverted (incorrect)	Expected behavior: Non-reverted (correct)
Anonymous	103	23	16.6	109.4
Registered	42	935	128.4	848.6

a large number compared to the small percentage of overall edits by anonymous users. FILTER rules can only be created by Wikidata's (at the time of writing) about 60 administrators,[3] who are the most active users and who carry out lots of other maintenance tasks, too.

For the new approach FAIR-S, we omit all user features, thus significantly reducing its bias. The remaining bias can be explained by unsuspicious features that are (slightly) correlated with the omitted feature `isRegisteredUser`. This effect is referred to as *indirect discrimination* [45] or *redlining* [7]. In general, it is undesirable to make all groups have exactly the same average scores as small differences might be justified by hidden confounding variables and it would be an overreaction often called *affirmative action* [48, 66].

3.3 Biases of Wikidata Reviewers

Another potential source of bias might be the human reviewers engaged in Wikidata's semi-automatic damage control system. If a reviewer encounters a poor edit, it is reverted, and otherwise typically nothing happens. To investigate the degree to which revert decisions may be biased, we used a manually annotated dataset, compiled as part of the Wikidata Vandalism Corpus 2015 (WDVC-2015) [26]. It consists of two random samples of 1,000 edits each that were rollback reverted by Wikidata reviewers and that were not reverted, respectively. These edits were carefully annotated with respect to whether they constitute vandalism, but unlike Wikidata's reviewers, the annotator was given no knowledge about the registration status (anonymous or registered) of the editor.

Among the 2,000 edits, 1,103 are *benign* edits, 145 of which were *incorrectly* reverted. Table 1f breaks down the observed behavior of Wikidata reviewers as a contingency table of registration status over revert decision correctness. Reverting a benign edit is an incorrect decision, whereas not reverting it is correct. From the marginals of this matrix (not shown), we obtain the expected behavior under the assumption of independence. We find that revert decisions by Wikidata reviewers are heavily biased against anonymous editors, since they are disproportionately often affected by incorrectly reverted edits: 103 edits by anonymous users are *incorrectly* reverted although we would expect only 16.6 edits considering the relatively small number of 126 edits by anonymous editors. Fisher's test shows that the bias against anonymous users is statistically highly significant ($p < 0.0001$). Figure 2 shows how the reviewer bias developed over time. Choosing time intervals so that an approximately equal amount of the 1,103 benign edits fell into each interval, the odds ratio severely increased over time, suggesting that bias against anonymous users among reviewers kept growing.

Only Wikidata's administrators and a few other powerful users with the so-called rollback rights can perform rollback reverts. Even when accounting for human fallibility, it appears that, on this sample of edits, anonymous editors were discriminated. We do not suspect any malicious intent of Wikidata reviewers. Rather, the fact that many true vandalism edits originate from anonymous editors may have led them to ingest some form of prejudice whenever an anonymous edit looks odd while skipping edits by registered editors without closer inspection. However, even if more vandalism originates from anonymous editors than from registered ones, this is not a sufficient justification to treat benign anonymous editors unfairly. After all, a revert decision should be based on content alone, and not on who the content comes from.

[3] https://www.wikidata.org/wiki/Special:ListGroupRights

[4] Tables c, and d were computed on the test set of the vandalism corpus WDVC-2016-Links. Vandalism scores of the models FAIR-S, WDVD, ORES, FILTER were obtained as described in Sections 4 and 6 and calibrated to approximate vandalism probabilities.

[5] Computed on the whole WDVC-2016-Links dataset.

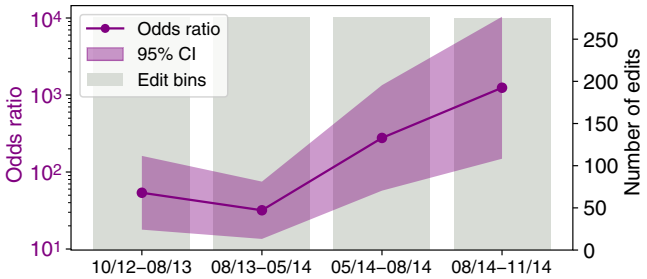


Figure 2: Reviewer bias against anonymous users as odds ratio over time, averaged over equisized edit bins.

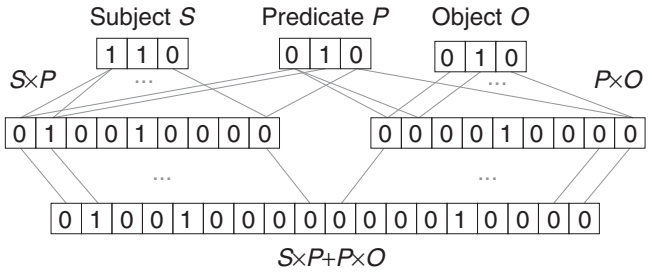


Figure 3: Example of an embedding of a subject-predicate-object triple in $n := 3$-dimensional predicate space (top), the outer products $S \times P$ and $P \times O$ (middle), and the final $2n^2 = 18$-dimensional embedding (bottom).

4 FAIR—UNBIASED VANDALISM DETECTION

Our bias analysis indicates that the main source of bias in state-of-the-art vandalism detectors are user-related features. That, together with the lack of features to characterize the actual content of Wikidata, creates a situation where bias and discrimination thrives. This section introduces FAIR, a new vandalism detector for Wikidata. We tackle the problem (1) by devising new content features based on explicit graph embeddings that protect Wikidata's primary asset, its subject-predicate-object triples between entities (FAIR-E), and, (2) by careful feature selection from the existing vandalism detectors with an eye on bias (FAIR-S).

4.1 FAIR-E: Graph Embeddings for Wikidata

Unlike for Wikipedia, where the full power of text mining can be unleashed to detect vandalism in its articles, the content of knowledge bases is much more difficult to be represented, which may explain the lack of corresponding features in Wikidata's vandalism detectors. We propose a novel graph embedding FAIR-E that results in a particularly low bias. To the best of our knowledge, this kind of graph embedding has not been used before, neither for Wikidata vandalism detection, nor elsewhere.

