Approving Automation: Analyzing Requests for Permissions of Bots in Wikidata

Mariam Farda-Sarbas
Human-Centered Computing | Freie Universität Berlin
mariam.farda.sarbas@fu-berlin.de

Hong Zhu
Human-Centered Computing | Freie Universität Berlin
hong.zhu@fu-berlin.de

Marisa Frizzi Nest
Human-Centered Computing | Freie Universität Berlin
marisa.nest@fu-berlin.de

Claudia Müller-Birn
Human-Centered Computing | Freie Universität Berlin
clmb@inf.fu-berlin.de

ABSTRACT

Wikidata, initially developed to serve as a central structured knowledge base for Wikipedia, is now a melting point for structured data for companies, research projects and other peer production communities. Wikidata's community consists of humans and bots, and most edits in Wikidata come from these bots. Prior research has raised concerns regarding the challenges for editors to ensure the quality of bot-generated data, such as the lack of quality control and knowledge diversity. In this research work, we provide one way of tackling these challenges by taking a closer look at the approval process of bot activity on Wikidata. We collected all bot requests, i.e. requests for permissions (RfP) from October 2012 to July 2018. We analyzed these 683 bot requests by classifying them regarding activity focus, activity type, and source mentioned. Our results show that the majority of task requests deal with data additions to Wikidata from internal sources, especially from Wikipedia. However, we can also show the existing diversity of external sources used so far. Furthermore, we examined the reasons which caused the unsuccessful closing of RfPs. In some cases, the Wikidata community is reluctant to implement specific bots, even if they are urgently needed because there is still no agreement in the community regarding the technical implementation. This study can serve as a foundation for studies that connect the approved tasks with the editing behavior of bots on Wikidata to understand the role of bots better for quality control and knowledge diversity.

KEYWORDS

Wikidata, Bot, Task Approval

ACM Reference Format:
Mariam Farda-Sarbas, Hong Zhu, Marisa Frizzi Nest, and Claudia Müller-Birn. 2019. Approving Automation: Analyzing Requests for Permissions of Bots in Wikidata. In *The 15th International Symposium on Open Collaboration (OpenSym '19), August 20–22, 2019, Skövde, Sweden.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3306446.3340833

OpenSym '19, August 20–22, 2019, Skövde, Sweden
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6319-8/19/08…$15.00
https://doi.org/10.1145/3306446.3340833

1 INTRODUCTION

Wikidata, the sister project of Wikipedia, was launched in October 2012. Similar to Wikipedia, Wikidata is a peer production community with more than 20 K active users[1]. It serves as the primary source for structured data of Wikimedia projects, especially for Wikipedia. Wikidata is also currently being used by commercial services, such as Apple's assistant Siri, in research projects, such as WikiGenomes[2] and other peer-production communities, such as OpenStreetMap[3]. Thus, the quality of data in Wikidata is crucial, not only for Wikidata itself but also for the projects which rely on Wikidata as their source of data.

However, early research on Wikidata's community has already shown that in addition to an active human editor community, bots are responsible for the majority of edits (e.g., [10, 16, 17]). Contrary to Wikipedia, bots were massively used in the Wikidata community from the beginning of the project; therefore, it seems they exhibit an essential duty in Wikidata. Bots have been studied for some time in the context of Wikipedia (e.g. [3, 5, 9]), however, their role in the context of the Wikidata community is less explored. Concerns were raised recently about the role of bots in Wikidata. The quality of bot-generated data, for example, and their influence on quality control and knowledge diversity has been questioned [17].

This motivation led to the research presented in this paper, which is guided by the question: What type of bot activities are approved by the Wikidata community? We are interested in understanding what type of tasks the Wikidata community permits to be performed through bots, i.e. to automate.

As opposed to previous work [10, 16, 20] or [18] which was based primarily on actual bot edits, we analyzed the requests for permission (RfP) process and the accompanying information in Wikidata. In the RfP process, a bot operator applies for task approval for each task a bot is intended to carry out. The community then decides on each task.

We collected all task descriptions from October 31, 2012, to July 01, 2018, and classified and analyzed them manually. Our results suggest that the community uses bots mainly to add data to Wikidata. These data originate primarily from Wikipedia. Furthermore, the main reasons for unsuccessful RfPs are either the operators themselves, because the specification of the task is insufficient, or the community, who could not agree on the design of the technical implementation. The contributions of our research are as follows:

[1] https://www.wikidata.org/wiki/Wikidata:Statistics.
[2] Community-created genome data portal available at www.wikigenomes.org.
[3] An online map with an open license, available at https://www.openstreetmap.org.

- We investigated bots from a community perspective, based on RfPs.
- We provided a classification scheme for the categorization of RfPs.
- We used our defined scheme to identify bot-requested tasks and data sources.
- We analyzed how the community decides on RfPs and classified the reasons for unsuccessful RfPs.
- We developed a dataset of bots RfPs.

In the following sections, we, firstly, present existing research on bots in peer-production by focusing on Wikimedia projects and how our work is built on them. We then explain our dataset and classification approach. Next come the findings of the study, followed by a discussion of the results. We conclude this article by highlighting possible directions for future research.

2 BOTS IN PEER PRODUCTION

Very different bots populate peer production communities: These bots collect information, execute functions independently, create content or mimic humans. The effects of bots on peer production systems and our society are increasingly being discussed, for example, when influencing voting behaviour [1] or imitating human behaviour [13].

Bots have been used in Wikipedia from early on[4]. Wikidata's community has profited from these experiences when handling their bots. We review, therefore, existing insights into the bot community on Wikipedia and building on that, highlight research on Wikidata that considers bot activity.

2.1 Bots in Wikipedia

Bots are software programs that automate tasks, usually repetitive or routine tasks which humans consider time-consuming and tedious (e.g. [3, 4, 16, 21]). They are operated and controlled by humans.

Wikipedia has developed a stable and increasingly active bot community over the years, although, bots were not widely accepted and trusted in the beginning [14]. Halfaker et al. distinguishes four types of bots in Wikipedia [9]: (1) Bots that transfer data from public databases into articles, (2) bots that monitor and curate articles, (3) bots that extend the existing software functionality of the underlying Wikipedia MediaWiki software and (4) bots that protect against vandalism.

