Investigating Annotator Bias with a Graph-Based Approach

Maximilian Wich
TU Munich,
Department of Informatics,
Germany
maximilian.wich@tum.de

Hala Al Kuwatly
TU Munich,
Department of Informatics,
Germany
hala.kuwatly@tum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

A challenge that many online platforms face is hate speech or any other form of online abuse. To cope with this, hate speech detection systems are developed based on machine learning to reduce manual work for monitoring these platforms. Unfortunately, machine learning is vulnerable to unintended bias in training data, which could have severe consequences, such as a decrease in classification performance or unfair behavior (e.g., discriminating minorities). In the scope of this study, we want to investigate annotator bias — a form of bias that annotators cause due to different knowledge in regards to the task and their subjective perception. Our goal is to identify annotation bias based on similarities in the annotation behavior from annotators. To do so, we build a graph based on the annotations from the different annotators, apply a community detection algorithm to group the annotators, and train for each group classifiers whose performances we compare. By doing so, we are able to identify annotator bias within a data set. The proposed method and collected insights can contribute to developing fairer and more reliable hate speech classification models.

1 Introduction

A massive problem that online platforms face nowadays is online abuse (e.g., hate speech against women, Muslims, or African Americans). It is a severe issue for our society because it can cause more than poisoning the platform's atmosphere. For example, Williams et al. (2020) showed a relation between online hate and physical crime.

Therefore, people have started to develop systems to automatically detect hate speech or abusive language. The advances in machine learning and deep learning have improved these systems tremendously, but there is still much space for enhancements because it is a challenging and complex task (Fortuna and Nunes, 2018; Schmidt and Wiegand, 2017).

A weakness of these systems is their vulnerability towards unintended bias that can cause an unfair behavior of the systems (e.g., discrimination of minorities) (Dixon et al., 2018; Vidgen et al., 2019). Researchers have identified different types and sources of bias that can influence the performance of hate speech detection models. Davidson et al. (2019), for example, investigated racial bias in hate speech data sets. Wiegand et al. (2019) showed that topic bias and author bias of data sets could impair the performance of hate speech classifiers. Wich et al. (2020) examined the impact of political bias within the data on the classifier's performance. To mitigate bias in training data, Dixon et al. (2018) and Borkan et al. (2019) developed an approach.

Another type of bias that caught researchers' attention is annotator bias. It is caused by the subjective perception and different knowledge levels of annotators regarding the annotation task (Ross et al., 2017; Waseem, 2016; Geva et al., 2019). Such a bias could harm the generalizability of classification models (Geva et al., 2019). Especially in the context of online abuse and hate speech, it can be a severe issue because annotating abusive language requires expert knowledge due to the vagueness of the task (Ross et al., 2017; Waseem, 2016). Nevertheless, due to the limited resources and the demand for large datasets, annotating is often outsourced to crowdsourcing platforms (Vidgen and Derczynski, 2020). Therefore, we want to investigate this phenomenon in our paper. There is already research concerning annotator bias in hate speech and online abuse detection. Ross et al. (2017) examined the relevance of instructing annotators for hate speech annotations. Waseem (2016) compared the impact of amateur and expert annotators. One of their findings was that a system trained with data

Proceedings of the Fourth Workshop on Online Abuse and Harms, pages 191–199
Online, November 20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

labeled by experts outperforms one trained with data labeled by amateurs. Binns et al. (2017) investigated whether there is a performance difference between classifiers trained on data labeled by males and females. Al Kuwatly et al. (2020) extended this approach and investigated the relevance of annotators' educational background, age, and mother tongue in the context of bias. Sap et al. (2019) examined racial bias in hate speech data sets and its impact on the classification performance. To the best of our knowledge, no one has investigated annotator bias by identifying patterns in the annotation behavior through an unsupervised approach. That is why we address the following research question in the paper: Is it possible to identify annotator bias purely on the annotation behavior using graphs and classification models?

Our contribution is the following:

- A novel approach for grouping annotators according to their annotations behavior through graphs and analyzing the different groups in order to identify annotator bias.
- A comparison of different weight functions for constructing the annotator graph modeling the annotator behavior.

