Bot Detection in Wikidata Using Behavioral and Other Informal Cues

ANDREW HALL, GroupLens Research, University of Minnesota – Twin Cities, USA
LOREN TERVEEN, GroupLens Research, University of Minnesota – Twin Cities, USA
AARON HALFAKER, Wikimedia Research, Wikimedia Foundation, USA

Bots have been important to peer production’s success. Wikipedia, OpenStreetMap, and Wikidata all have taken advantage of automation to perform work at a rate and scale exceeding that of human contributors. Understanding the ways in which humans and bots behave in these communities is an important topic, and one that relies on accurate bot recognition. Yet, in many cases, bot activities are not explicitly flagged and could be mistaken for human contributions. We develop a machine classifier to detect previously unidentified bots using implicit behavioral and other informal editing characteristics. We show that this method yields a high level of fitness under both formal evaluation (PR-AUC: 0.845, ROC-AUC: 0.985) and a qualitative analysis of “anonymous” contributor edit sessions. We also show that, in some cases, unflagged bot activities can significantly misrepresent human behavior in analyses. Our model has the potential to support future research and community patrolling activities.

CCS Concepts:
- Human-centered computing → Human computer interaction (HCI)

KEYWORDS
Wikidata; Peer production; Structured data; Machine learning; Bots; Automation

ACM Reference format:

1 INTRODUCTION

The week-long period from April 22 to April 29, 2016 seems to represent an historic peak of human productivity. At around noon GMT on April 22, someone began making edits to Wikidata, a structured data peer production community and sister project to Wikipedia. Rather than choosing to have a relaxing Friday afternoon and evening, this dedicated contributor got caught up in the editing spirit and pushed on – editing late into the night. They performed their edits with incredible consistency, on average making an edit every 30 seconds. Amazingly, this contributor did not stop at all that night – nor the next, nor the next.... In fact, they continued working for 7 straight days, finally deciding to take a break over 21,000 edits later. What’s more, they made all their edits anonymously, without even logging in to a Wikidata contributor account.

This work is supported by the University of Minnesota and indirectly by the Wikimedia Foundation (Aaron’s employer).
Author’s addresses: Andrew Hall, Loren Terveen: Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN, 55455, USA; Aaron Halfaker: Wikimedia Foundation, Inc., 1 Montgomery Street #1600, San Francisco, CA, 94104, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
Copyright 2018 held by Owner/Author(s). 2573-0142/2018/11 – ART64 https://doi.org/10.1145/3274333

Proceedings of the ACM on Human-Computer Interaction, Vol. 2, No. CSCW, Article 64, Publication date: November 2018.
It's unlikely that a single human could contribute at this pace and scale. Indeed, this editing behavior looks suspiciously robotic. A robot or “bot” in the context of peer production is defined as “an autonomous software program which is developed and operated by volunteers.” [2]. As in Wikipedia, Wikidata bots can do more work than humans, and can do so faster and with greater consistency. Bot editing is particularly widely used in Wikidata (88% of all edits [18]). This is in part due to the relative ease with which its structured (key-value pair) content can be imported from external data sources. Community rules require bot operations to occur via Wikidata accounts approved to run with a “bot flag”. But obviously, the prolific anonymous contributor above was not using a bot flagged account – they were anonymous! Was this contributor really superhuman? Or was someone operating a bot against community policy? Highly accurate bot detection in peer production is important for several reasons. (1) Inaccuracies in bot identification could invalidate analyses of peer production that attempt to distinguish the role of bot and human edits. If studies of human contributors inadvertently use datasets containing a large amount of bot activity, results supposedly about human behavior could instead largely reflect bot behavior. Since bot edit rates often are much higher than human edit rates, just a few bot activity sessions mistakenly labeled as human activity could dramatically skew the results of some analyses. For example, if the dataset used for the seminal conclusion that “Wikipedians are born, not made” [14] contained bot edits (something that the authors noted explicitly that they wanted to avoid), it could turn out that bots are “born” and human contributors are “made”. Further, for work studying bot contributions (e.g., [4,15,21]), it is important to include as many bot contributions as possible to provide a complete picture of the role bots play in the community. (2) Highly accurate bot detection is important since the Wikidata community is concerned with detecting all bot activities. Even just one undetected bot that is producing thousands of edits has the potential to do large amounts of damage to the community. To avoid damaging scenarios, mandatory bot approval policies have been put in place in both Wikidata and Wikipedia [27,30].

Concerns about unidentified bot activity are not merely theoretical. While not a topic of extensive analysis, previous work in Wikipedia and OpenStreetMap has identified behavioral anomalies as resulting from unidentified bots [5]. Further, the authors of this paper have personally come across unidentified bots during data explorations. Thus, while unidentified bots occur with certainty, the prevalence of such edits remains to be seen.

To facilitate both peer production research and Wikidata community policy enforcement, our work offers the following contributions.

1.1 Contributions

- We describe an effective bot detection strategy using machine classification and implicit behavioral and other informal editing characteristics and show that it is effective in identifying unflagged bot activity in Wikidata.