The majority of research focuses on bots that protect against vandalism. Geiger et al. [7], for example, investigated the process of fighting vandalism in Wikipedia by using trace ethnography. They show how human editors and bots work together to fight vandalism in Wikipedia. They conjecture that such distribution of concerns to human and algorithmic editors might change the moral order in Wikipedia. Halfaker et al. [8] show how a distributed cognitive network of social and algorithmic actors works efficiently together to detect and revert vandalism on Wikipedia. In another study, Geiger & Halfaker [5] investigated the impact of a counter-vandalism bot's downtime on the quality control network of Wikipedia and found that during this downtime, the quality control network performed slower but was still effective.

Another piece of research focuses on how "such tools transform the nature of editing and user interaction". Geiger shows "how a weak but pre-existing social norm was controversially reified into a technological actor" [3]. He refers to the example of the HagermanBot, which has been implemented to sign unsigned discussion entries in Wikipedia. Halfaker et al. [9] show that bots are not only responsible for the enforcement of existing guidelines on a larger scale, but also that their activities can have unexpected effects. The number of reverts of newcomers' edits, for example, has elevated, while (surprisingly) the quality of those edits has stayed almost constant. Editors increasingly apply algorithmic tools for monitoring edits of newcomers. In 2010, 40 percent of rejections of newcomers' contributions were based on these algorithmic tools [9]. This contradicts attempts of the community to engage more new editors. Moreover, Geiger and Halfaker defined bots as "assemblages of code and a human developer" and show that bot's activity is well aligned with Wikipedia's policy environment [6].

The research suggests that bots are more critical to the success of the Wikipedia project than expected previously, despite the reluctance of the Wikipedia community to allow bots at the beginning [14]. Bots have a significant role in maintaining this text-based knowledge base, especially in fighting vandalism. As bots in Wikidata have their roots in Wikipedia, we expect to see similarities between bots in both peer production systems - Wikipedia and Wikidata. Before we look closer to see if the same areas of use of bot activities emerge from Wikidata, we give an overview of the existing insights into the bot community in Wikidata.

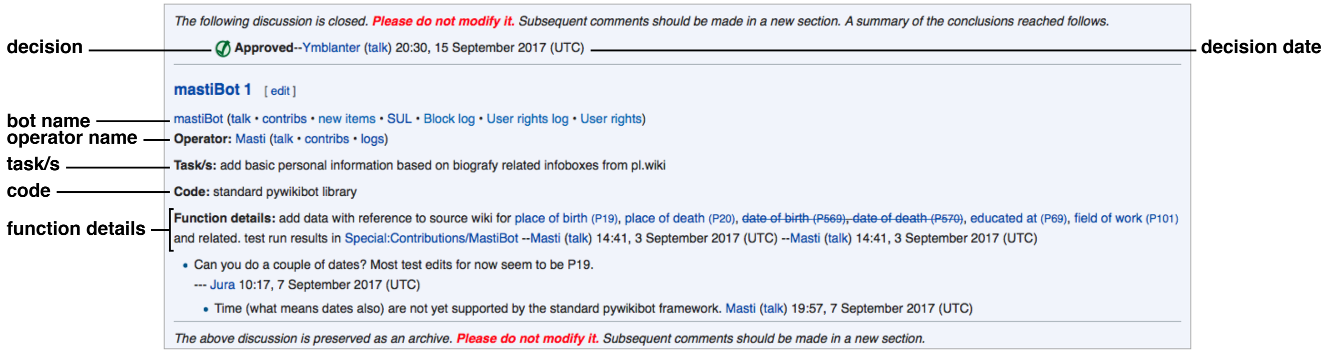
2.2 Bots in Wikidata

Wikidata inherited bots from its sister project Wikipedia and bots started editing Wikidata with its launch by linking Wikidata item pages to their respective Wikipedia language pages. The current research on Wikidata bots shows that bots perform most of the edits in Wikidata [16, 20]. Steiner [20], in his research, aims to understand the editing distribution of editors on Wikidata and Wikipedia. He provides a web application to observe real-time edit activity on Wikidata for bots and logged-in and anonymous users. The study shows that the number of bots vs. the number of edits has grown in a linear form and most of Wikidata's edits, i.e. 88%, account for bot edits.

Müller-Birn et al. [16] can confirm these insights in a later study by studying the community editing patterns of Wikidata through a cluster analysis of contributors' editing activities. They determine six editing patterns of the participating community (i.e. reference editor, item creator, item editor, item expert, property editor and property engineer) and show that bots are responsible for simpler editing patterns, such as creating items, editing items, statements or sitelinks.

Further studies focus on how bot edits contribute to data quality in Wikidata. In one study on Wikidata's external references, Piscopo et al. [18] find that the diversity of external sources in bot edits is lower than in human edits. In another study, Piscopo et al. [19] explore the influence of bots and humans (registered and anonymous) contributions on the quality of data in Wikidata. The

[4] https://en.wikipedia.org/wiki/Wikipedia:History_of_Wikipedia_bots#"rambot"_and_other_small-town_bots.



*The following discussion is closed. **Please do not modify it.** Subsequent comments should be made in a new section. A summary of the conclusions reached follows.*

Approved--Ymblanter (talk) 20:30, 15 September 2017 (UTC)

mastiBot 1 [edit]

mastiBot (talk • contribs • new items • SUL • Block log • User rights log • User rights)

Operator: Masti (talk • contribs • logs)

Task/s: add basic personal information based on biografy related infoboxes from pl.wiki

Code: standard pywikibot library

Function details: add data with reference to source wiki for place of birth (P19), place of death (P20), ~~date of birth (P569), date of death (P570)~~, educated at (P69), field of work (P101) and related. test run results in Special:Contributions/MastiBot --Masti (talk) 14:41, 3 September 2017 (UTC) --Masti (talk) 14:41, 3 September 2017 (UTC)

- Can you do a couple of dates? Most test edits for now seem to be P19.
  --- Jura 10:17, 7 September 2017 (UTC)
  - Time (what means dates also) are not yet supported by the standard pywikibot framework. Masti (talk) 19:57, 7 September 2017 (UTC)

*The above discussion is preserved as an archive. **Please do not modify it.** Subsequent comments should be made in a new section.*

Figure 1: Example of a request-for-permissions (RfP) page with the extracted fields.

research shows that equal contributions of humans and bots have a positive impact on data quality, while more anonymous edits lead towards a lower quality.

Hall et al. [10] analyzed these anonymous edits on Wikidata to detect bots in this group which had not previously been identified in Wikidata. The study shows that two to three percent of the contributions (more than 1 million edits), considered as human contributions previously came from unidentified bots. They emphasize that it might be a concerning issue for Wikidata and all projects relying on these data. Even if vandalism causes a small portion of these edits, the damaging effect on Wikidata might be high.