2 Data

For our study, we use the Personal Attacks corpora from the Wikipedia Detox project (Wulczyn et al., 2017). It contains 115,864 comments from English Wikipedia that were labeled whether they comprise personal attack or not. In total, there are 1,365,217 annotations provided by 4,053 annotators from the crowdsourcing platform Crowdflower — approximately 10 annotations for each comment. Each annotation consists of 5 categories distinguishing between different types of attack: *quoting_attack*, *recipient_attack*, *third_party_attack*, *other_attack*, and *attack*. In our experiments, we only use the 5th category (*attack*) because it covers a broader range than the other labels. Its value is 1 if "the comment contains any form of personal attack" (Wikimedia, n.d.). Otherwise it is 0. The corpora also contain demographic information (e.g., gender, age, and education) of 2,190 annotators. But this data is not relevant to our study.

3 Methodology

Our approach is to group annotators according to their annotation behavior and analyze performance of classification models trained on annotations from these groups. To do so, we firstly group the annotators according to their annotation behavior using a graph. Secondly, we split the data set by the groups and their respective annotations. Thirdly, we train classifiers for each annotator group and then compare their performances. The reader can find a detailed description of the steps in the following[1]:

Creating Annotator Graph

In the first step, we create an undirected unweighted graph to model the annotation behavior of the annotators (e.g., how similar the annotations of two annotators are). Each node represents an annotator. An edge between two nodes exists if both annotators annotate at least one same data record. Additionally, each edge has a weight that models the similarity between the annotations of the data records. To calculate the weight, we selected four functions that we will compare:

1. **Agreement Rate**: It is the percentage in which both annotators agree on the annotation for a data record:

$$a = \frac{n_{agree}}{n_{agree} + n_{disagree}}$$

where n_{agree} is the number of data records that both annotated and assigned the same labels to and $n_{disagree}$ is the number of data records that both annotated and assigned different labels.

2. **Cohen's Kappa** (Cohen, 1960): It is often used as a measure for inter-rater reliability.

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

where p_0 is the "proportion of observed agreements" (Sim and Wright, 2005, p.258) among the data records annotated by both annotators and p_e is "proportion of agreements expected by chance" (Sim and Wright, 2005, p.258) among the records. The range of κ is between -1 and $+1$. $+1$ corresponds perfect agreement; ≤ 0 means agreement at chance or no agreement (Cohen, 1960). If both annotators select the same label for all records, κ is not

[1]Code available on GitHub: `https://github.com/mawic/graph-based-method-annotator-bias`

defined. In this case, we remove the edge. An alternative would be to keep the edge and assign 1. But we rejected this idea because of the following consideration. Let us assume that we have 4 annotators (A,B,C, and D). A and B assigned the same label to the same comment. C and D assigned the same labels to the same 20 comments. In both cases, κ is not defined. Assigning the same value (e.g., 1) to both edges would weigh both equally. But the edge between C and D should receive a higher weight because the agreement between A and B could be a coincidence.

3. **Krippendorff's Alpha** (Krippendorff, 2004): It is another inter-rater reliability measure, which is defined as follows:

$$\alpha = 1 - \frac{D_0}{D_e}$$

"where D_0 is the observed disagreement among values assigned to units of analysis [...] and D_e is the disagreement one would expect when the coding of units is attributable to chance rather than to the properties of these units" (Krippendorff, 2011, p.1). Further details of the calculation are provided by Krippendorff (2011). Similar to κ, α is not defined if the annotators choose the same label for all records. We handle this case in the same way as above.

4. **Heuristic**: To overcome the undefined issue, we define a heuristic weight function taking the relative agreement rate and the number of commonly annotated data records (overlap) between two annotators into account. The function is defined by four boundary points:

   - The maximum weight (1.0) is reached, if two annotators commonly annotated n data records and agree on all annotations. n is the maximal number of data records that is commonly annotated by two annotators and is defined by the data set.
   - The minimum weight (0) is reached, if two annotators commonly annotated n data records and disagree on all annotations.
   - A weight that is 20% larger than the minimum weight (0.2) is reached, if two annotators commonly annotated only one data record and disagree.
   - A weight that is 60% larger than the minimum weight (0.6) is reached, if two annotators commonly annotated only one data record and agree.