- Using our model, we show that, for the most part, unflagged bot activities are rare. Hence, the activities of unflagged bots appear unlikely to significantly change results in many studies of human and bot behavior. However, analyses that are sensitive to outliers (e.g. max session

---

1 [https://www.wikidata.org/wiki/Wikidata:Bots](https://www.wikidata.org/wiki/Wikidata:Bots)

2 For example, we have identified anonymous edits coming from the Wikimedia internal network (Wikimedia operates Wikipedia). The IP addresses associated with these edits had a prefix of “10.86.” Such internal servers are not equipped to support a GUI/browser and thus the edits almost certainly came from bots.
duration) or that span relatively short subsets of Wikidata history should first extract previously unidentified bots from “human” contributions to ensure accurate analyses.

- We show that there is a meaningful amount of non-compliance with the bot policy in Wikidata. According to our model, 3% of registered “human” user edits overall and 2% of anonymous “human” user edits overall are from bots. These percentages are important since all bot activity that does not align with community policy matters to the Wikidata community.

- We make our datasets and bot detection code available under an open license: https://github.com/hall1467/wikidata_bot_prediction_model. This will facilitate future Wikidata contributor behavior research as well as better allow the Wikidata community to enforce its bot policies.

2 BACKGROUND AND RELATED WORK

Peer production communities such as Wikipedia, OpenStreetMap, and Wikidata all provide widely used data. In fact, Wikipedia is the world’s 5th most visited website3. On top of direct use by end users (e.g. reading Wikipedia articles), peer produced data is used in dozens of applications such as Google Search and MapBox, and also by various algorithms, e.g. [12].

The success of Wikipedia in particular has helped spur the development of its sister project Wikidata, which produces structured data. Wikidata’s key-value pairs currently describe more than 40 million entities ranging from all types of concrete real-world things (e.g. “Earth”, “iPhone”) to abstract concepts (e.g. “Life”). For instance, the entity (or item as named in Wikidata) for Barack Obama contains key-value (or Property-value) information about his “date of birth” (“4 August 1961”), “spouse” (“Michelle Obama”), “occupation” (“politician”), and dozens of other facts about him. Rather than being directly consumed by end-users as in Wikipedia, Wikidata’s data is intended specifically for use by applications. Google Search uses Wikidata with Google Knowledge Graph; Apple Siri and Wikipedia’s infoboxes are two other examples of consumers.

2.1 Bot Use in Peer Production

Bots contribute to peer production in many ways. Halfaker and Riedl [7] called Wikipedia bots ‘force multipliers’ for people; they do some of the same things as human editors, but on a larger scale (e.g. for imports), and they also “[protect] the encyclopedia from malicious activity” (e.g. reverting vandalism). Bots are prominent players in the editing ecosystem, so there has been much debate about the roles appropriate for bots. For example, bots can be enforce policy by editing community discussions to include member signatures [1]. However, this use of bots resulted in community pushback against bots correcting human behavior. Skepticism towards bots is justified given some unfortunate experiences with them in peer production. A notable example occurred in OpenStreetMap, with a large-scale import of US TIGER4 geographic data [31]. This import provided basic map features like highways and rivers as a foundation for OpenStreetMap’s US map to build upon [31]. However, the import is notorious as a questionable use of automation since it introduced large amounts of “outdated and erroneous” data [24]. Instead of saving human editors years of manual work, the import has taken years of work to fix

3 https://www.alexa.com/topsites
4 https://www.census.gov/geo/maps-data/data/tiger.html
[32]. A similar problem occurred in Wikipedia with rambot imports of geographic data to thousands of articles on cities [26].

Due to the potential of bots to cause harm, communities have built bureaucratic infrastructures to govern bot deployment. English Wikipedia developed the Bot Approvals Group (BAG) [27] to “[oversee] most areas and processes dealing with bots” [25]. Bots cannot run without BAG approval [27]. Wikidata has adopted a similar approvals process. OpenStreetMap has a policy called “mapping for locals” [8] intended to ensure automated imports are used only if approved by contributors in the geographic region the imports affect [33]. These approval processes are burdensome, so some contributors seek to circumvent them. To the extent they succeed, they violate policy and hide their bot activities from analysis.

Note that the level of automation used in peer production contributions is on a spectrum; thus, while “human edits” ultimately involve manual human work, they often also have a degree of automation. This includes edits from semi-automated tools, which may produce thousands of edits without human input (e.g. AutoWikiBrowser in Wikipedia [29]). Given the motivations for identifying unflagged bots discussed above, we focus here on identifying unflagged bot edits; however, developing techniques to measure the effects of semi-automated edits is a valuable direction of future work.

2.2 Bot Detection

Current bot detection methods used in peer production rely on explicit signals (e.g. flagged bot accounts). Explicit signaling is common in communities such as Wikidata and Wikipedia since these communities are highly regulated and so tend to have relevant policies. Despite a long history of bot research, the methods used in many influential Wikipedia studies (e.g. [3, 6, 11, 14]) are not always straightforward to apply and still miss some bots. Geiger and Halfaker [4] noted that “...getting a list of bot accounts has long been a challenge for Wikipedia researchers.” As mentioned, Wikipedia and Wikidata policies require users to gain approval to run bots. This occurs through a separate “bot flagged” account. The communities maintain tables of both current and former approved bot accounts. Geiger and Halfaker state that a widely-used method of identifying bots in Wikipedia is to consider user accounts that are currently flagged as bots (some accounts will be de-flagged when they are inactive). One other technique mentioned is to find usernames with “bot” in them. These techniques have been applied in Wikidata research [13, 15, 18] and may have missed bots/bot edits for several specific reasons. First, contributors may forget to switch to their “bot account” to run bots, or even forget to log in at all, instead editing anonymously. Second, if a user wishes to avoid going through the bot approval process, they may secretly run their bot anyway, as an anonymous contributor or through their personal account. Third, Wikidata did not have an effective bot policy for the first 14 months of its existence. And that policy undoubtedly was not consistently followed by all bot maintainers immediately after it was instituted.