This danger is reflected by Piscopo [17], who highlights significant challenges for Wikidata that might endanger its sustainability: Namely, a lack of quality control because of the large amount of data added by bots, a lack of diversity because of the usage of a few sources only and existing threats to user participation because of bot usage.

Existing research focuses primarily on the activity levels of bots in Wikidata based on their edits. Some work (e.g. [17]) conveys the impression that bots are independent of humans. However, this ignores the fact that humans operate bots. The community officially grants most bot activities in a well-defined process.

It is visible from the literature that bots are, so far, studied mainly from their activity angle both in Wikipedia and Wikidata. In Wikipedia, bots are used primarily for quality assurance tasks, i.e. vandalism detection and maintenance tasks, for example, removing/replacing templates on articles, while Wikidata’s bots are performing tasks mostly related to content editing. A possible reason could be the structured nature of Wikidata content, which is less challenging to deal with than the unstructured data in Wikipedia. It is interesting to dig into the content editing activities for which the bots have asked most in Wikidata and approved by the community. While adding more content through bots seems to contribute to data quality from completeness angle, this raises the question whether this data is based on sources and improves data quality from the data trustworthiness aspect. The following study provides the ground for answering such questions in the future. We investigate the RfP process of bots in more detail. We describe which tasks Wikidata’s community find useful for bots and in what cases the community does not support a bot request.

3 APPROVING BOT TASKS

In this section, we describe the bot approval process on Wikidata, explain our data collection, detail the classification process and, finally, give an overview of the resulting dataset.

3.1 Approval Process in Wikidata

Building on the experiences of the Wikipedia community, Wikidata has had a well-defined policy system for bots almost from the beginning (November 2012)[5]. Except for cases like editing Wikidata’s sandbox[6] or their own user pages, every (semi-) automatic task carried out by a bot needs to be approved by the community. It means that before operators can run their bots on Wikidata, they have to open an RfP for their bot[7]. The RfP is, thus, the formal way of requesting bot rights for an account where the decisions on RfPs are based on the community consensus.

An RfP is caused by either an editor’s need, or by a community request[8]. Such a bot request is well documented and available to all interested community members on Wikidata.

Figure 1 shows a typical RfP page. It consists of a bot name[9], an operator name (bot owner), tasks (a brief description of what the bot intends to do), link to the source code, function details (a detailed description of bot tasks and sources of imported data) and the final decision (RfP approved or not approved with a reason).

Bot owners have to use a template for providing all information for the decision-making process and need to clarify questions regarding their request during the decision-making process. They also often have to provide a number of test edits (50 to 250 edits).

[5] https://www.wikidata.org/w/index.php?title=Wikidata:Bots&oldid=549166.
[6] Wikidata’s sandbox can be used for test edits: https://www.wikidata.org/wiki/Wikidata:Sandbox.
[7] All open requests are available at www.wikidata.org/wiki/Wikidata:Request_for_permissions/Bot.
[8] An example can be found at: https://www.wikidata.org/w/index.php?title=Wikidata:Bot_requests&oldid=38203229#Polish_.22Italian_comune.22_descriptions.
[9] According to the bot policy, bot accounts are recommended to have the word bot in their names.

Table 1: Example of the information extracted from an RfP page.

URL	www.wikidata.org/[...]/Bot//MastiBot
Bot Name	mastiBot
Operator Name	Masti
Task and Function	add basic personal information based on biografy related infoboxes from pl.wiki
Decision	approved
Decision Date	15/09/2017
First Edit	03/09/2017
Last Edit	21/02/2018
No. of Edits	8
No. of Editors	5

The community handles RfPs in accordance with the Bot Approval Process[10]. The community discuss whether the task requested is needed and whether the implementation of the task functions properly. After the decision has been made, an administrator or a bureaucrat close the request, if approved, a bureaucrat will flag the account, and if not, the request is closed stating the reason for the unsuccessful request.

After a successful request, each bot operator has to list the task the bot performs with a link to the RfP on its user page. However, the bot operator is required to open a new RfP if there is a substantial change in the tasks the bot performs. Consequently, bots can have multiple requests which are shown on the user page. Furthermore, in special cases, a bot is also allowed to carry out administrative tasks, such as blocking users or deleting or protecting pages. In this case, the operator needs to apply for both, the RfP and the administrator status[11].

3.2 Data Collection

We collected all bot requests that were in the final approval stage in July 2018[12], i.e. we ignored all tasks without a final decision, from Wikidata's archive for requests[13]. We collected our data based on web scraping, i.e. web data extraction programs implemented in Python. Bot requests which were listed several times in different archives were only parsed once[14]. This resulted in 685 task approval pages.

We extracted the following information from each page (cf. Figure 1): URL, bot and operator name, decision, decision date, tasks, code and function details. We additionally collected the date of the first and last page edit[15], the number of page edits and the number of distinct editors who contributed to the request for each page by using the MediaWiki API[16]. This extracted information (cf. Table 1) was processed and stored in a relational database[17].

3.3 Task Approval Data

We collected 683 distinct requests from Wikidata; however, two of them are no longer available and, consequently, we excluded them. Of the resulting 681 requests, 600 requests (88%) were successful, and 81 (12%) were unsuccessful.

An average of five people ($\sigma = 2.8$) participated in an approval request, for example, by taking part in the discussion or stating the final decision. These people accounted for an average of slightly above ten edits for each request ($\sigma = 8.5$).

Based on the requests collected, we identified 391 distinct bots and 366 distinct bot operators on Wikidata (cf. Table 2). Some operators applied for more than one task for their bot in one request (e.g. update label, update description, update alias), with five being the highest number. Furthermore, bot owners can operate more than one bot. The majority of operators (319 editors) have only one bot, while three editors were running three bots and there was even one operator managing seven bots simultaneously. Similarly, one bot can also be operated by more than one user, for example, three editors were managing the ProteinBoxBot together.

Table 2: Overview on task requests (n = 683, number of all task requests).

Decision Result	Requests	No. of Bots	Operators
Successful	600	323	299
Unsuccessful	81	68	67
Both	681	391	366

3.4 Classification Process

We classified the data collected manually to gain a deeper understanding of the different tasks that bot operators carry out on Wikidata. In the following, we describe this process in detail.

We employed the qualitative method of content analysis which is used to analyze unstructured content [12] which in our case, is the text in the RfPs. We classified the data manually using the open coding approach of the grounded theory. The data we were dealing with were applicants' own sentences (in vivo codes), thus, we developed an emergent coding scheme and categorized the textual information. Two of the authors[18] coded the data in a three-phase process by ensuring the consistency of the codes and reducing possible bias during the coding.