Editing Wikidata means editing subject-predicate-object triples. The user interface of Wikidata enforces that every edit affects exactly one triple,[6] so that representing an edit boils down to representing the edited triple. We encode the subject, predicate, and object of a triple in predicate space and capture all pairwise interactions between them (see Figure 3 for an illustration). For the subject, we encode all *outgoing* predicates as a binary vector S. The predicate is directly encoded as one-hot encoding P. For the object, we encode all *incoming* predicates as a binary vector O. The whole triple is then encoded as concatenations of pairwise combinations: $S \times P + P \times O$, where the +-operator denotes the concatenation of vectors and $\times$ the outer product, i.e., all combinations of elements. Let n be the number of predicates, then a triple vector has $2n^2$ dimensions. In Wikidata, there are about $n = 6,000$ predicates. To avoid overfitting, we restrict the set of predicates: We compute all predicates that were present in the item graph at the end of the training set and we set all other predicates in our embedding vectors to zero, thus effectively removing predicates to attributes such as strings and numbers and removing predicates that have been deleted earlier. Moreover, we restrict the representation to $n := 100$ predicates for S, P, and O, respectively. Using a suitable training dataset, we compute the top 100 predicates according to frequency for S, P, and O independently, and then proceed to computing the combined representation, yielding a 20,000-dimensional vector. As machine learning algorithm, we employ logistic regression.

To give an example how the embedding works: Suppose we want to represent the newly created triple ⟨Alejandro Cuello (Q15924626), occupation (P106), actor (Q33999)⟩. The subject is a person with predicates "date of birth" (P569), "sex or gender" (P21), and "country of citizenship" (P27), the triple predicate is "occupation" (P106), and the object has incoming predicates "occupation" (P106), "field of work" (P101), and "position held" (P39). Our model learns that the triple predicate "occupation" goes well with the incoming object predicates "occupation", "field of work" and "position held." In contrast, the triple ⟨Steve Jobs (Q19837), instance of (P31), animal (Q729)⟩ is highly unusual. Adding another "instance of" (P31) relationship to a subject with subject predicates "country of citizenship" (P27) and "sex or gender" (P21) often points to vandalism; as does adding an "instance of" relationship to an object with incoming object predicate "parent taxon" (P171).

Variants. We experimented with multiple variants of our approach, including different values for n, different interactions between subject, predicate, and object (see Section 6), and predicates to attributes. We experimented with taking the *outgoing* predicates of the object instead of the *incoming* predicates, too. However, the above encoding outperformed all other variants on the validation set in terms of predictive performance. Varying the strength of L2 regularization had little effect on predictive performance but had a large effect on bias, so we disabled regularization to minimize bias (setting scikit-learn's parameter $C = 10000$). We also experimented with the path ranking algorithm [14, 20], but the resulting model did not outperform the above model.

Limitations. If a subject-predicate-object triple uses a predicate not among the ones selected, its embedding is the zero vector, which was the case for about 15% of triples in our dataset. Our logistic regression classifier assigns the same, small vandalism probability to all these cases (determined by the intercept of logistic regression). For triples updated in a revision, we only consider the new version of the triple. Our embedding does not distinguish additions and removals of triples and we leave it for future work to incorporate such a distinction in a way that improves performance. As the set and distribution of predicates changes over time, retraining the classifier may be required from time to time.

[6] Wikidata's API allows for editing multiple triples at once, which is rather the exception. In that case, we use the main triple from Wikidata's automatically generated comment.

4.2 FAIR-S: Selecting Unbiased Features

As an alternative approach to debiasing vandalism detectors, we systematically evaluated candidate feature subsets for subject-predicate-object triples from the union of all features developed for previous detectors and call our model FAIR-S: We used the features of WDVD [27], the features of ORES [25, 49], and a few new features, including the ones from FAIR-E, while intentionally omitting user-related features and features not targeting subject-predicate-object triples between entities. Features were added to the set until predictive performance on our validation set did not improve anymore. Table 2 shows the resulting feature set, which constitutes a local optimum in the space of possible feature subsets, since removing any of the features or adding further features from our candidate set decreases performance in terms of ROC_{AUC}. Seven features from Heindorf et al. [27], three features from Sarabadani et al. [49], and four new features were selected. We use a random forest with 32 trees and a maximal depth of 16 for our experiments.[7] In a pilot study, we experimented with other algorithms including logistic regression, neural networks, and gradient boosted decision trees, carefully tuning their hyperparameters, yet, corroborating previous findings, random forests outperformed them all in this setting [27, 49]. Below, the features are described in more detail.

Subject. We characterize a subject by how many different users have edited it (`subjectLogCumUniqueUsers`), how many edits have been performed on it (`subjectLogFrequency`), how many labels and aliases it has (`subjectNumberOfLabels`, `subjectNumberOfAliases`), the number of words in its English label (`subjectLabelWordLength`), and how often the predicate has been edited for this subject (`subjectPredicateCumFrequency`). These features signal how popular a subject is (`subjectLogCumUniqueUsers`), how large a subject is (`subjectNumberOfLabels`), and how complex a subject is (`subjectLabelWordLength`). The number of words serves as a proxy for the complexity of an item. Items with one-word labels might be about organizations or places, items with two words are typically persons, and items with many words are complex topics. Other forms of encoding from our candidate features did not yield any improvement. In general, we found that subject features are prone to overfitting, since there are about 2.5 million different subjects and most subjects have never been vandalized.

Predicate. We represent each predicate by the number of times it appears in the training set (`predicateFrequency`). Other forms of encoding from our candidate feature set did not yield improvements.

Object. In contrast to subject features, object features are less prone to overfitting since objects are reused many times. We capture the popularity of an object by the number of revisions it appears in (`objectFrequency`) and the number of times an object has been edited in combination with a given predicate (`objectPredicateCumFrequency`). Moreover, one of the features of FAIR-E was included: We compute the set of all incoming predicates of an object in the knowledge graph and take this set as a feature (`objectPredicateEmbedFrequency`), the idea being that the incoming predicates of an object capture in what context and for what purposes it should be used. To avoid overfitting, we restrict this set to the top 100 most frequent predicates on the training set.