In some contexts outside peer production – such as malware detection, video games, Twitter, etc. – it is not possible to recognize bots using explicit techniques. These contexts are often less regulated, and bots deployed in them will not identify themselves since their goal is precisely to avoid detection. Thus, implicit bot detection signals must be used in these contexts. Fortunately,

---

bot behavior patterns often are quite distinct from human patterns, and bot detection techniques (e.g. [9,10,19,20]) can take advantage of implicit behavioral signals such as repetition.

2.3 Activity Session Behavioral Patterns

Because of the limitations of peer production bot recognition approaches using explicit signals/labels, we took insight from bot recognition techniques developed in contexts where only implicit behavioral signals are available. Similar behavioral patterns to those identified in the previous section have occurred in peer production. Specifically, implicit characteristics of human edit "sessions" have been shown to follow predictable patterns [3,5]. In the context of such work, an edit session is a contiguous series of edits performed by a user without a substantial break. This work showed that people follow approximately consistent distributions of inter-edit and between-session times, and that bot activity deviates from these human patterns.

Analyzing user contributions at the session-level makes sense for two additional reasons as well. First, some registered users will make most of their edits manually but occasionally decide to run a bot though their own account for a short time. Detecting bots at the account level would not detect this behavior since (presumably) most of such users’ activity would follow human-like patterns. Second, bot detection at the account-level cannot be done for anonymous user contributions. Anonymous contributions are also known as “logged-out” or “IP address” contributions, since only the IP address is logged rather than a username/user id. IP addresses are often dynamically allocated. Hence, an IP address could correspond with a certain user on one day and a different user on the next. While anonymous contributions compose a small percentage of contributions in peer production [18] -- less than 1% in our Wikidata dataset -- work has indicated that anonymous contributions in Wikipedia are disproportionately valuable to the community [34,35]. Thus, important questions remain to be answered in Wikidata regarding the roles of anonymous contributors and the value they add.

3 METHODS

We next discuss the steps we followed to build our bot prediction model. In-depth details of the generation of training, testing, and other necessary datasets used can be found in Appendix A. Further, while not required to understand the discussions in this paper, the reader may wish to review information on Wikidata terminology at [36].

We built our model using scikit-learn. We tried two different models: a random forest classifier and a gradient boosting classifier. We expected these ensemble models to be effective at making predictions since they have proved effective in similar contexts such as Wikipedia vandalism detection [17]. We applied hyper-parameter optimization for both models, optimizing initially for ROC-AUC and PR-AUC. Three-fold cross validation was used, which is scikit-learn’s default. ROC-AUC has been a standard machine learning metric [16]. However, our training dataset was heavily skewed towards human sessions. In such cases, PR-AUC, a metric commonly used in information retrieval when skewness is a concern, is preferred [16]. We chose the gradient boosting classifier because it had a higher PR-AUC on test data (0.528 versus 0.486).

All features used in our model were based on “informal” characteristics of edits and editing sessions. Such “informal” characteristics were not based on the formal, explicit signals (e.g. bot

http://scikit-learn.org/stable/
flags) used in prior work to identify bots. Details of the necessary step of “session-izing” our data can be found in Appendix A. As will be discussed, we iterated twice in our model feature selection.

We first offer insights into our initial choice of model features. As stated, previous work showed that bots and humans have different inter-edit time patterns in peer production [5]. Further, bots have been shown to have longer periods of activity in (for example) online games [9]. We used these two implicit behavioral characteristics in our model and also considered the standard deviation of the time difference between edits – a similar behavioral signal as used in web robot detection [19]. We also conjectured that bots would focus mostly on actual items while people would likely also edit elsewhere in the community, for example performing policy development or engaging in other discussion. Such distinct editing activities occur in distinct “namespaces” and we derived features to measure the number of edits to these different areas of the community. Essentially, we felt bot and human edit sessions would have different distributions of edits across namespaces. Appendix B provides details about our initial features under Activity Pattern Features.

After developing and formally testing our initial model’s fitness, we looked for ways to improve it. We applied the model to random samples from our anonymous user contribution sessions dataset (described in detail in Appendix A) and spot-checked sessions classified as “bot” and “non-bot”. This helped us identify more features. One feature was a variant of one of our initial features, designed to better detect bots based on inter-edit time, specifically by counting the number of edits per session with an inter-edit time less than 2 seconds. The remaining features took advantage of the content in revision comments which we describe next.

Revision comments provide information about the nature of a Wikidata revision (used interchangeably in this paper with the word edit). Revision comments are composed of both a structured and an unstructured (free-form) component. Consider the comment “/* wbcreateclaim-create:2| */ [[Property:P107]], [[Q618123]]”. The structured part is between “/*” and “*/” (e.g. “/* wbcreateclaim-create:2| */”) and is automatically generated by Wikidata’s software when a revision occurs. In this case, the structured part indicates that a claim was created. The rest of the comment (e.g. “/* [[Property:P107]], [[Q618123]]”) contains information that a contributor is free to modify and replace. Bots sometimes provide additional information in this unstructured space about the nature of the revision. In our example, default information was left in place indicating the claim modified was mapping property 107 to item 618123. An example of a bot leaving its own information is “/* wbsetlabel-set:1|de */” Bot: change label for de after page move: Stena Nordica -> Malo Seaways”.