We read a sample of the data collected and discussed the diverse request details. We noticed that some common categories could be found within all requests based on some particular perspectives such as the potential actions of the requesting bots and the main sources of the data to be imported from bots. We focused, thus, on task and function details of the RfP page and extracted the intended

[10] A detailed description is available at www.wikidata.org/wiki/Wikidata:Bots.

[11] Further information can be found at https://www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Administrator.

[12] The first request in our data set was opened on October 31, 2012 and the last one was on June 29, 2018.

[13] www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Archive#Requests_for_bot_flags.

[14] For example, www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/VIAFbot is listed in www.wikidata.org/wiki/Wikidata:Requests_for_permissions/RfBot/March_2013 and in www.wikidata.org/wiki/Wikidata:Requests_for_permissions/RfBot/April_2013 and only the later one was used.

[15] The first edit can be interpreted as the request opening and last edit as the request closing or date of archiving.

[16] www.mediawiki.org/wiki/API:Main_page.

[17] The database is released under a public license on GitHub: https://github.com/FUB-HCC/wikidata-bot-request-analysis.

[18] The first and second authors of this paper.

use (task) and the data source of the bot edits which were approved or closed as successful.

In the first phase, we categorized 30 randomly chosen RfPs collaboratively to develop a shared understanding. Based on the first categories, we carried out a second round in which another set of randomly chosen 30 RfPs were categorized individually. We then compared the results and checked the agreement level[19]. After discussing diverging cases and cross-validating our category set, we continued with another round of categorizing the data (40 RfPs) individually, and we had 39 agreements out of the 40 cases. We continued with further rounds of coding individually; we met frequently on a regular basis and discussed new or unclear cases and cross-validated our category sets to ensure the consistency of our data classification.

We developed a codebook[20] as the primary reference of the classification process by starting from the first phase and updating it continually during the classification period. The codes are structured in verb-noun pairs denoting the content entities and operations, which are inspired by the work of Müller-Birn [15] and initially drafted after the first 30 RfP classifications. Based on this, we developed it further by structuring all newly presented information into the verb-noun pairs and adding them to the codebook. The first category describes the potential scope of bot activity. The sub-categories contain all the entity types of Wikidata, such as items, claims, statements, references and sitelinks. The unknown category refers to the tasks without a clear definition. We developed an overview which shows a scheme of all entity types and their respective activities together with the number of requests for these edits (cf. Table 3).

Another dimension of our classification considers the origin of the data which the bot operators plan to include in Wikidata. This category contains three main sub-categories: Internal, external and unknown sources. Internal sources are those within the Wikimedia sphere, whereas external sources are outside Wikimedia. Those requests without a specific source of data were categorized as unknown.

After coding all successful RfPs, we became curious about certain RfPs, which were not approved. Were there some unusual task requests or operators having difficulties in convincing the community of their bots' safe activities? To find this out, we did another process of coding which resulted in 81 unsuccessfully closed RfPs. We classified these RfPs as before except we included the reasons for the request failure. The category *unsuccessful*, thus, contains all RfPs that are not approved for reasons such as, being duplicates, inactive for a long time, not in line with bot policy or withdrawn (cp. Table 6).

4 RESULTS

We organized the results of our RfP analyses in two sections: Firstly, we focused on the approved bots requests and their contents and, secondly, we looked into the details of highly discussed task approval requests.

Table 3: Task requests categorized into edit focus, activity type and the request result (approved/denied) (n = 681, number of all task requests).

Edit Focus	Activity	Approved	Denied
Item	add	85	12
	update	40	3
	merge	6	1
	mark-deleted	1	0
Term	add	22	5
	update	11	0
Label	add	31	3
	update	15	1
	delete	1	0
Description	add	33	5
	update	10	1
Alias	add	4	0
	update	1	0
	delete	1	1
Statement	add	141	19
	update	15	0
	delete	1	1
Claim	add	161	18
	update	33	7
	delete	13	3
Qualifier	add	7	0
Reference	add	9	3
	update	4	0
	delete	1	0
Rank	update	1	0
Sitelink	add	52	3
	update	23	4
	delete	1	0
	merge	1	0
	revert	1	0
Badge	add	4	0
Page	add	0	1
	update	8	1
	delete	1	0
Community	maintain	25	5
Misc.	−−	26	13

4.1 Approved Requests

In this part, we show the types of activities which were approved, the sources they used for editing, and the relationship between bots and their operators. We focus on three aspects of the tasks: The activity focus, the activity type and the origin of the data used by the bots.

4.1.1 Activity Type and Focus. The activity focus considers what the bots have planned to edit in Wikidata. It can be seen in Table 3 that most task requests aim to edit items in Wikidata. The different categories represent the different parts of an item which are organized into terms, i.e. labels, descriptions and aliases, statements

[19] We measured the inter-rater reliability by using Cohen's kappa. At this stage, we had a substantial agreement level, i.e. 0.65.

[20] The codebook along with the data is provided on GitHub: https://github.com/FUB-HCC/wikidata-bot-request-analysis.

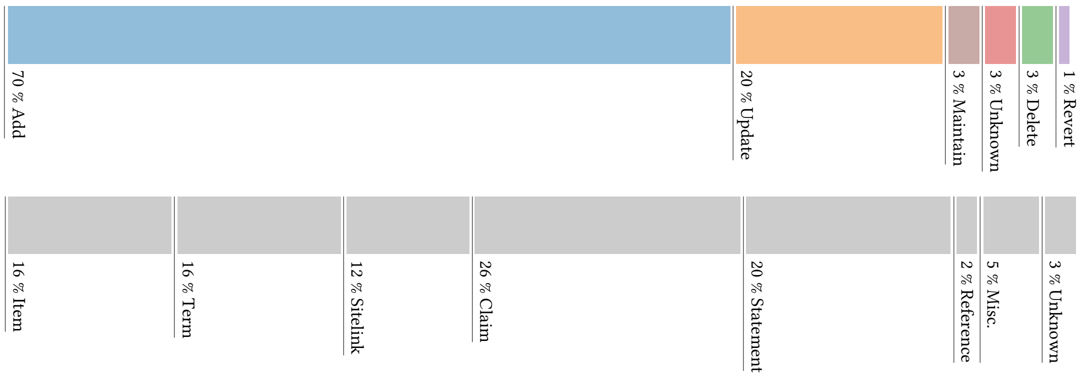


Figure 2: Approved task requests organized into activity type (the upper half part) and edit focus (the lower half part) (n = 794, 262 RfPs are requests with multiple tasks, edit focus *Term* consists of *Alias, Label and Description*. The edit focus *Miscellaneous* contains the smaller categories, such as *Qualifier*, *Rank*, *Badge* and *Maintenance related data*).