   The transition between the four boundary points is gradually calculated. The algorithm can be found in the appendix. The purpose of the approach is to consider the overlap besides the agreement rate because the larger the overlap the more reliable is the agreement rate. Cohen's Alpha and Krippendorff's Alpha provide this, but their weakness is the undefined issue, which is a realistic scenario for our annotation task.

All weight functions are normalized between 0 and 1 to make the results comparable, if they are not already in this range.

Detecting Annotator Groups

The goal of the next step is to group the annotators according to their annotation behavior. For this purpose, we apply the Louvain method, an unsupervised algorithm for detecting communities in a graph (Blondel et al., 2008). After that, we filter the communities with at least 250 members. Otherwise, the groups do not comprise enough data records that were annotated by their members in order to train a classification model.

Splitting Data According to Groups

After detecting the groups, we split the comments and annotations according to the groups. For each weight function and the corresponding graph, we do the following: We select those comments that were annotated by at least one member of every group. For each group, we create a data set containing these comments and the annotations from the group's members. The label for each comment is the majority vote of the group's annotators. In addition, we create a further data set that serves as a baseline and is called group 0 for all experiments. The data set contains the same comments, but the labels are the results of all 4,053 annotators. After that, all data sets for a weight function are split in a training and test set in the same manner to ensure the comparability of the data sets. This is done for each of the four weight functions.

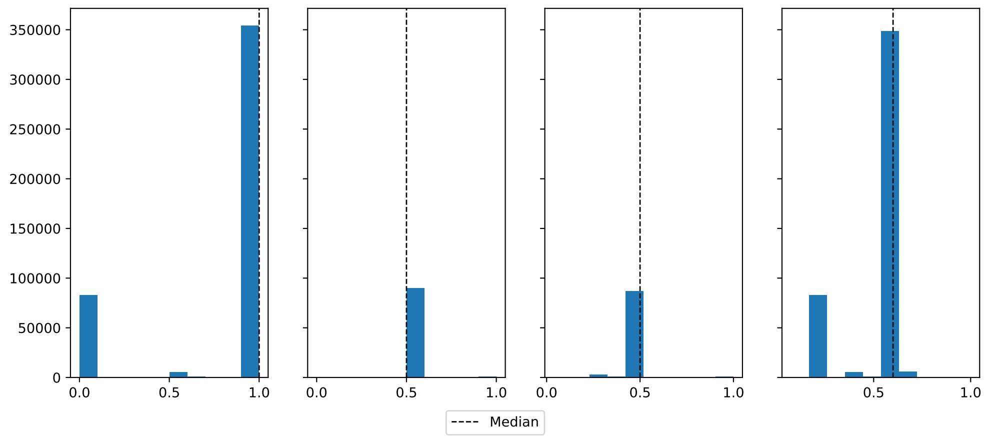
Weight function	Agreement Rate	Cohen's Kappa	Krippendorff's Alpha	Heuristic Function
Number of nodes	4,053	4,053	4,053	4,053
Number of edges	444,344	91,308	91,308	444,344
Average degree	219.3	45.1	45.1	219.3
Density	0.054	0.011	0.011	0.054
Connected components	1	1	1	1
Distribution of edge weights				

Table 1: Graph metrics

Training Classification Models for Groups and Comparing Their Performances

For the classification model, we use a pre-trained DistilBERT that we fine-tune for our task (Sanh et al., 2019). It is smaller and faster to train than classical BERT, but it provides comparable performance (Sanh et al., 2019). In the context of abusive language detection, it shows a similar performance like larger BERT models (Vidgen et al., 2020). Since we need to train several models for different weight functions and groups, we choose the lighter model.

The basis of our classification model is the pre-trained `distilbert-base-uncased`, which is the distilled version version of `bert-base-uncased`. It has 6 layers, a hidden size of 768, 12 self-attention heads, and 66M parameters. To fine-tune the model for our task, we apply the 1cycle learning rate policy suggested by Smith (2018) with a learning rate of 5e-6 for 2 epochs. The batch size is 64 and the size of the validation set is 10% of the training set. Furthermore, we limit the number of input tokens to 150. The task that DistilBERT is fine-tuned for is to distinguish between the labels ”ATTACK” and ”OTHER”.