[7] We use a fixed seed for feature selection. Due to some low-performing and redundant features, the feature set is not stable for different seeds. But even slightly different feature sets yield similar predictive performance and bias values.

Table 2: Features of our vandalism detector FAIR-S and their performance scores and biases on our test dataset, computed with a random forest with 32 trees and maximal depth 16.

Feature Group / Feature	Reference	Performance ROC_{AUC}	Performance PR_{AUC}	Bias Diff	Bias Ratio
Subject	—	0.907	0.071	0.01199	4.05
`subjectLogCumUniqueUsers`	[27]	0.901	0.052	0.01005	3.51
`subjectLogFrequency`	[27]	0.880	0.042	0.00795	2.95
`subjectNumberOfLabels`	[49]	0.859	0.031	0.00321	1.76
`subjectNumberOfAliases`	[49]	0.761	0.026	0.00198	1.47
`subjectLabelWordLength`	—	0.721	0.008	0.00165	1.38
`subjectPredicateCumFrequency`	—	0.687	0.008	0.00096	1.22
Predicate	—	0.729	0.011	0.00196	1.46
`predicateFrequency`	[27]	0.729	0.011	0.00196	1.46
Object	—	0.729	0.026	0.00234	1.55
`objectPredicateEmbedFrequency`	—	0.682	0.013	0.00188	1.44
`objectFrequency`	[27]	0.650	0.010	0.00107	1.25
`objectPredicateCumFrequency`	—	0.613	0.007	0.00014	1.03
Edit Features	—	0.889	0.062	0.00926	3.32
`editProportionOfTriplesAdded`	[49]	0.866	0.022	0.00418	2.00
`editSubactionFrequency`	[27]	0.851	0.020	0.00381	1.91
`editPrevActionFrequency`	[27]	0.629	0.024	0.00297	1.70
`editActionFrequency`	[27]	0.575	0.005	0.00052	1.12

Edit. We characterize the change of an edit with four features: the edit operation, such as create, update, or remove (`editActionFrequency`, `editSubactionFrequency`), the previous action performed on the same item (`editPrevActionFrequency`), and the size of the edit operation relative to the current size of the item (`editProportionOfTriplesAdded`).

Variants. Both for the subject and the object, we experimented with using their *super types* according to Wikidata's *instance of* hierarchy as a feature. Neither of these features improved predictive performance nor bias. For the object, similar information is already captured with our predicate embedding (e.g., in `objectPredicateEmbedFrequency`). Also, a bag-of-words model of the subject's and the object's textual labels and descriptions did not help.

5 EVALUATION DATA

This section describes the Wikidata datasets used in our evaluation, how they have been pre-processed, split into training, validation, and test sets, as well as their characteristics. Moreover, we investigate the proportion of model bias due to data bias.

5.1 Wikidata Revision History and Graph

We derive our dataset from the Wikidata Vandalism Corpus 2016 (WDVC-2016) [25]. This dataset ranges from October 2012 to June 2016 and contains all human edits along with labels whether the edit is considered vandalism. The labels were automatically obtained by analyzing the use of the rollback tool of Wikidata, which is explicitly meant to revert vandalism [26, 27]. A manual analysis revealed that 86% of reverted revisions are indeed vandalism and only about 1% of the non-reverted revisions. We derive a subset of this dataset, called WDVC-2016-Links, that contains all edits pertaining to the knowledge *graph*, i.e., the actual content of

Table 3: Evaluation data: (a) Derivation of the new dataset WDVC-2016-Links from WDVC-2016. (b) Datasets for training, validation, and test in terms of vandalism triples, total triples, subjects, predicates, and objects. (c) The vandalism corpus WDVC-2016-Links broken down by domain. (d) Estimation of the true bias without noisy labels. We perform stratified sampling on the test dataset and manually annotate the samples to estimate the true bias. (e) Wikidata vandalism corpus WDVC-2016-Links over time by domain for all edits, and, (f) for vandalism edits only.

(a)

Filtering steps	Edits	Prop.
WDVC-2016	82,679,918	100 %
w/o semi-automatic editing tools	35,462,023	43 %
w/o labels, descriptions, aliases	26,959,949	33 %
w/o sitelinks	17,371,199	21 %
w/o qualifiers, references	15,883,276	19 %
w/o attributes	11,631,335	14 %
w/o item creation, merges, misc	7,002,290	8 %
WDVC-2016-Links	7,002,290	8 %

(b)

Dataset	Vand.	Edits	Subjects	Pred.	Objects
Training	16,608	5,890,968	2,187,993	747	790,960
Validation	2,604	550,798	201,102	610	131,666
Test	2,429	560,524	212,902	603	131,479

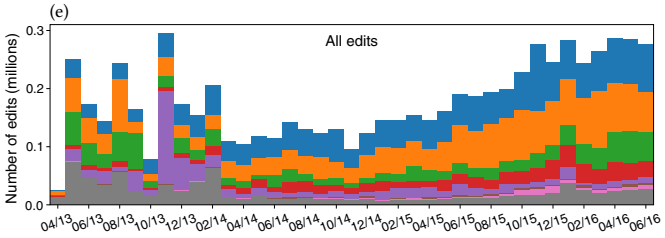
(c)

Domain	Performance (*all* edits) Total	Vand.	Prob.	Bias (*benign* edits) Total	Anon.	Prob.
Generic	1,863,273	5,755	0.31%	1,857,518	36,174	1.95%
Person	1,905,257	10,539	0.55%	1,894,718	46,671	2.46%
Organization	67,890	312	0.46%	67,578	2,601	3.85%
Events	96,115	68	0.07%	96,047	639	0.67%
Works	618,464	1,598	0.26%	616,866	32,252	5.23%
Terms	680,904	437	0.06%	680,467	2,279	0.33%
Place	951,589	1,628	0.17%	949,961	16,857	1.77%
Others	818,798	1,304	0.16%	817,494	13,841	1.69%

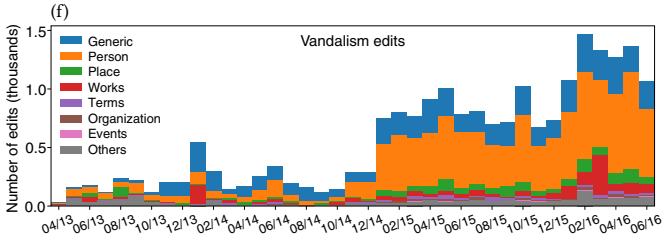
(d)