Most of our comment-based features distinguished different types of editing behavior by examining structured content. Among other things, such features measured the number of edits to different aspects of an item, for example, to descriptions, aliases, and labels. One feature based on the unstructured part of comments identified bots simply by looking for the word “bot” in the user-provided unstructured comment text. Similarly, another feature also relied on a (pseudo-)explicit signal, namely the presence of a “generic” boilerplate comment containing only a structured part of a certain syntax, for example: “/* wbeditentity-update:0| */”. This particular comment syntax only comes from API edits carried out by bots. Although the latter two features looked for specific and explicit features, they still differed from bot flags. Specifically, these two

---

features are based on informal usage rather than mandated community policies and have not been considered in any previous bot detection methods. More details on the second iteration of model features are listed in Appendix B under Revision Comment-Based Features.

Finally, we re-tuned the gradient boosting model using the additional features and a new distinct test set of registered user contributions to avoid overfitting. Using a grid-search based hyper-parameter optimization, we optimized for PR-AUC by adjusting the number of estimators, max depth, and learning rate. The resulting parameters were: number of estimators = 1100, max depth = 3, and learning rate = 0.1.

4 RESULTS

4.1 Formal Model Evaluation on Registered User Edits

Table 1 provides fitness information for our gradient boosting bot prediction model, both before and after we added the “Revision Comment-Based” features. These statistics result from applying the models to their respective registered user testing datasets: 1) the dataset generated for testing the initial “Activity Pattern” features and 2) the dataset generated for testing the “Activity Pattern” plus “Revision Comment-Based” (“iteration 2”) features. As can be seen, revision comment-based features considerably improved model performance. Figure 1 provides PR-AUC and ROC-AUC graphs from the final model (which contained both feature sets). Using testing data, with default parameters, this model had a precision of 0.88 and recall of 0.69.

<table>
<thead>
<tr>
<th>Features</th>
<th>PR-AUC</th>
<th>ROC-AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.528</td>
<td>0.888</td>
</tr>
<tr>
<td>Initial + Iteration 2</td>
<td>0.845</td>
<td>0.985</td>
</tr>
</tbody>
</table>

Figure 1: Precision-Recall and Receiver Operating Characteristic Curves

4.2 Qualitative Model Evaluation on Anonymous User Edits

As stated earlier, anonymous edits are understudied and potentially quite valuable to peer production [34,35]. Hence, to better understand such edits, we wanted to ensure our model was effective at detecting anonymous bot contributions. When building our model, we hypothesized that the explicitly flagged bots used for model training would have behavioral characteristics similar enough to those of anonymous bots and thus the model trained on one set would be effective on the other. However, we wanted to be sure that this was the case. In this section, we describe how we went about ensuring our model performed well on anonymous contributions.

For unregistered (“anonymous”) user contributions, we did not have labeled testing data so we needed to evaluate model fitness in a different way. We first applied our model to our
anonymous contributions dataset (described in detail in Appendix A) and generated likelihood estimates for each session. These likelihood estimates give the probability that a given session was produced by a bot. Setting a model’s confidence threshold at a given likelihood estimate will correspond to a given level of model precision and recall when applied to the test dataset. If a user wants precision in the model to be higher than 0.95, the confidence threshold in the model can be set to a corresponding likelihood estimate. The model then would return predictions with precision of 0.95, assuming it is applied to data from the same population it was tested on. However, we cannot assume that the registered user data used to train/test the model was similar to the anonymous user data we applied it to. Instead, we had to test whether this was true. Therefore, to get a sense for how the model performed across meaningful likelihood ranges, we sampled at likelihoods with matching test set recall ranges of 0.0-0.1, 0.1-0.2, 0.2-0.3, 0.3-0.4...0.9-1.0. This strategy allowed us to check how precision (specificity) changed as we increased recall (sensitivity) when applying the model to anonymous contributions. If our model was effective, we should expect to see two results from this analysis. First, the model should find bots in the anonymous contributor pool at roughly the same rate as among registered accounts. Second, as one would expect in any prediction model, precision should be high when the recall associated with likelihood values is small and precision should decline as the recall associated with likelihood values increases. Eventually, precision should approach zero when all bots have been removed. If anonymous user sessions have significantly different behavioral characteristics than registered user sessions, we might find that precision stays roughly the same as recall increases. This would mean the model is ineffective at telling bot and human sessions apart.

To determine whether our model was successful for anonymous user contributions in these two regards, we needed to generate ground truth data. Specifically, we needed to know what sessions were from bots and what sessions were from humans at likelihood/recall strata. We refer to these strata as “recall strata” from this point forward for ease of discussion. We generated 20 random session samples per recall strata. We next developed a code book, seen in Table 2, of which the purpose of each code was to identify bot-like session characteristics. Additionally, a final code in our codebook explicitly indicated overall coder judgement as to whether a session was from a bot or human. The first and third author had developed this code book. The first author then systematically went through each sampled edit session within each strata -- examining revision comments, timestamps, namespaces, and page titles. He analyzed enough revisions per session to feel confident with codes he applied. Depending upon the size of the session, this involved looking at as many as 50-100 or more revisions per session.