(i.e. claims with qualifiers), references, ranks and sitelinks with badges[21]. The number of approved task requests show that the majority of bots seem to focus on adding data to statements in general and claims more specifically. Figure 2 shows the distribution of approved tasks into activity types and edit focuses. Again, the majority of task requests deal with the addition of data.

Furthermore, there are 30 requests dealing with community concerns regarding maintenance activities, such as archiving pages/ sections, moving pages, revert edits, remove duplicates, generate statistical reports as some of the examples.

In addition to this, there are 24 requests concluded as unknown tasks, half of which are written extremely briefly[22] or in an unclear way[23], so that it was difficult for us to identify their potential tasks. Some other cases can only be classified to a specific task based on particular assumptions, for instance, ShBot[24] uses a Bot-Framework QuickStatements[25] to batch edits, which shows a great possibility of importing data to statements. However, this tool could also be used to remove statements or importing terms, so, there is still a challenge to identify the requests' focus without analyzing the edits. We categorized all tasks abiding strictly by our codebook, thus, all the RfPs which needed assumptions are therefore classified as unknown.

We assumed that the data addition to Wikidata is related primarily to Wikidata's inception, i.e. that the majority of task approval would be more at the beginning of Wikidata's lifetime. However, Figure 3 shows that the most often requested activity types in the first six months were "add" and "update". At the end of 2017, these two activity types were often requested again. We found the peak in 2017 for requesting tasks such as adding and updating items unusual and investigated it further. Two operators[26] carried out 31 requests in 2017 for importing items for different languages. Since all these requests were permitted, it explains the sudden acceleration of importing and updating tasks shown in the graph.

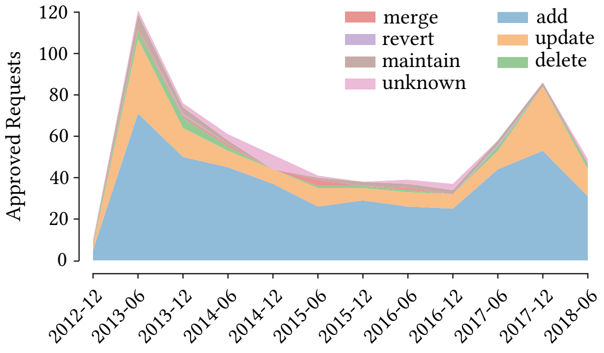


Figure 3: Activity types of approved task requests over time (n = 600, all 81 unsuccessful task requests were not included).

In the following step we looked more closely at the sources of the data which bots are supposed to add to Wikidata.

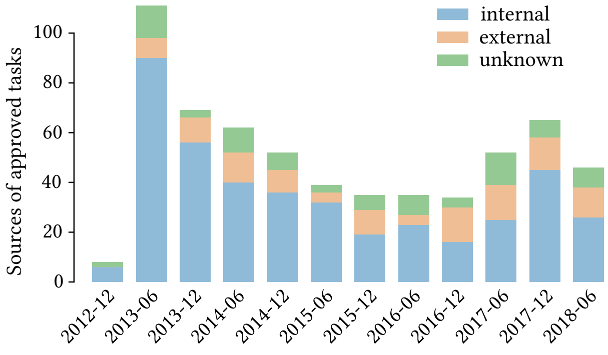


Figure 4: Sources of approved task requests over time (n = 600, all 81 unsuccessful task requests were not included).

[21] These badges present if a sitelink to a specific article on Wikipedia language is an excellent article.

[22] e.g. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/Glavkos_bot

[23] e.g. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/BotMultichillT

[24] www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/ShBot.

[25] Further information are available at meta.wikimedia.org/wiki/QuickStatements.

[26] Both editors are employed by Wikimedia Sverige and were engaged in a GLAM initiative.

Table 4: All internal, the top 10 external sources most used and unknown sources.

Origin	Source	Task Requests
Internal (455)	Wikipedia	263
	Wikidata	110
	Wiki-loves-monuments	30
	Wikicommons	13
	Wikisource	5
	Wikivoyage	4
	Wikinews	2
	Wikiquote	2
	WikiPathways	2
	Wikimedia Project	24
External (38)	MusicBrainz	8
	VIAF	7
	OpenStreetMap	4
	DBpedia	4
	Freebase	4
	GRID	3
	GND	2
	GitHub	2
	YouTube	2
	Disease Ontology Project	2
Unknown	Unknown	107

4.1.2 Data Origin. In addition to the types of tasks in RfPs, the sources from which data are imported also exist in the RfP. We tried to identify all sources independently of the focus or activity type in the classification of all task approval pages. We differentiate internal, external and unknown sources:

(1) Internal: including all sources from the Wikimedia ecosystem,
(2) External: All sources outside Wikimedia’s ecosystem, and
(3) Unknown: All cases where the source of data was a local file/database or not clearly stated.

Table 4 shows the number of tasks approved per data source. The majority of approved task requests (454) deal with data from the Wikimedia ecosystem, with Wikipedia providing most of the data. In addition to Wikipedia, the Wiki Loves Monuments (WLM) initiative, is organized worldwide by Wikipedia community members, and Wikimedia projects. The category Wikimedia project refers to requests where the operator provided information on the source being restricted to Wikimedia projects’ scope, however, a specific attribution to one of the project was not possible.

Another important source refers to Wikidata itself. We found that only a small number of these data are used in maintenance tasks[27]; the majority of requests are for tasks, such as retrieving information from Wikidata, adding descriptions, and updating or adding new labels by translating the existing labels in other languages.

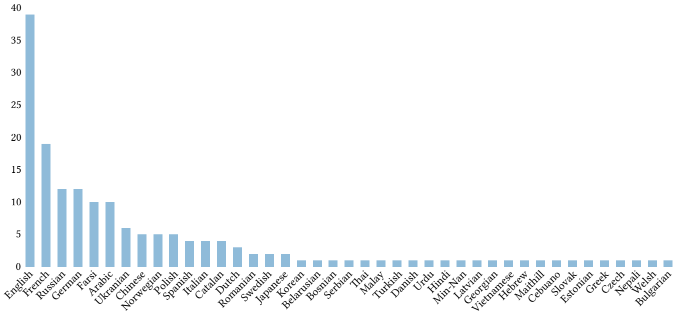


Figure 5: Wikipedia language versions used as sources mentioned on task approval pages (*n* = 195, all 488 other requests either did not provide a specific Wikipedia language version or did not have Wikipedia as the source).