After training the models, we compare their performances (F1 macro). For this purpose, each model is evaluated on its own test set and the one from the other groups including group 0, which represents all annotators. Instead of reporting the F1 score, we report them relatively to our baseline (group 0) because it allows a better comparison of the results. Additionally, the actual F1 score are not relevant for this analysis.

4 Results

The experiments show that our proposed method enables the grouping of annotators according to similar annotation behavior. Classifiers separately trained on data from the different groups and evaluated with the other groups' test data exhibit noticeable differences in classification performance, which confirms our approach. The detailed results can be found in the following:

Annotator Graph

We created one graph for each weight function. Table 1 provides the key metrics of the generated graphs. It is conspicuous that the graphs with Cohen's Kappa and Krippendorff's Alpha weight function have only 91,308 edges, while the other twos have 444,344. This difference also causes the divergence of the average degree and density. The reason for the difference is that many relations between two annotators comprise only one comment. If both agree on an annotation, Cohen's Kappa and Krippendorff's Alpha are not defined; consequently, we do not have an edge. Therefore, graphs

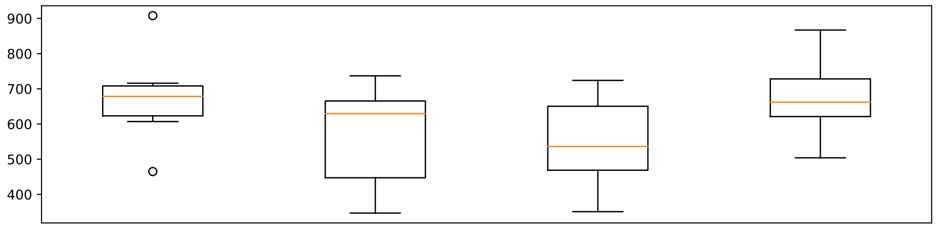
Weight function	Agreement Rate	Cohen's Kappa	Krippendorff's Alpha	Heuristic Function
Number of identified groups	6	13	12	6
Number of selected groups	6	6	6	6
Number of annotators	4,053	3,407	3,282	4,053
AVG(annotators/groups)	579.00	486.71	468.86	579.00
SD(annotators/groups)	266.17	239.75	224.99	258.24
Size of training set/test set	31,170 / 7,793	18,951 / 4,738	18,204 / 4,551	31,738 / 7,934
Distribution of group sizes				

Table 2: Results of community detection

with these weight functions have fewer edges.