4.2.1 Summary of Coding Results. Appendix C provides a detailed breakdown of coding results, including a table of results broken down by strata. As stated, the purpose of this coding was to check whether our model identifies bots in the anonymous contribution pool at roughly the same rate as for registered users and that it does so with high precision when recall is low and with declining precision as recall increases. Both characteristics occurred with our model.

Between 0.0 and ~0.3 recall, precision is very high. Between ~0.3 and ~0.7, precision drops noticeably as more human sessions appear. Finally, when recall is greater than ~0.7, most sessions are from humans.

---

8 In one case, there were only 18 sessions in the stratum, in which case we analyzed them all.
Based on our results, we can make application-specific recommendations about how our model can be used. For applications requiring high recall bot prediction (e.g. filtering out bots for research of anonymous human contributors), we recommend optimizing a confidence threshold for recall of at least 0.7-0.8. For applications requiring high precision bot prediction (e.g. filtering out humans for research of anonymous bot contributions), we recommend optimizing a confidence threshold for recall of 0.2-0.3 or lower. Per our informal coding, when recall is between 0.0 and 0.1, precision is 1.0. When recall is between 0.1 and 0.2, precision is 0.9.

Our coding provides interesting insights into session characteristics in low recall strata where the prediction model was the most confident. Boilerplate bot comments (BBC), explicit bot comments (EBC), and fast edits (FE) codes were frequently applied when recall was below 0.4. We had derived model features to capture session characteristics represented by each of these three codes and our coding process identified the effectiveness of these features.

### 4.3 Summary of Model Evaluations
We ensured our bot prediction model worked effectively for both registered contributions (verified via a formal quantitative analysis) and anonymous contributions (verified via a qualitative thematic analysis [28]).

## 5 APPLYING THE MODEL TO REGISTERED AND ANONYMOUS USERS
Given the scenario described and the questions posed in the Introduction, is there compelling evidence to be concerned about impacts from unidentified bots? We applied our model to all registered “human” and anonymous user contributions from November 2012 to April 2017 for items and properties in Wikidata (the two main “content” namespaces) to find out. Per the
recommendations mentioned in the last section, we set a confidence threshold that would correspond to 0.3 recall on our test data.

We found that 3% of registered “human” contributions were predicted to have come from bots. We broke this percentage down by month and found that 8 months in Wikidata’s history were predicted to have had more than 5% of such edits coming from bots (mean = 0.02, s.d. = 0.04). April 2017 experienced the largest percentage of predicted unflagged bot edits at 18%.

We found that 2% of anonymous contributions were predicted to have come from bots. We also broke this percentage down by month and found that 6 months in Wikidata’s history were predicted to have had more than 5% of such edits coming from bots (mean = 0.02, s.d. = 0.04). May 2016 experienced the largest percentage of predicted unflagged bot edits at 16%.

5.1 Implications on Behavioral Research in Peer Production

Application of our model indicates that most human or bot behavioral studies in Wikidata do not likely need to consider the effects of unidentified bots. When studying behavior across periods of time of several months or greater, the chances of unidentified bot activity skewing results appear minimal if considering the behavioral characteristics of the “typical” contributor (e.g. taking means, medians, and similar outlier robust statistics).

However, for short-term behavioral studies, it’s possible that anomalous spikes in undetected bot activity can affect results. For example, consider the scenario in April 2017 when 18% of “human” edits were predicted to be unflagged bots. Application of our model would be necessary to avoid false understanding of human behavior. Further, analysis of “outlier” contributor behavior (e.g. what human contributors produced the most edits in each month?) can be affected significantly by undetected bots. Consider Figure 2. The “All Sessions” plot represents the longest edit sessions per month in our anonymous contributor session dataset. The “Non-bot Only Sessions” plot represents the same analysis after we applied our bot prediction model to filter out bots. To ensure that nearly all bots would be removed, we used a confidence threshold that would correspond to 0.8 recall per the recommendations in the previous section.

Figure 2: Maximum Monthly Anonymous User Session Lengths
5.2 Implications for the Wikidata Community and Applications using Wikidata

The 2-3% of contributions in human contribution datasets that are unidentified bots are of serious concern to the Wikidata community and applications using Wikidata. This equates to over 1 million such edits. If even a relatively small portion of these are vandalism, the effect on data quality in the community can be large. Further, such damaging edits could propagate downstream to applications that use Wikidata such as Wikipedia and Google Knowledge Graph. Hence, we hope that our prediction model in the future can help the community identify and quickly mitigate any potentially damaging behavior. Further, work [17] has used contributor type (bot or human) as a Wikidata vandalism detection model feature. Our work could help improve model performance.

6 DISCUSSION

6.1 Model and Feature Implications

6.1.1 Future Studies. Our model opens the door for a number of exciting research directions and socio-technical contributor tools. For example, future important questions in Wikidata require accurate bot detection in order to be answered. Much more work is needed to understand how human, bot, and various editing tool contributions relate to content quality, and also how such contributions relate to interest and consumption patterns. Considering the initial findings in Wikidata that a higher percentage of bot edits correlates with higher quality content [15], are bots putting their effort towards content that gets used in applications like Google Knowledge Graph and Wikipedia? Further, work in Wikipedia has indicated that the supply of good content often does not align with reader interest (demand) [22]. Are bots creating “good” content in Wikidata that is of little interest or use? Additionally, how much – and what type – of work is being done by Wikidata bots that are not explicitly flagged as bots?