There are 128 task approval requests that relate to a task of importing external data into Wikidata. The external sources most often mentioned are MusicBrainz[28] and VIAF[29] (cf. Table 4). Both sources are used mainly to add identifiers to Wikidata items. Other external sources are OpenStreetMap[30], DBpedia[31] and Freebase[32].

As shown in Figure 4, the internal data sources have remained those most often mentioned in the task requests. External sources and unknown sources are on the same level and external sources show a slight increase over time.

Most data from Wikipedia comes from 43 different language versions (cf. Figure 5) with English, French, German, Russian and Persian being the top five languages. The results show that bots contribution to certain languages had made these languages more prominent than others in Wikidata.

There is a total of 109 RfPs in total with unknown sources; 85 of them were approved; some of these requests mentioned a local file or database as a data source. Other RfPs are not adding data to Wikidata, but are, for example, moving pages.

4.2 Disputed Requests

There are 81 RfPs in our dataset which were closed unsuccessfully. We wanted to understand better why the community decided to decline these requests, thus, we investigated the main reasons for all unsuccessful RfPs.

Our data show that operators themselves are most commonly responsible for their RfPs being rejected. Most of the time, operators were not responding to questions in a timely manner or did not provide enough information when required. There are only a few cases where RfPs were not approved by the community, i.e. no community consensus was reached or the bot violated the bot policy. In one case[33], for instance, an editor was asking for bot rights for its own user account instead of the one for the bot. This

[27] Among 104 requests with a data source of Wikidata, there are 16 requests holding tasks of maintenance, such as archive discussions or update database reports.

[28] MusicBrainz is an open and publicly available music encyclopedia that collects music metadata, available at www.musicbrainz.org.

[29] VIAF is an international authority file which stands for Virtual International Authority File, available at www.viaf.org.

[30] OpenStreetMap is an open license map of the world, available at: www.openstreetmap.org.

[31] DBpedia is also a Wikipedia-based knowledge graph wiki.dbpedia.org.

[32] Freebase was a collaborative knowledge base which was launched in 2007 and closed in 2014. All data from Freebase was subsequently imported into Wikidata.

[33] www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/Florentyna.

Table 5: Top 5 most edited task requests closed as unsuccessful.

Reasons	No. of RfP Edits	No. of Editors	RfP Created At	Reference
Data source, introduced errors	88	10	2018-03	[26]
Bot name, bot edits, bot performance	43	6	2015-12	[24]
Automatic adding of implicit statements	41	21	2013-04	[23]
Support from less active users, duplicate task	32	16	2018-03	[27]
Bot name, conflict with users, running without flag	32	11	2014-10	[25]

violates bot policy as operators are required to create a new account for their bots and include the word "bot" in the account name.

Table 6 shows the main reasons why tasks were rejected: RfPs which had already been implemented by other bots (duplicate), or RfPs requesting for tasks which were not needed, such as one bot which wanted to remove obsolete claims related to property (P107)[34] but there were no more items associated with P107.

Furthermore, RfPs were closed as unsuccessful because the community could not trust on the operators' behaviors. One bot, for example, has requested many RfPs which were not accepted. The community was questioning its editing behavior and required this user to gain community trust first. It can be seen that unsuccessful closing is not only related to the task types that users had requested.

In the following, we describe six RfPs which were edited and discussed most of all (cf. Table 5). We defined the number of edits on an RfP page as a proxy for "challenging" cases.

Phenobot [26] was asking for correcting species names based on the UniProt Taxonomy database. The community doubted the data source (UniProt Taxonomy database) and insisted on more reliable information. The bot has introduced some error reports, therefore, the community had to ask for edit-rollbacks. Since then the operator has been inactive for two years which, finally, led to this request being closed unsuccessfully.

WikiLovesESBot [27], which wanted to add statements related to Spanish Municipalities from Spanish Wikipedia, gained continuous support from nine users until a user asked for more active Wikidata users to comment. After this, one active user checked the bot contributions and pointed out that this bot was blocked. Moreover, another active user figured out that this request was asking for duplicate tasks since there was already another user also working on Spanish municipalities. The RfP remained open for about half a year without any operator activities and came to a procedural close.

Another remarkable case is VlsergeyBot [25], which intended to update local dictionaries, transfer data from infobox to Wikidata, and update properties and reports in Wikidata. At first, the account was named "Secretary" which was against the bot policy. After a community suggestion, it was renamed to "VlsergeyBot." After that, a user opposed the request and said that "Vlsergey‘s activity generates a number of large conflicts with different users in ruwiki" and then another user stated that "maybe you have a personal conflict with Vlsergey" and discussions looked like a personal conflict. The community, then, considered whether this account needs a bot flag, and found that the bot was already running without a bot flag for hours and even broke the speed limit for bots with a flag. The operator promised to update his code by adding restrictions before the next run. Finally, the operator withdrew his request stating that he "does not need a bot flag that much".

ImplicatorBot [23] was intended to add implicit statements into Wikidata automatically. Implicated statements are relationships between items which can be inferred from existing relationships. A property of an item stating that a famous mother has a daughter implies, for example, that the daughter has a mother, even though it is not represented yet. The task sounds quite straightforward, but the community was hesitating to approve the task considering the possibility of vandalism. Furthermore, changes in properties are still common, and such a bot might cause tremendous further work in the case of those changes. Even if the operator had shared his code, discussions proceeded with the highest number of editors discussing a request (i.e. 21) and finally, the operator stated that he is “sick and tired of the bickering.” Despite the community insisting that they needed the task, the operator withdrew it.

Similar to ImplicatorBot, Structor [24] was withdrawn by the operator because the operator was "tired of bureaucracy." The bot wanted to add claim, label and description, particularly to provide structured information about species items. There were many questions raised by a small number of community members (i.e. a total of six different users participated in the discussion), including the bot name, the source of edits, duplicate entries, the edit summary, and edit test performances. The operator was actively responding to all these questions. However, the whole process was so surprisingly complex (the discussions continued for over seven months from the start, that the operator was inactive for a long time, then withdrew his request.