Corpus User	Truth	n	Gold Truth	n	Weight	WDVD ∅ Score	True Bias wt. sum
Anon.	vand.	2,061	benign	19	1.98%	0.279443	0.005526
			vand.	231	n/a	n/a	n/a
	benign	10,270	benign	189	98.02%	0.079500	0.077928
			vand.	61	n/a	n/a	n/a
Reg.	vand.	368	benign	29	0.01%	0.032105	0.000003
			vand.	221	n/a	n/a	n/a
	benign	547,825	benign	239	99.99%	0.000579	0.000579
			vand.	11	n/a	n/a	n/a
Sum		560,524		1,000		Diff.	0.083
						Ratio	143.4

(e)



(f)



the knowledge base. Table 3a details the successive filtering steps we applied. We omit edit operations by semi-automatic editing tools like Wikidata Game because they perform little vandalism and we believe tailored quality checks should rather be directly built into them. Moreover, we remove edits affecting labels, descriptions, aliases, sitelinks, qualifiers, references, attributes, and special operations for item creation and item merging, yielding a dataset consisting of edits affecting links between entities.

While the Wikidata vandalism corpus contains the data in incremental form, i.e., edit by edit, for our novel content-based features, we need to represent the data as a graph. For these features, we base our computation on the static Wikidata graph from February 29, 2016, i.e., the graph capturing the data ahead of the time interval covered by the validation set.[8]

5.2 Datasets for Training, Validation, and Test

Following previous work, we split the dataset by time to enable classification of edits as soon as they arrive without exploiting information "from the future." Table 3b shows the data splits along key statistics on triples. Overall, the dataset contains 21,641 vandalism cases among 7 million edits; a class imbalance of 0.3%. Our training set ranges from April 18, 2013, to February 29, 2016, consisting of 6 million edits. For efficiency, we derive triples from Wikidata's automatically generated edit comments, which were not available before April 2013. The two-month periods for validation and testing range from March to April and May to June, respectively, each comprising more than 550,000 edits. The large number of subjects renders subject feature prone to overfitting.

[8] https://archive.org/download/wikidata-json-20160229

5.3 Corpus Analysis

Table 3c breaks down the vandalism corpus by domain. We categorize edits by domain according to Wikidata's classification system.[9] In both absolute and relative terms, the most vandalism edits occur on persons such as Barack Obama, Christiano Ronaldo, and Justin Bieber, potentially causing high vandalism scores for all edits in this category. At the same time, among all categories, persons have the highest number of *benign* edits by anonymous users which still should receive low vandalism scores. This illustrates the inherent difficulty to achieve high predictive performance and low bias at the same time. An unbiased vandalism model must discriminate vandalism from non-vandalism without discriminating certain editor groups. Biases are still relatively small in this case, compared to the abuse filter in Table 1e.

The plots (e) and (f) in Table 3 overview our corpus both in terms of time and domains. The number of edits per month increases, as does the number of vandalism cases. The drop of manual edits around March 2014 can be explained by emerging automatic tools that automate routine data import and maintenance tasks, thus temporarily reducing manual edit operations. The increasing vandalism around January 2015 coincides with the redesign of Wikidata's user interface, driving vandals from the header area of labels, descriptions, and aliases to the body area containing links and statements. The person domain grows fastest, both in terms of benign and vandalism edits. Overall, the distribution of vandalism across domains remains relatively stable over time, with the exception of two outliers: an increase of vandalism affecting the works domain in January 2013 and March 2016.

[9] https://www.wikidata.org/w/index.php?title=Wikidata:List_of_properties&oldid=552806031

5.4 Estimating Model Bias Without Data Bias

To investigate the effect of the noisy automatic vandalism labels on bias, we carry out a manual analysis: Drawing a stratified sample of 1,000 edits with respect to registration status (anonymous vs. registered) and automatic ground truth (vandalism vs. benign), we obtain gold labels by manual review of 250 edits per stratum, enabling an estimation of the true bias. The reviewer was kept unaware to which stratum a given edit belonged. Table 3d summarizes our findings and shows the average vandalism scores of WDVD for the data subsets we found to be *truly benign* and that form the basis for the bias computation. The WDVD scores were obtained by calibrating its original scores with isotonic regression on the large-scale corpus ground truth. As a result, WDVD's estimated bias without noisy labels is a score difference of 0.083 and a score ratio of 143.4. This is about half the bias measured on the automatically annotated corpus (0.121 and 310.7, respectively), which still necessitates countermeasures to reduce it. We leave creating a larger unbiased dataset that allows for training, calibrating, and evaluating (unbiased) vandalism detectors to future work.

6 EVALUATION AND BIAS OPTIMIZATION

We carried out a series of experiments to minimize the bias of our models while maintaining a competitive predictive performance. Table 4a gives an overview of the results we obtained for FAIR-E and FAIR-S, the state-of-the-art baselines WDVD, ORES, and FILTER, and variants thereof obtained by post-processing scores, reweighting training samples, and combining models in an ensemble. FAIR-E exhibits low bias at reasonable predictive performance; FAIR-S achieves higher predictive performance at a higher bias.

The literature has investigated trade-offs between different notions of fairness and predictive performance [4, 8, 10, 33], indicating that the two optimization goals cannot be satisfied at the same time, dependent on the notion of fairness and performance measure. Nevertheless, a sustainable model must strike a balance between fairness and predictive performance to ensure the long-term success of an online community. The state-of-the-art in vandalism detection has not yet reached its full potential in this respect.

In what follows, we briefly recap the baseline models and the prediction performance measures, and then detail the aforementioned experiments. To ensure reproducibility, the code base underlying our research is published alongside this paper.[10] While feature extraction is carried out in Java, our experiments are implemented in Python using scikit-learn, version 0.19.1. To calibrate classifier scores before computing bias, we use isotonic regression.