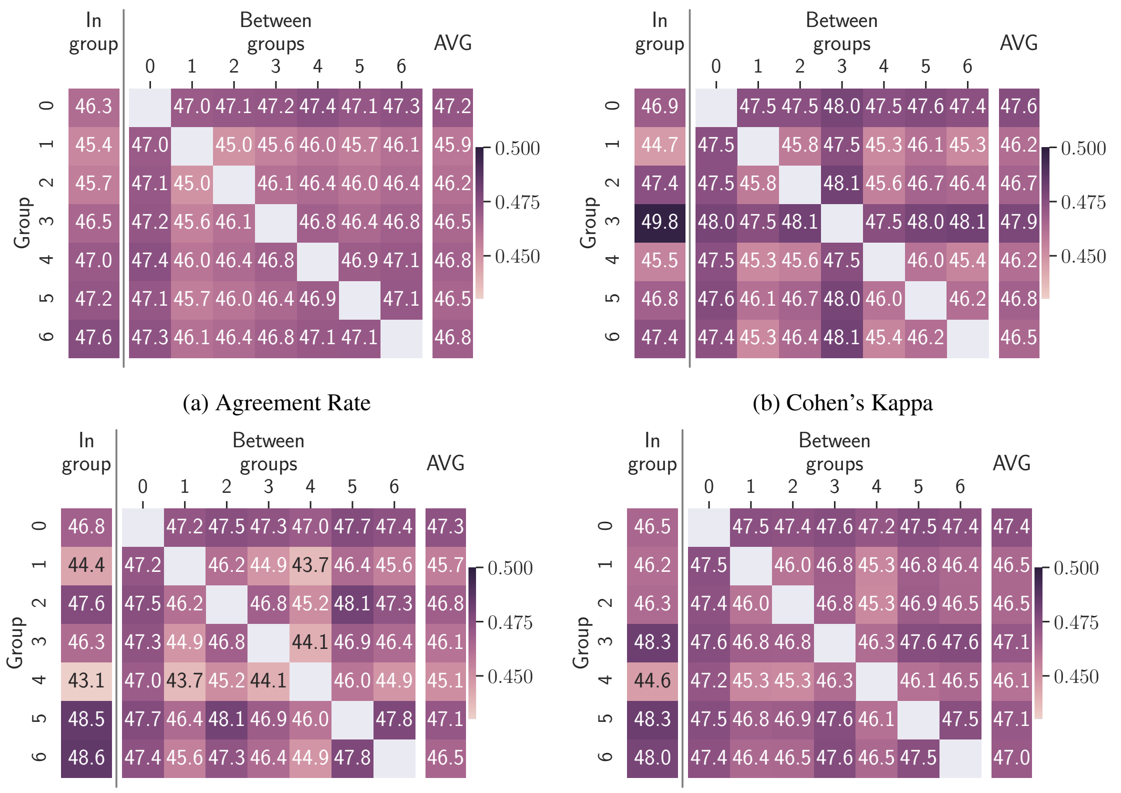
Community Detection

Table 2 shows the results of community detection. While the Louvain algorithm split the graphs with the Agreement Rate and Heuristic Function as weight functions in 6 groups, 13 groups in the graph with Cohen's Kappa were detected and 12 in the one with Krippendorff's Alpha. An explanation for the divergence is the difference between the number of edges of the graphs. Since the groups have various numbers of members, we select only these with at least 250 annotators due to two reasons. Firstly, we have the same number of groups for all weight functions. Secondly, we ensure that we have enough annotated comments to train the classifiers. It may be noted at this juncture that only comments were selected for the training/test set if they were annotated by the group. Therefore, groups with a small number of annotators would have reduced the size of the training/test set. The distribution of the size of the training/test set is similar to the one of the numbers of identified groups. For Agreement Rate we have 31,170 annotated comments for the training set and 7,793 for the test set, for the Heuristic Function 31,738 and 7,934, for the Cohen's Kappa 18,951 and 4,738, and for Krippendorff's Alpha 18,204 and 4,551. The smaller data sizes for the last two are related to the smaller average size of groups.

To compare the different groups, we computed the inter-rater agreement for each group and between the groups by using Krippendorff's Alpha. To calculate the rate between the groups, we compute Krippendorff's Alpha using the union of all annotations from both groups. The inter-rater agreement scores (in percent, 100% means perfect agreement) for all four weight functions are depicted in Figure 1. The first column of each subfigure shows the inter-rater agreement within each group. The 7 columns right to the line provide the inter-rater agreement between the groups, and the last column shows the average inter-rater agreement between the groups. Please note that the inter-rater agreement scores are not comparable between the different weight functions/subfigures because the groups, the comments, and the annotations are different. The scores can only be compared with scores from the same graph with the same weight function.

If we look at the inter-rater agreement within the groups (first column of each subfigure), we see that the groups exhibit varying scores and that the deviations to the baseline (group 0, data set average) also differ. If the score is higher than the baseline, the group is more coherent in regards to the annotations. If it is lower, the group is less coherent. Furthermore, the more scores are higher than the baseline, the better because it means that the algorithm is able to create more coherent groups. Considering these aspects, we can say that the Heuristic Function produces the best results. Its groups 3 (48.3%), 5 (48.3%), and 6 (48.0%) together have the largest distance to the baseline (46.5%) than the top three groups of the other groups.