Requests denied because the user is not trusted, are also interesting. ElphiBot_3[35] [22], for example, which is managed together by three operators, sent a request along with the source code for updating the interwiki links in Wikidata after the category in Arabic Wikipedia has been moved. The community preferred, however, someone who is more responsible and more familiar with the technical aspects of Wikipedia and Wikidata. What concerned the discussants most was that the operators were not capable enough to fix the bugs introduced by their bot. The community did not feel comfortable approving this request because the operators' treatment for fixing bugs was simply to reverse the mistakes manually. The requirement to run a bot should not only be to know how to code but surely also to respond in a timely manner and notify those who can fix the issue.

[34] P107 was a property representing GND type.

[35] This RfP has a total of 29 edits and 8 editors contributed in the discussions.

Table 6: Main reasons for unsuccessful task requests (*n* = 81, number of unsuccessful task requests).

Reason	Description	Frequency
No activity from operator	Operators are not active or do not respond to the questions regarding their bots.	35
Withdrawn	Operators wanted to close the requested task.	14
Duplicate	Requested tasks are redundant and are already done through other bots or tools.	9
No community consensus	Community opposed the requested tasks.	6
User not trusted	Operator has questionable editing behavior and ability to fix any bugs introduced.	6
Task not needed	Tasks do not need a bot or are not a significant problem for the time being.	5
Don't need bot flag	The requested tasks can be performed without a bot flag.	3
No info provided	The fields in the RfP page are left empty.	2
Against bot policy	Task is not in line with bot policy.	1

5 DISCUSSION

In this study, we collected existing data from the RfP pages on Wikidata to find out what kind of tasks the community allows to be automated or shared with bots.

Looking at the type and focus of the tasks which were mentioned most often in RfPs, we found that the dominant request type over time is "adding data." This observation suggests that Wikidata's community uses bots to increase the coverage of the provided data which aligns with the vision of the Wikimania Foundation "Imagine a world in which every single human being can freely share in the sum of all knowledge." [36]. Updating data is the second highest requested task which shows bots also take part in correcting mistakes or refreshing data according to changes (e.g., updating the sitelinks of moved pages) and contribute in data completeness, which is defined as one data quality dimension [2].

However, we were surprised that there were fewer requests regarding adding or updating references which could support the trustworthiness, a data quality dimension as well, of data imported by bots. This result supports existing analyses on the edit activities of bots [18]. However, it would be interesting to bring these two lines of research together - the task approval and edit perspective - to understand more closely the development over time. Thus, we can infer that bots are allowed to assist the Wikidata community in ensuring data quality from the completeness angle, however, this is less visible from the trustworthiness angle. In comparison to Wikipedia, where bots perform primarily maintenance tasks, bot requests in Wikidata concentrate mainly on the content, i.e. data perspective.

The majority of data sources mentioned in RfPs come from inside Wikimedia projects, mainly Wikipedia (cf. Section 4.1.2). This observation implies that Wikidata is on its way to serve as the structured data source for Wikimedia projects, one of the main goals for the development of Wikidata. Among the five language versions of Wikipedia most used: English, French, Russian, German and Persian (cf. Figure 5), the first three languages show a dominance of Western knowledge in bots imports. In a study, Kaffee et al. [11] had earlier found the influence of western languages in Wikidata, with the most dominant ones being English, Dutch, French, German, Spanish, Italian and Russian. The similarity of the results in both studies could imply that the reason these languages have better coverage than other languages is a result of having bots. Kaffee et al. also found that language coverage is not related to the number of speakers of a language, which supports our assumption. We could not find evidence that languages other than Western ones are underrepresented on bot requests on purpose. We rather expect that the current bot requests are a representative sample of Wikidata's community.

We can see from the external sources that bots use these sources mostly for importing identifiers (e.g. VIAF) or for importing data from other databases (e.g. DBpedia, Freebase). This insight supports Piscopo's [17] argument that bot edits need to be more diverse. We suggest that further efforts should be made to import or link data from different sources, for example from research institutions and libraries. With the increased usage of the Wikibase software[37], we assume that more data might be linked or imported to Wikidata. Our classification revealed that bots import data from local files or databases already. However, such data imports often rely on the community trusting the bot operators and do not seem large-scale.

The requested maintenance tasks within the RfP show similar patterns to those in the Wikipedia community. Bots are being used to overcome the limitations of the MediaWiki software [9]. An administrative task, such as deletion, is not often requested in RfPs. However, bots on Wikidata are allowed to delete sitelinks only. Similar to Wikipedia, the Wikidata community comes closer to a common understanding of what bots are supposed to do and what not.

The issue of unknown tasks, which was not clearly defined in RfPs, shows the trust the community has in single operators, probably due to their previous participation history. There is a noticeable number of approved RfPs which we coded as unknown due to their vague task description, while there are also cases where task descriptions were clear, but the community did not trust the operators and, thus, they were not approved. These two observations indicate that trust is also given importance by the community in addition to their defined policy and procedures.

The success rate of RfPs is relatively high since only a small number of RfPs are closed as unsuccessful. Our findings show that among the 81 unsuccessful RfPs, only some of the RfPs were rejected by the community directly; the majority of them were unsuccessful due to the reason that the operators were not responding or had

[36] Information on Wikimedia strategy can be found on: https://wikimediafoundation.org/about/vision/.

[37] Wikibase is based on MediaWiki software with the Wikibase extension https://www.mediawiki.org/wiki/Wikibase.

withdrawn the request. Therefore, operator's inactivity is a higher reason for failure than community refusal. In some cases (i.e. the RfPs discussed most) we can see that the community considers every detail of the tasks and then comes to a decision, however, in some cases, they approve the request without a detailed discussion, as can be seen in the cases of unknown tasks and unknown sources. This result could indicate that in addition to the defined procedure for RfPs, the community applies a more flexible approach when deciding on RfPs considering the context (e.g. editors experience) of the application into account.

In summary, the high approval rate of RfPs shows that the Wikidata community is positive towards bot operations in Wikidata and willing to take advantage of task automation through bots. Wikidata's community profited from the experiences of the Wikipedia community and built productive human-bot processes in quite a short period. However, we will leave the impact of these processes on data quality to future research.

6 CONCLUSION

We studied the formal process of requesting bot rights in Wikidata to find out what kind of task types are allowed to be automated by bots. Our study provides a detailed description of the RfP process. We retrieved the closed RfPs from the Wikidata archive up to mid-2018. We defined a scheme, in the form of a codebook, to classify the RfPs and developed our dataset. The RfPs were studied mainly from two perspectives: 1) What information is provided during the time the bot rights are requested and 2) how the community handles these requests. We found that the main tasks requested are adding claims, statements, terms and sitelinks into Wikidata, as well as the main source of bot edits have their roots in Wikipedia. This contrasts with Wikipedia where bots are performing mostly maintenance tasks. Our findings also show that most of the RfPs were approved and a small number of them were unsuccessful mainly because operators had withdrawn or there was no activity from the operators.