6.1 Baselines and Performance Measures

For the final evaluation on the test set, all baselines were trained on both the training and the validation set of WDVC-2016-Links, i.e., on about 6 million revisions ranging from April 2013 to April 2016. Unless otherwise indicated, the same hyperparameters and machine learning algorithms as reported in Heindorf et al. [25] were used: WDVD employs multiple instance learning on top of bagging and random forests. It uses 16 random forests, each build on 1/16 of the training dataset with the forests consisting of 8 trees, each having a maximal depth of 32 with two features per split using the default Gini split criterion. ORES uses a random forest with 80 decision trees considering 'log2' feature per split using the 'entropy' criterion. FILTER uses a random forest with scikit-learn's default hyperparameters, i.e., 10 decision trees without a maximal depth.

We employ performance measures for both predictive performance and bias performance, the latter using the bias measures introduced in Section 3. Predictive performance is measured in terms of area under curve of the receiver operating characteristics (ROC_{AUC}) and area under the precision-recall curve (PR_{AUC}). Both curves capture predictive performance across different operating points, where each point on one of the curves has a corresponding point on the other [12]. However, corresponding points are weighted differently when computing the area under the curves: ROC_{AUC} emphasizes operating points at high recall ranges and PR_{AUC} emphasizes points at high precision ranges. Hence, ROC_{AUC} is best suited to analyze semi-automatic vandalism detectors, where revisions are ranked and manually checked by reviewers, whereas, PR_{AUC} is best suited to analyze fully automatic detection systems, where vandalism is reverted without human intervention.

6.2 Debiasing via Feature Engineering

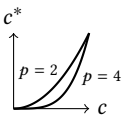
We obtain our debiased models FAIR-E and FAIR-S via feature engineering. For FAIR-E, we employ graph embeddings capturing the content of an edit in contrast to meta data such as user reputation, thus obtaining a bias ratio of only 5.6 on the test set compared to over 300 of the state-of-the-art approach WDVD. We experimented with different variants of what feature interactions to consider and Table 4b overviews our results on the validation set. In general, we observe that, with increasing model complexity, both predictive performance and bias increase. Given the relatively low bias, we opt to optimize predictive performance and choose the model $S \times P + P \times O$ as FAIR-E, taking into account all pairwise interactions between subject and predicate as well as predicate and object.

Based on feature selection and explicitly omitting user features, FAIR-S achieves a bias ratio of only 11.9 at 0.316 PR_{AUC} and 0.963 ROC_{AUC}. Plot (c) in Table 4 shows the trade-offs between bias and predictive performance obtainable by adding or by removing single features from the set of candidate features and outlines the Pareto front on the validation set. For example, removing the feature `objectPredicateCumFrequency` from FAIR-S yields the left-most point with both lower predictive performance and lower bias. We choose the best predictive model without user features as FAIR-S. Compared to WDVD, which consists of 47 features, FAIR-S consists of only 14 features, thus simplifying the model, reducing its runtime, and rendering its decisions much more explainable.

6.3 Debiasing via Post-Processing Scores

As an alternative to feature engineering, we can obtain comparable results in terms of bias and predictive performance by post-processing the scores of WDVD. We scale uncalibrated scores $c(i)$ from sample i of the protected class, while leaving those from the unprotected class unchanged, thus obtaining $c^*(i)$:

$$c^*(i) = \begin{cases} c(i)^p, & \text{revision } i \text{ from anon. user} \\ c(i), & \text{revision } i \text{ from reg. user} \end{cases}$$



[10] http://www.heindorf.me/wdvd

Table 4: Evaluation results: (a) Performance and bias of approaches on the test set of the vandalism corpus WDVC-2016-Links. (b) Optimization of FAIR-E on the validation set by testing all pairwise feature interactions between subject (S), predicate (P), and object (O) in predicate space. (c) Optimization of FAIR-S on the validation set. The plot shows ROC_{AUC} over bias of FAIR-S (purple) after adding (green) or removing (orange) a feature. The dotted line shows non-dominated points on the Pareto front. (d) Error analysis with respect to bias on our golden dataset.

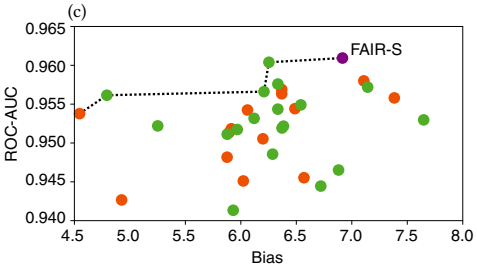
(a)

Debiasing Experiment Model	Performance		Bias	
	PR_{AUC}	ROC_{AUC}	Diff	Ratio
Feature engineering				
FAIR-E	0.177	0.865	0.016	5.6
FAIR-S	0.316	0.963	0.031	11.9
Post-processing scores				
WDVD with p=3.88	0.230	0.966	0.015	5.3
WDVD with p=3.22	0.340	0.976	0.030	11.8
Weighting training samples				
WDVD with $\alpha = 8.1$	0.160	0.963	0.015	5.3
WDVD with $\alpha = 4.3$	0.314	0.973	0.030	11.5
Combining models				
FAIR-E + WDVD	0.229	0.974	0.033	11.7
FAIR-E + ORES	0.238	0.967	0.033	11.8
FAIR-E + FILTER	0.224	0.953	0.033	11.8
Baselines				
WDVD	0.547	0.990	0.121	310.7
ORES	0.434	0.965	0.114	133.1
FILTER	0.302	0.924	0.096	69.2

(b)

Features	Performance		Bias	
	PR_{AUC}	ROC_{AUC}	Diff	Ratio
$\boldsymbol{S \times P + P \times O}$	**0.112**	**0.848**	**0.0081**	**2.90**
$S \times P + S \times O$	0.095	0.741	0.0071	2.65
$S \times O + P \times O$	0.068	0.721	0.0057	2.28
$S \times P \times O$	0.073	0.675	0.0064	2.45
$S \times P$	0.025	0.718	0.0021	1.45
$S \times O$	0.028	0.629	0.0023	1.50
$P \times O$	0.047	0.733	0.0031	1.69
$S + P + O$	0.068	0.739	0.0042	1.94
$S + P$	0.038	0.690	0.0036	1.78
$S + O$	0.034	0.719	0.0024	1.52
$P + O$	0.068	0.739	0.0042	1.94
S	0.020	0.643	0.0012	1.26
P	0.008	0.659	0.0008	1.17
O	0.011	0.663	0.0004	1.09