The groups resulting from the graph with the Agreement Rate as weight function have less strongly varying inter-rater agreement rates than the groups of the other weight functions. In the case of Cohen's Kappa, one group with a strong inter-rater agreement is formed (49.8%) — the



(a) Agreement Rate

(b) Cohen's Kappa

(c) Krippendorff's Alpha

(d) Heuristic Function

Figure 1: Inter-rater agreement within and between groups for different weight functions

highest deviation for all weight functions. But the other groups are similar to the baseline or worse. Krippendorf's Alpha as weight function produces results that are comparable to the ones from the Heuristic Function. The deviations from its top three groups, however, are smaller than the ones from the corresponding groups of Heuristic Function. Furthermore, the groups of the Heuristic Functions cover all 4,053 annotators, while the ones from Krippendorf's Alpha comprise only 3,282. Therefore, we choose the groups from the Heuristic Function for the last part of the experiment.

Classification Models and Their Performances

Instead of reporting the macro-F1 scores for the classifiers trained on the different group-specific training sets and tested on the all group-specific test sets, we report them relatively to the baseline (trained on group 0 and tested on group 0) for easier comparison. The baseline has a macro-F1 score of 87.54%. In addition to the relative scores, the figures contain an extra column and row with average values for better comparability.

It is conspicuous that the deviations reported in the first column of each matrix are lower than the rest. The reason is the following: These columns report the performances of the classifiers for the different groups on the baseline test set. Since the baseline test set has the largest number of annotations, the labels are more coherent. Consequently, classifiers perform better on the baseline test set than on their own, less coherent test sets.

The first thing that attracts our attention is column 4 because it has the largest deviations, meaning that all classifiers perform quite worse on the test set of group 4. We can explain this phenomenon with the low inter-rater agreement rate of this group (44.6%, compare Figure 1d). This is also the explanation of why row 4 has the lowest average of all rows. In this context, it is surprising that row 6 has the second-lowest average of all rows, while it has a relatively large inter-rater agreement rate (48.0%). A possible explanation can be that the annotations within the group are coherent but less coherent with respect to all other annotations. In the case of group 3 and 5 (inter-rater agreement rate of 48.3% and 48.%), the average deviations are comparable to the others. This indicate that the annotation behavior of the group members is more coherent with the overall annotation behavior.

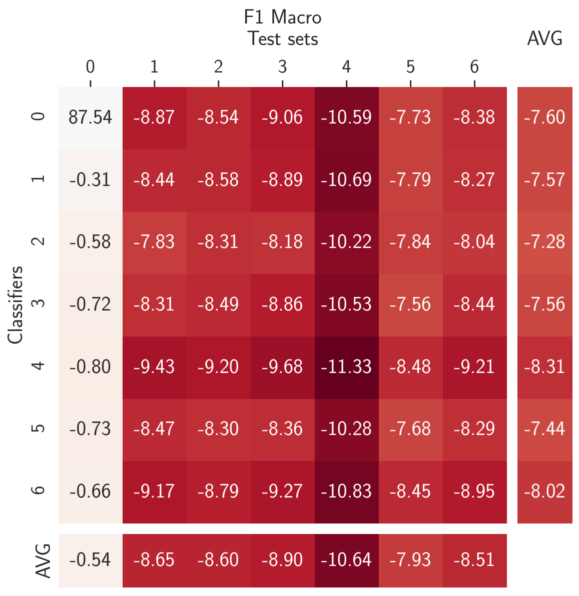


Figure 2: Macro F1 scores for the Heuristic Function

The cross-group agreement average (last column in Figure 1d) confirms this because both groups have the largest values aside from the baseline. In the case of group 5, this hypothesis is supported by the fact that all classifiers perform on average better on the test set of group 5 than on the test sets from groups 1, 2, 3, 4, and 6.

5 Discussion

The results show that the proposed method is suitable for identifying annotator groups purely based on annotation behavior. The deviations in inter-rater agreement rates of the groups and in the classifiers' performances prove this.

In regards to the weight functions, we found that the Agreement Rate is not suitable compared to the other functions. This is not surprising because there is a reason why Cohen's Kappa and Krippendorff's Alpha are used as a metric for the inter-rater agreement. An advantage of our Heuristic Function in regards to Cohen's Kappa and Krippendorff's Alpha as weight functions is that it does not have the undefined issue if two annotators assign only one type of label to the comments to be labeled. A potential improvement could be to combine Cohen's Kappa and Krippendorff's Alpha weight function with the Heuristic Function.