6.1.2 Feature Implications. Revision comment-based features appear to be highly effective at predicting bot edits. Our initial model did not use these features and had a PR-AUC of 0.528 and ROC-AUC of 0.888. The addition of 13 features based on characteristics of revision comments helped move PR-AUC to 0.645 and ROC-AUC to 0.985. Wikidata’s structured comments denote the types of edits occurring. Capturing when these edit types occur appears to be a highly-predictive way of determining whether edits are from bots or humans. Further, based on our qualitative analysis, features derived from edit comments that are exclusive to bots (e.g. “generic” comments and comments containing the word “bot”) appeared to be effective at identifying bots. For example, “generic” comments in Wikidata are indicative of bot edits coming from Wikidata’s editing API. However, to ensure that other features also provided useful bot prediction signal, we removed the features “Word ending in “bot” (case insensitive) in revision comment” and ‘# of bot generated generic comments’ and found that model performance changed minimally (PR-AUC: 0.849, ROC-AUC: 0.982). This indicates that the cumulative predictive power of other features in this set is quite useful.

Bot prediction models in other peer production communities such as OpenStreetMap and Wikipedia almost certainly would benefit from similar features to those employed in Wikidata. As shown, inter-edit time is an effective differentiator of bots and humans in Wikipedia [3,5] and could be incorporated into a future model. Additionally, Wikipedia edit tags [37] can perhaps provide explicit and implicit prediction signals similar to those identified in Wikidata revision comments. Further, although there is no “structured” component of Wikipedia revision....
comments to use to identify distinct editing behaviors, work [23] has been successful at “deriving that structure” with a prediction model to identify editing behaviors from unstructured Wikipedia comments. Further, features such as inter-edit time might also be relevant for predicting whether OpenStreetMap changesets (similar to Wikidata and Wikipedia editing sessions) are human or bot-produced. Such changesets also contain tags which can provide signal regarding whether edits originate from bots [38]. For example, the tag “bot=yes” can be applied to explicitly indicate bot changesets [38].

6.2 Model Improvements

We inform future work to improve the prediction model in several ways. First, when recall is between ~0.3~0.7 (and especially between ~0.4~0.5), we coded a large number of sessions as “unknown” (it was unclear whether a session was from a bot or a human). From a feature engineering standpoint, close attention should be paid to characteristics of these sessions as additional, high-value bot prediction signal could be found there. Further, we uncovered an additional likely source of bot prediction signal in a comment: “/* wbsetreference */”. In the future, this (and other similar comments, if they exist) could be incorporated into the feature that we called “# of bot generated generic comments”. Additionally, we noticed that bots often increment through items in order of their “item id”. This fact could also be useful as future prediction signal. Further, there was a single session in the 0.8-0.9 recall stratum that was coded as a bot (see Appendix C for details). Outliers such as this may be indicative of humans running a bot in a “semi-automated” tool-like fashion. Given the lack of outliers such as this, it appears our model did well differentiating bots from semi-automated tools operated by humans. Future work may wish to differentiate human edits based on their level of automation (e.g. semi-automated tool use versus completely manual editing). Finally, the edit tags mentioned in the previous section are also used in Wikidata and could potentially be a good source of prediction signal.

7 CONCLUSION

We built a bot detection model for Wikidata based on behavioral and other informal contribution characteristics. Our model improves upon current methods of bot recognition in peer production that rely on contributors formally and explicitly denoting bots as such. We ensured our model performed well by formally evaluating it against registered user data and by qualitatively evaluating its ability to perform with anonymous edits where formal, explicit signals for testing are not available. By applying to this latter dataset, we begin exploration into an understudied and potentially important dynamic in Wikidata and other peer production communities. Through application of our model, we find that while some studies of contributor behavior may need to be concerned about unidentified bots affecting results, most likely do not. Our findings are also important to the Wikidata community as they indicate that bot approval processes have not been completely effective. Finally, we provide our train/test data, code, and model hyper-parameters for others to use in their work.

ACKNOWLEDGMENTS

We thank members of the Wikimedia Foundation and the Wikidata community for assistance and insight related to this project. We also thank members of GroupLens Research for further assistance.
REFERENCES

Received April 2018; revised July 2018; accepted September 2018.

A DETAILS OF DATA PREPARATION

A.1 Extracting Revision Data and Sessionization

Building our prediction model required historical Wikidata revision data. When a bot or human edits Wikipedia, they produce a revision (we use the word “edit” interchangeably) which contains the changes made as well as metadata such as namespace, page title, revision id, user id, username (in the case of anonymous user edits, IP address), timestamp, and comment. A given namespace corresponds to a distinct type of Wikidata content. Namespaces exist for items, properties (which represent attributes of items), general community discussion, policy development, and more. To access revision data for all pages throughout Wikidata’s history, we used the “stub-meta-history” xml dump files from May 1, 2017 from https://dumps.wikimedia.org. There were over 476 million revisions, predominantly from namespace 0 (item entities). We extracted this dump data using the mwxml utility\textsuperscript{10}.

We next “session”-ized all revision data. As mentioned previously, prior work [3,5] has defined a human editing “session” as a period in which a human makes edits without taking large breaks. This work identified implicit behavioral signals in editing sessions that have predictive power to distinguish bots from human contributors. Specifically, people generally edit with predictable inter-edit times. Bots typically edit with significantly smaller inter-edit times. We incorporated these and other implicit session characteristics in our model.