(c)



(d)

Users	Manually reviewed edits (n=1,000)			
Gold Truth:	Vandalism (n=524)		Benign (n=476)	
FAIR-S:	Vandalism (correct)	Benign (incorrect)	Vandalism (incorrect)	Benign (correct)
Anonymous	250	42	73	135
Registered	173	59	28	240

After scaling, we apply isotonic regression to calibrate the scores to represent probabilities, i.e., $c^*(i) \approx Pr(i = \text{vandalism} \mid x_i)$, in order to prevent bias being lower just due to a smaller interval of scores. By experimentally varying the parameter $p > 1$, we obtain classifiers that have approximately the same bias as our models FAIR-E and FAIR-S, allowing for performance comparisons. As Table 4a shows, we can achieve approximately the same predictive performance as with feature engineering. The advantage of post-processing scores is that the model does not have to be retrained. However, the model's decisions are also much more difficult to understand. It is less clear what effect single features have, thus hindering an intuitive explanation to Wikidata editors whose edits have been reverted. Apart from the polynomial scaling function introduced above, we experimented with other families of functions, including linear, fractional, and exponential functions. Polynomial scaling outperformed all other variants in terms of PR_{AUC}.

6.4 Debiasing via Weighting Training Samples

As another option to reduce biases, we experimented with reweighting training samples. We start by assigning each of the four groups of protected/unprotected, benign/vandalism revisions the same weight. Then we increase the weight of benign edits by protected users such that it becomes prohibitively expensive for the algorithm to make mistakes on this set of revisions. Symmetrically, we increase the weight of vandalism edits by unprotected users. Thus, using the notation from Section 3, the weight of each training sample i is obtained by:

$$w(i) = \begin{cases} \alpha \cdot \frac{1}{|A|}, & \text{revision } i \in A \\ 1 \cdot \frac{1}{|B|}, & \text{revision } i \in B \\ 1 \cdot \frac{1}{|D|}, & \text{revision } i \in D \\ \alpha \cdot \frac{1}{|E|}, & \text{revision } i \in E \end{cases}$$

		Truth benign	Truth vand.	∪
Protected	yes	A	B	C
	no	D	E	F
∪		G	H	I

By varying the constant $\alpha > 1$, we obtain classifiers with varying trade-offs between bias and predictive performance. For the weighting experiment in Table 4a, we reduce the maximal tree depth of WDVD from 32 to 16 to increase the number of training samples per leaf, giving the weighting a larger effect. We experimentally determine α to achieve approximately the same bias as our models FAIR-E and FAIR-S, allowing for performance comparisons. However, predictive performance is lower than that obtained by feature engineering or post-processing scores. We believe the reason is that reweighting leads to overfitting on certain subsets of the data and the effect on bias is rather indirect by reducing the misclassification rate on these subsets, instead of tackling bias directly. Hence, we suggest to rather use feature engineering or post-processing scores. Besides, we also tried to employ separate constants α_1 and α_2 per group, but no better bias and predictive performance was achieved, so we resorted to the basic weighting scheme.

6.5 Debiasing via Combining Models

In yet another debiasing attempt, we train ensembles of models with high predictive performance and low-bias models to derive a model with characteristics in between. A weighted average between the scores of the two models is computed. Table 4a shows the results: While it is possible to use ensembles to reconcile predictive performance with bias, this method appears to be inferior to feature engineering and post-processing scores.

6.6 Debiasing Error Analysis

Table 4d shows the results of an error analysis for FAIR-S. The continuous scores of FAIR-S were converted to binary decisions, so that predicted vandalism prevalence equals the vandalism prevalence in the golden set from Table 3d. The table shows, that 73 benign edits by anonymous users were classified as vandalism. Manually investigating these edits, we find that most of the edited triples are

updates (53), rather than removals (11) or creations (9). In most cases, the subject is a human. Predicates frequently affected are "occupation" (P106, 10 cases), "instance of" (P31, 8), "country of citizenship" (P27, 6), and "sex or gender" (P21, 6). It might be beneficial to develop vandalism detectors tailored to update operations, e.g., by comparing the old and new object of a triple. Also specific features targeting problematic predicates might help. Beyond intrinsic plausibility checks, also double-checks in external databases or web search engines seem promising.

7 DISCUSSION

Biases in machine learning are still widely neglected by researchers and practitioners in machine learning and data science, who primarily focus on optimizing predictive performance, disregarding fairness constraints that are essential for the long-term success of online communities. Approaches that discriminate against sex, race, religion, or that are otherwise biased against minorities—or even against majorities[11]—jeopardize an online community's long-term goal of promoting a feeling of fairness, security, and belonging. We hope that our endeavor helps towards increasing awareness.

Our bias definition emphasizes fair treatment of *benign* edits, which is also known as *equality of opportunity* [23], and which we believe is well-aligned to the goal of newcomer retention. Fair treatment of both *benign* and *vandalism* edits would correspond to the stricter fairness notion *equalized odds* [23], making it potentially more challenging to find a good balance between predictive performance and fairness. Another fairness notion that is sometimes mentioned by the media, lawmakers, and the literature is *statistical parity* [16, 17], requiring that the same proportions of anonymous and registered edits are classified as vandalism. This neglects that the two populations might have different vandalism prevalences, and Dwork et al. [16] argue that it is an inadequate notion of fairness. First results [4, 8, 10, 33] suggest that there is a trade-off between fairness and predictive performance for many notions of fairness, requiring difficult trade-offs when the set of features is fixed. We believe decisions about an edit should focus on the edit's "content", instead of the reputation of the user who submitted it. But even without features prone to incur bias, there can still be bias against certain groups, since seemingly harmless features might correlate with biased features. For example, in Table 2, features related to edit operations and subjects introduce (a small) bias against anonymous edits. However, forcing two groups to have exactly the same score distribution might introduce reverse discrimination, also known as affirmative action [3, 48, 66], since there might be hidden confounding variables justifying certain differences. In the future, causal modeling [32, 34, 64] might help to decide what causal relations to consider.