Future directions of this research will focus on analyzing the actual bot activities and comparing them with the results of this study to find out whether bots are really performing the tasks that they had asked for. This could help us understand better how bots evolve over time and whether the community controls bot activities based on their permitted tasks, or bots can perform any edit once they get the permission to operate as a bot.

ACKNOWLEDGMENTS

We thank the Deutscher Akademischer Austauschdienst (DAAD) [Research grant: 57299294], and the German Research Foundation (DFG) [Fund number: 229241684] who supported this research financially, and the "anonymous" reviewers for their valuable comments which helped to improve the paper.

REFERENCES

[1] Emilio Ferrara, Onur Varol, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (2016), 96–104.

[2] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web* 9, 1 (Jan. 2018), 77–129. https://doi.org/10.3233/SW-170275

[3] R. Stuart Geiger. 2009. The social roles of bots and assisted editing programs. In *Int. Sym. Wikis.*

[4] R Stuart Geiger. 2012. Bots are users, too! Rethinking the roles of software agents in HCI. *Tiny Transactions on Computer Science* 1 (2012), 1.

[5] R. Stuart Geiger and Aaron Halfaker. 2013. When the levee breaks: without bots, what happens to Wikipedia's quality control processes?. In *Proceedings of the 9th International Symposium on Open Collaboration (WikiSym '13).* ACM, New York, NY, USA, 6:1–6:6. https://doi.org/10.1145/2491055.2491061

[6] R Stuart Geiger and Aaron Halfaker. 2017. Operationalizing conflict and cooperation between automated software agents in Wikipedia: a replication and expansion of "even good bots fight". *ACM Hum.-Comput. Interact.* 1, 2 (Nov. 2017), 33. https://doi.org/10.1145/3134684

[7] R. Stuart Geiger and David Ribes. 2010. The work of sustaining order in Wikipedia: the banning of a vandal. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10).* ACM, New York, NY, USA, 117–126. https://doi.org/10.1145/1718918.1718941

[8] Aaron Halfaker, R Stuart Geiger, Jonathan T. Morgan, and John Riedl. 2013. The rise and decline of an open collaboration system: how Wikipedia's reaction to popularity is causing its decline. 57(5) (2013), 664–688. https://doi.org/10.1177/0002764212469365

[9] Aaron Halfaker and John Riedl. 2012. Bots and cyborgs: Wikipedia's immune system. *Computer* 45, 3 (March 2012), 79–82. https://doi.org/10.1109/MC.2012.82

[10] Andrew Hall, Loren Terveen, and Aaron Halfaker. 2018. Bot detection in Wikidata using behavioral and other informal cues. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 1–18. https://doi.org/10.1145/3274333

[11] Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. 2017. A glimpse into Babel: an analysis of multilinguality in Wikidata. In *Proceedings of the 13th International Symposium on Open Collaboration (OpenSym '17).* ACM, New York, NY, USA, 14:1–14:5. https://doi.org/10.1145/3125433.3125465

[12] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research Methods in Human-Computer Interaction* (2nd ed.). Morgan Kaufmann.

[13] Brenda Leong and Evan Selinger. 2019. *Robot Eyes Wide Shut: Understanding Dishonest Anthropomorphism.* ACM, New York, New York, USA.

[14] Randall M. Livingstone. 4 January 2016. Population automation: An interview with Wikipedia bot pioneer Ram-Man. *First Monday* 21, 1 (4 January 2016). https://doi.org/10.5210/fm.v21i1.6027

[15] Claudia Müller-Birn. 2019. *Bots in Wikipedia: Unfolding their duties.* Technical Reports Serie B TR-B-19-01. Freie Universität, Berlin. https://refubium.fu-berlin.de/handle/fub188/24775

[16] Claudia Müller-Birn, Benjamin Karran, Janette Lehmann, and Markus Luczak-Rösch. 2015. Peer-production system or collaborative ontology engineering effort: What is Wikidata?. In *Proceedings of the 11th International Symposium on Open Collaboration.* ACM, 20.

[17] Alessandro Piscopo. 2018. Wikidata: A New Paradigm of Human-Bot Collaboration? *arXiv:1810.00931 [cs]* (Oct. 2018). http://arxiv.org/abs/1810.00931 arXiv: 1810.00931.

[18] Alessandro Piscopo, Lucie-Aimée Kaffee, Chris Phethean, and Elena Simperl. 2017. Provenance information in a collaborative knowledge graph: an evaluation of Wikidata external references. In *The Semantic Web – ISWC 2017 (Lecture Notes in Computer Science).* Springer, Cham, 542–558. https://doi.org/10.1007/978-3-319-68288-4_32

[19] Alessandro Piscopo, Chris Phethean, and Elena Simperl. 2017. What makes a good collaborative knowledge graph: group composition and quality in Wikidata. In *Social Informatics (Lecture Notes in Computer Science)*, Giovanni Luca Ciampaglia, Afra Mashhadi, and Taha Yasseri (Eds.). Springer International Publishing, 305–322.

[20] Thomas Steiner. 2014. Bots vs. Wikipedians, Anons vs. Logged-Ins (Redux): a global study of edit activity on Wikipedia and Wikidata. In *Proceedings of The International Symposium on Open Collaboration (OpenSym '14).* ACM, New York, NY, USA, 25:1–25:7. https://doi.org/10.1145/2641580.2641613

[21] Margaret-Anne Storey and Alexey Zagalsky. 2016. Disrupting developer productivity one bot at a time. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016).* ACM, New York, NY, USA, 928–931. https://doi.org/10.1145/2950290.2983989

[22] Wikidata. 2013. RfP ElphiBot_3. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/ElphiBot_3 [Last accessed 01 August 2018].

[23] Wikidata. 2013. RfP ImplicatorBot. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/ImplicatorBot [Last accessed 01 August 2018].

[24] Wikidata. 2014. RfP Structor. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/Structor [Last accessed 01 August 2018].

[25] Wikidata. 2014. RfP VlsergeyBot. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/VlsergeyBot [Last accessed 01 August 2018].

[26] Wikidata. 2016. RfP Phenobot. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/Phenobot [Last accessed 01 August 2018].

[27] Wikidata. 2016. RfP WikiLovesESBot. www.wikidata.org/wiki/Wikidata:Requests_for_permissions/Bot/WikiLovesESBot [Last accessed 01 August 2018].