To session-ize all the revisions from logged-in users for model training and testing (ground truth labels are discussed in section A.2), we applied the mwsessions utility\textsuperscript{11} with default parameters. As recommended by prior research [3,5], the default cutoff of greater than one hour of inactivity was used to determine the end of one session and the beginning of another by the same user. This produced 4,990,652 logged-in user sessions corresponding to 177,134 unique user ids. The median number of sessions per user id was 1, while the mean was 28.17.

\textsuperscript{10} http://pythonhosted.org/mwxml/
\textsuperscript{11} http://pythonhosted.org/mwsessions/
We then randomly sampled 100,000 sessions for model training and 100,000 for testing. We filtered out sessions from the test set that were also in the training set to ensure no overlap. We also filtered out sessions that did not contain at least one edit to a Property- or an Item-page – the main content of Wikidata. Further, since we are concerned mainly with large sequences of potential bot activity, we filtered out sessions with less than three edits. After these filtering steps, we had 36,252 training and 35,191 testing sessions.

To test our classifier model's performance on anonymous user bot activities, we gathered a similar sample of anonymous activities. We session-ized all of Wikidata’s 2,114,900 anonymous revisions into 677,050 sessions, session-izing based on IP address. After following the same filtering steps that we used for training and testing datasets, we had 110,288 sessions for testing the detection of anonymous bot editing.

Recall that anonymous contributor edits are tied to an IP address; therefore, one cannot assume that a given IP address will correspond to only one user over long periods of time. However, an IP address is unlikely to change within a continuous activity session, so it serves as a durable-enough identifier for our purposes [3].

A.2 Ground Truth Bot Account Data
To obtain labeled training and testing data for registered user edits, we compiled a list of all currently and formerly approved bot accounts. We obtained this data by looking for "bot flags" in two publicly accessible tables called "user_groups" and "user_former_groups"[12]. We then merged this bot account data with the registered user revision training and testing data based on the user id fields in each dataset[13]. All accounts not labelled as bots were considered human accounts unless their username ended in “bot” (insensitive to case), in which case they were omitted from the training and testing datasets. This was a conservative approach in that we wanted to train/test our model on the least noisy data possible in order to get the best predictions. In our training set, 34,797 were labeled as human sessions and 1455 as bot. In our testing set, 33,775 were labeled as human sessions and 1416 as bot. Since there are more bot edits than human edits in Wikidata, one might wonder why our dataset contains so many more human sessions than bot sessions. This is because bot edit sessions tend to produce more edits than human sessions.

---

[13] One “bot flagged” user account was not considered to be a bot account when merging because we had reason to believe that it had not been used exclusively for bot editing.
B MODEL FEATURES

<table>
<thead>
<tr>
<th>Activity Pattern Features</th>
<th>Description (if needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-edit time mean</td>
<td></td>
</tr>
<tr>
<td>Inter-edit time standard deviation</td>
<td></td>
</tr>
<tr>
<td># of edits</td>
<td></td>
</tr>
<tr>
<td>(Session) length in seconds</td>
<td></td>
</tr>
<tr>
<td># of edits with inter-edit time &lt; 5 seconds</td>
<td></td>
</tr>
<tr>
<td># of edits with inter-edit time &gt;= 5 seconds and &lt;= 20 seconds</td>
<td></td>
</tr>
<tr>
<td># of edits with inter-edit time &gt; 20 seconds</td>
<td></td>
</tr>
<tr>
<td># of Namespace 0 edits</td>
<td>The main &quot;item&quot; namespace.</td>
</tr>
<tr>
<td># of Namespace 1 edits</td>
<td>The &quot;talk&quot; namespace of Namespace 0.</td>
</tr>
<tr>
<td># of Namespace 2 edits</td>
<td>Namespace for contributors to provide information about themselves.</td>
</tr>
<tr>
<td># of Namespace 3 edits</td>
<td>The &quot;talk&quot; namespace of Namespace 2.</td>
</tr>
<tr>
<td># of Namespace 4 edits</td>
<td>Namespace for specification of community rules/policies.</td>
</tr>
<tr>
<td># of Namespace 5 edits</td>
<td>The &quot;talk&quot; namespace of Namespace 4.</td>
</tr>
<tr>
<td># of Namespace 120 edits</td>
<td>Where properties are defined.</td>
</tr>
<tr>
<td># of Namespace 121 edits</td>
<td>The &quot;talk&quot; namespace of Namespace 120.</td>
</tr>
</tbody>
</table>