Wikidata makes for an interesting case study to analyze and mitigate biases as it has one of the largest online communities and provides opportunities to pay particular attention to the content rather than the user reputation. We believe some lessons learned in this project can be transferred to other projects, too. It is common practice to identify malicious edits via meta data such as geolocation of IP addresses, age of user account, or browser information such as user language. While those features are simple to obtain, they do not directly judge the quality of an edit and harm well-intentioned users. Our graph embeddings serve as an example of how features purely judging the content of an edit help to reduce unintentional biases. Perhaps surprisingly, we can obtain comparable performance-bias trade-offs by artificially scaling scores after a biased model has been trained. On the one hand, this approach might serve as an easy route to debias existing models. On the other hand, we feel that it is rather a "black-box approach" and might be less suitable for understanding predictions and explaining them to editors.

Finally, we would like to point out some limitations and directions for future research. Our model FAIR-E was specifically designed for edits affecting links between entities and extending it to attributes, labels, descriptions, and aliases might be challenging, with the exception of our model FAIR-S. Here, such an extension is rather straightforward as it selects features from WDVD which is capable of classifying all edits in Wikidata. Our analysis is based on a dataset derived from the rollback actions of Wikidata reviewers and we observed some reviewer bias in this dataset. We leave it to future work to create a large-scale, unbiased dataset. For example, the Wikimedia Foundation might decide to hide user information from Wikidata reviewers, forcing them to purely consider the content of an edit. However, this might make the review process more expensive as reviewers cannot skip or quickly skim large amounts of edits by registered users anymore. Besides fair vandalism detectors, further means to increase newcomer retention might include vandalism detectors explaining their decisions, improved user interfaces, onboarding programs for newcomers, increased social interactions, and gamification—all accompanied by data-driven processes such as A/B testing or reinforcement learning.

8 CONCLUSION AND OUTLOOK

Machine learning models ingest biases through training data and features—sometimes even aggravating them—fueling a vicious cycle of reinforcing biases in a larger system. In this work, we developed a vandalism detector for Wikidata's damage control system that does not contribute significant bias of its own. Compared to the state-of-the-art, it considerably reduces bias against edits by anonymous and newly registered editors. We achieve this result by omitting user-related features and by developing features that purely encode the content of an edit, rather than any meta information. In the future, we plan to investigate biases at further online platforms and to develop a general framework for bias mitigation, e.g., by employing evolutionary algorithms to explore the Pareto front of non-dominated models in bias-performance space.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (CRC 901). Special thanks go to the anonymous reviewers for their helpful comments.

[11] Anonymous *edits* are a clear minority in our dataset since only about 2.4% of edits are by anonymous editors. However, anonymous *editors* might possibly be in the majority, since the number of distinct IP addresses from anonymous editors is about 2.5 times higher than the number of distinct user accounts from registered editors (our data does not reveal how often the same anonymous editor uses a different IP address).

REFERENCES

[1] B. T. Adler, L. de Alfaro, S. M. Mola-Velasco, P. Rosso, and A. G. West. 2011. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In *CICLing*. Springer, 277–288.

[2] R. Baeza-Yates. 2018. Bias on the Web. *Commun. ACM* 61, 6 (2018), 54–61.

[3] S. Barocas and A. D. Selbst. 2016. Big data's disparate impact. *Cal. L. Rev.* 104 (2016), 671.

[4] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. 2018. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research* (2018).

[5] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *NIPS*. 4349–4357.

[6] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[7] T. Calders and S. Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.* 21, 2 (2010), 277–292.

[8] A. Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.

[9] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PLOS ONE* 10, 6 (2015), 1–13.

[10] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. 2017. Algorithmic Decision Making and the Cost of Fairness. In *KDD*. ACM, 797–806.

[11] F. Darari, S. Razniewski, R. E. Prasojo, and W. Nutt. 2016. Enabling Fine-Grained RDF Data Completeness Assessment. In *ICWE*. Springer, 170–187.

[12] J. Davis and M. Goadrich. 2006. The Relationship Between Precision-Recall and ROC Curves. In *ICML*. ACM, 233–240.

[13] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *AIES*. ACM, 67–73.

[14] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *KDD*. ACM, 601–610.

[15] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. 2016. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. *IEEE Data Eng. Bull.* 39, 2 (2016), 106–117.

[16] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. 2012. Fairness Through Awareness. In *ITCS*. ACM, 214–226.

[17] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *KDD*. ACM, 259–268.

[18] L. Galárraga, S. Razniewski, A. Amarilli, and F. M. Suchanek. 2017. Predicting Completeness in Knowledge Bases. In *WSDM*. ACM, 375–383.

[19] A. Gangemi, A. G. Nuzzolese, V. Presutti, F. Draicchio, A. Musetti, and P. Ciancarini. 2012. Automatic Typing of DBpedia Entities. In *ISWC*. 65–81.

[20] M. Gardner and T. M. Mitchell. 2015. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *EMNLP*. ACL, 1488–1498.

[21] A. Halfaker, R. S. Geiger, J. T. Morgan, and J. Riedl. 2013. The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity is Causing Its Decline. *American Behavioral Scientist* 57, 5 (2013), 664–688.

[22] A. Halfaker, A. Kittur, and J. Riedl. 2011. Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. In *Int. Sym. Wikis*. 163–172.

[23] M. Hardt, E. Price, and N. Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NIPS*. 3315–3323.

[24] S. Heindorf, M. Potthast, H. Bast, B. Buchhold, and E. Haussmann. 2017. WSDM Cup 2017: Vandalism Detection and Triple Scoring. In *WSDM*. ACM, 827–828.

[25] S. Heindorf, M. Potthast, G. Engels, and B. Stein. 2017. Overview of the Wikidata Vandalism Detection Task at the WSDM Cup 2017. In *WSDM Cup 2017 Notebook Papers*.

[26] S. Heindorf, M. Potthast, B. Stein, and G. Engels. 2015. Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis. In *SIGIR*. ACM, 831–834.