The results of our method can be linked to annotator bias in the following manner: An identified annotator group that has a high inter-rater agreement within the group, but poor classification performance on the other test sets indicates that it has a certain degree of bias as the group's annotation behavior differs from the rest. For such insights, we see currently two possible use cases:

- The insights can be used to mitigate annotator bias. The annotations of these groups can either be weighted differently or deleted to avoid transferring the bias to the classification model.
- The insights can be used to build classification models that model the annotator bias. This can be helpful for tasks that do not have one truth. In the case of online abuse, it is possible that one group is more tolerant towards abusive language and another one less tolerant.

The novelty of our approach is that it is unsupervised and does not require any stipulation of bias that you want to detect in advance. Existing approaches, such as Binns et al. (2017), who investigated gender bias, or Sap et al. (2019) and Davidson et al. (2019), who examined racial bias, defined in their hypothesis which kind of bias they want to uncover. Our method, however, does not require any pre-defined categories to detect bias.

6 Conclusion

In this paper, we proposed a novel graph-based method for identifying annotator bias through grouping similar annotation behavior. It differs from existing approaches by its unsupervised nature. But the method requires further research and refinement. To address our limitations, we propose the following future work:

Firstly, we used only one data set for our study. The approach, however, should be also tested and refined with other data sets. The Wikipedia Detox project, for example, provides two more data sets with the same structure, but with different tasks (toxicity and aggression). In general, data availability is a challenge of this kind of research because hate speech data sets mostly contain aggregated annotations. Therefore, we urge researchers releasing data sets to provide the unaggregated annotations as well.

Secondly, other approaches for grouping the annotators should be investigated. We used only one community detection method, the Louvain algorithm. But there are many more methods, such as the Girvan-Newman algorithm (Girvan and Newman, 2002) and the Clauset-Newman-Moore algorithm (Clauset et al., 2004).

Thirdly, our methods should be extended so that it can handle smaller groups. Our current approach requires at least 250 annotators in a group to ensure that we have enough training data. But it would be interesting to investigate smaller groups in the hope that these groups are more coherent in regards to their annotation behavior.

Acknowledgments

This research has been partially funded by a scholarship from the Hanns Seidel Foundation financed by the German Federal Ministry of Education and Research.

References

Hala Al Kuwatly, Maximilian Wich, and Georg Groh. 2020. Identifying and measuring annotator bias based on annotators' demographic characteristics. In *Proc. 4th Workshop on Online Abuse and Harms*.

Reuben Binns, Michael Veale, Max Van Kleek, and Nigel Shadbolt. 2017. Like trainer, like bot? inheritance of bias in algorithmic content moderation. In *International conference on social informatics*, pages 405–415. Springer.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Proc. 28th WWW Conf.*, pages 491–500.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. *arXiv preprint arXiv:1905.12516*.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 1161–1166.

Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

K. Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Content Analysis: An Introduction to Its Methodology. Sage.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proc. 57th ACL Conf.*, pages 1668–1678.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Julius Sim and Chris C Wright. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3):257–268.

Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Bertie Vidgen, Austin Botelho, David Broniatowski, Ella Guest, Matthew Hall, Helen Margetts, Rebekah Tromble, Zeerak Waseem, and Scott Hale. 2020. Detecting east asian prejudice on social media.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data: Garbage in, garbage out. *arXiv preprint arXiv:2004.01670*.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.

Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proc. First Workshop on NLP and Computational Social Science*, pages 138–142.

Maximilian Wich, Jan Bauer, and Georg Groh. 2020. Impact of politically biased data on hate speech classification. In *Proc. 4th Workshop on Online Abuse and Harms*.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 2019. Detection of Abusive Language: the Problem of Biased Datasets. *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 602–608.

Wikimedia. n.d. Research:detox/data release. `https://meta.wikimedia.org/wiki/Research:Detox/Data_Release`.

Matthew L Williams, Pete Burnap, Amir Javed, Han Liu, and Sefa Ozalp. 2020. Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. *The British Journal of Criminology*, 60(1):93–117.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399.