Revision Comment-Based Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description (if needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td># of claim creations</td>
<td></td>
</tr>
<tr>
<td># of unique claims changed</td>
<td>This was measured by determining the number of properties changed.</td>
</tr>
<tr>
<td># of distinct pages edited (across all namespaces)</td>
<td></td>
</tr>
<tr>
<td># of distinct edit types</td>
<td>We parsed the “structured” part of comments to find different revision types. For example, “wbsetdescription” indicates a description is being set. “wbcreateclaim-create” indicates the creation of a new claim.</td>
</tr>
<tr>
<td># of bot generated generic comments</td>
<td>Certain boilerplate revision comments can only be created via the Wikidata editing API (e.g. &quot;*/wbeditentity-update:0</td>
</tr>
<tr>
<td>Word ending in “bot” (case insensitive) in revision comment</td>
<td></td>
</tr>
<tr>
<td># of sitelinks changed</td>
<td></td>
</tr>
<tr>
<td># of aliases changed</td>
<td></td>
</tr>
<tr>
<td># of labels changed</td>
<td></td>
</tr>
<tr>
<td># of descriptions changed</td>
<td></td>
</tr>
<tr>
<td>Same exact edit occurred more than once (possible edit war)</td>
<td></td>
</tr>
<tr>
<td># of revisions removing content</td>
<td></td>
</tr>
<tr>
<td># of revisions modifying content</td>
<td></td>
</tr>
<tr>
<td># of edits with inter-edit time &lt; 2 seconds</td>
<td></td>
</tr>
</tbody>
</table>

C DETAILED ANONYMOUS CONTRIBUTOR QUALITATIVE CODING RESULTS

We break down the high-level results of our anonymous contributor qualitative coding in each stratum. Results are summarized in Table 3.

Recall: 0.0-0.1 stratum. As seen in Table 3, all 15 sessions sampled (this stratum had 15 sessions total) were clearly from bots. Most sessions (11) in this stratum contained revisions with boilerplate bot comments (coded as BBC) that made bot recognition straightforward. Some also contained quickly occurring edits (that were coded as FE).

Table 3: Overview of the Results of the Anonymous Contributor Qualitative Coding Summary.
This table organizes results of our coding by “recall stratum” (defined in the text). The fundamental pattern is that low-recall strata include all or nearly all bot sessions, high-recall strata include all or nearly all human sessions, and that the intermediate strata contain a mix of the two. The coding results also provide explanations of the makeup of the sessions in each stratum, specified by the codes defined in Table 2 of the paper. This analysis confirms that the model is effective at identifying edit sessions made by bots.

<table>
<thead>
<tr>
<th>Recall Stratum</th>
<th># Bots Sessions</th>
<th># Human Sessions</th>
<th># Unknown Sessions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0-0.1</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>11/15 sessions were coded BBC. Lots of FE sessions</td>
</tr>
<tr>
<td>0.1-0.2</td>
<td>18</td>
<td>0</td>
<td>2</td>
<td>9/20 sessions were EBC or FE. Quite a few BBC sessions as well</td>
</tr>
<tr>
<td>0.2-0.3</td>
<td>16</td>
<td>4</td>
<td>0</td>
<td>Quite a few EBC or FE sessions again. A few obvious human sessions including edit war(s)</td>
</tr>
<tr>
<td>0.3-0.4</td>
<td>11</td>
<td>7</td>
<td>2</td>
<td>Most bot sessions were BBC or FE sessions</td>
</tr>
<tr>
<td>0.4-0.5</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>Half of the sessions were coded as unknown</td>
</tr>
<tr>
<td>0.5-0.6</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>A few bot sessions were SIM. One session simply had blank comments</td>
</tr>
<tr>
<td>0.6-0.7</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>More edits appeared human than bot. A number were still hard to identify as either and coded as unknown</td>
</tr>
<tr>
<td>0.7-0.8</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>Nearly all sessions were obviously human. A couple sessions appeared to be possible vandalism</td>
</tr>
<tr>
<td>0.8-0.9</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>19/20 sessions were obviously human in behavior. One contained “(BOT)” in beginning of the unstructured part of its comment</td>
</tr>
<tr>
<td>0.9+</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>All sessions were clearly human</td>
</tr>
</tbody>
</table>

**Recall: 0.1-0.2 stratum.** 18 sessions were clearly from bots. 2 were hard to identify and coded as unknown. Of the bot sessions, half were coded as EBC (they contained “bot” in their revision comments) or FE. As with the previous stratum, a number of sessions were coded as BBC as well.

**Recall: 0.2-0.3 stratum.** 16 sessions were clearly from bots and 4 were clearly from humans. As with previous sessions, several EBC or FE sessions occurred. One of the human sessions appeared to be part of edit war(s) -- the contributor was restoring content on multiple pages. For example, related to the item for Kim Kardashian, the contributor stated in the comment of a restoring edit: “...I definitely don’t understand why [you] removed Kendall and Kylie, they’re her sisters”.

**Recall: 0.3-0.4 stratum.** As seen in Table 3, at this point, a fair number of human sessions began appearing (7 human, 11 bot, 2 unknown). Bot sessions were primarily BBC or FE.

**Recall: 0.4-0.5 stratum.** Often no clear indicators existed to code sessions as either bot or human in this stratum, and the result was a large increase in sessions coded as unknown (10). There were 4 human and 6 bot sessions.

**Recall: 0.5-0.6 and 0.6-0.7 strata.** At this point, model precision had decreased significantly. The number of unknown sessions had decreased from the previous stratum while human sessions increased. 13 sessions (cumulatively across the two stratum) were from bots, 14 from humans, and 13 were coded as unknown.
Recall: 0.7-0.8 stratum. This represented a clear turn towards human sessions. As seen in Table 3, 16 sessions were clearly from humans, 2 from bots, and 2 were coded as unknown. One human editor appeared to be vandalizing pages, providing comments such as “cgyjcfgmhuk”.

Recall: 0.8-0.9 and 0.9+ strata. In the remaining two strata, all sessions were clearly from human editors, except for one in the 0.8-0.9 stratum which was coded as EBC.