[27] S. Heindorf, M. Potthast, B. Stein, and G. Engels. 2016. Vandalism Detection in Wikidata. In *CIKM*. ACM, 327–336.

[28] A. Jain and P. Pantel. 2010. FactRank: Random Walks on a Web of Facts. In *COLING*. Tsinghua University Press, 501–509.

[29] S. Javanmardi, D. W. McDonald, and C. V. Lopes. 2011. Vandalism Detection in Wikipedia: A High-Performing, Feature-Rich Model and its Reduction Through Lasso. In *Int. Sym. Wikis*. ACM, 82–90.

[30] F. Kamiran, T. Calders, and M. Pechenizkiy. 2010. Discrimination Aware Decision Tree Learning. In *ICDM*. IEEE Computer Society, 869–874.

[31] J. Kiesel, M. Potthast, M. Hagen, and B. Stein. 2017. Spatio-Temporal Analysis of Reverted Wikipedia Edits. In *ICWSM*. AAAI Press, 122–131.

[32] N. Kilbertus, M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. 2017. Avoiding Discrimination through Causal Reasoning. In *NIPS*. 656–666.

[33] J. M. Kleinberg, S. Mullainathan, and M. Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *ITCS*, Vol. 67. 43:1–43:23.

[34] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva. 2017. Counterfactual Fairness. In *NIPS*. 4069–4079.

[35] J. Lajus and F. M. Suchanek. 2018. Are All People Married?: Determining Obligatory Attributes in Knowledge Bases. In *WWW*. ACM, 1115–1124.

[36] N. Lao, T. M. Mitchell, and W. W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *EMNLP*. ACL, 529–539.

[37] J. Lehmann, D. Gerber, M. Morsey, and A. N. Ngomo. 2012. DeFacto - Deep Fact Validation. In *ISWC*. Springer, 312–327.

[38] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. AAAI Press, 2181–2187.

[39] A. Melo, H. Paulheim, and J. Völker. 2016. Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification. In *WIMS*. ACM, 14:1–14:10.

[40] E. Minkov, W. W. Cohen, and A. Y. Ng. 2006. Contextual Search and Name Disambiguation in Email Using Graphs. In *SIGIR*. ACM, 27–34.

[41] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[42] M. Nickel, V. Tresp, and H. Kriegel. 2012. Factorizing YAGO: Scalable Machine Learning for Linked Data. In *WWW*. ACM, 271–280.

[43] C. Nishioka and A. Scherp. 2018. Analysing the Evolution of Knowledge Graphs for the Purpose of Change Verification. In *ICSC*. IEEE Computer Society, 25–32.

[44] H. Paulheim and C. Bizer. 2013. Type Inference on Noisy RDF Data. In *ISWC*. Springer, 510–525.

[45] D. Pedreschi, S. Ruggieri, and F. Turini. 2009. Measuring Discrimination in Socially-Sensitive Decision Records. In *SDM*. SIAM, 581–592.

[46] M. Potthast, B. Stein, and R. Gerling. 2008. Automatic Vandalism Detection in Wikipedia. In *ECIR*. Springer, 663–668.

[47] E. Raff, J. Sylvester, and S. Mills. 2018. Fair Forests: Regularized Tree Induction to Minimize Model Bias. In *AIES*. ACM, 243–250.

[48] A. Romei and S. Ruggieri. 2014. A multidisciplinary survey on discrimination analysis. *Knowledge Eng. Review* 29, 5 (2014), 582–638.

[49] A. Sarabadani, A. Halfaker, and D. Taraborelli. 2017. Building Automated Vandalism Detection Tools for Wikidata. In *WWW (Companion Volume)*. ACM, 1647–1654.

[50] J. Schneider, B. S. Gelley, and A. Halfaker. 2014. Accept, decline, postpone: How newcomer productivity is reduced in English Wikipedia by pre-publication review. In *OpenSym*. ACM, 26:1–26:10.

[51] B. Shi and T. Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowl.-Based Syst.* 104 (2016), 123–133.

[52] C. H. Tan, E. Agichtein, P. Ipeirotis, and E. Gabrilovich. 2014. Trust, but Verify: Predicting Contribution Quality for Knowledge Base Construction and Curation. In *WSDM*. ACM, 553–562.

[53] A. Torralba and A. A. Efros. 2011. Unbiased Look at Dataset Bias. In *CVPR*. IEEE Computer Society, 1521–1528.

[54] K. Tran and P. Christen. 2013. Cross Language Prediction of Vandalism on Wikipedia Using Article Views and Revisions. In *PAKDD*. Springer, 268–279.

[55] Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* 29, 12 (2017), 2724–2743.

[56] W. Y. Wang and K. McKeown. 2010. "Got You!": Automatic Vandalism Detection in Wikipedia with Web-based Shallow Syntactic-Semantic Modeling. In *COLING*. Tsinghua University Press, 1146–1154.

[57] X. Wang, M. Bendersky, D. Metzler, and M. Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *SIGIR*. ACM, 115–124.

[58] Z. Wang, J. Zhang, J. Feng, and Z. Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. AAAI Press, 1112–1119.

[59] C. Wilkie and L. Azzopardi. 2017. Algorithmic Bias: Do Good Systems Make Relevant Documents More Retrievable?. In *CIKM*. ACM, 2375–2378.

[60] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. 2014. Toward Computational Fact-Checking. *PVLDB* 7, 7 (2014), 589–600.

[61] K. Yang and J. Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM*. ACM, 22:1–22:6.

[62] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *WWW*. ACM, 1171–1180.

[63] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. 2013. Learning Fair Representations. In *ICML (3) (JMLR Workshop and Conference Proceedings)*, Vol. 28. JMLR.org, 325–333.

[64] L. Zhang and X. Wu. 2017. Anti-discrimination learning: a causal modeling-based framework. *I. J. Data Science and Analytics* 4, 1 (2017), 1–16.

[65] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K. Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *EMNLP*. ACL, 2979–2989.

[66] I. Zliobaite. 2015. On the relation between accuracy and fairness in binary classification. *CoRR* abs/1505.05723 (2015).

[67] I. Zliobaite. 2017. Measuring discrimination in algorithmic decision making. *Data Min. Knowl. Discov.* 31, 4 (2017), 1060–1089.