The Roles Bots Play in Wikipedia

LEI (NICO) ZHENG, Stevens Institute of Technology, USA
CHRISTOPHER M. ALBANO, Stevens Institute of Technology, USA
NEEV M. VORA, Stevens Institute of Technology, USA
FENG MAI, Stevens Institute of Technology, USA
JEFFREY V. NICKERSON, Stevens Institute of Technology, USA

Bots are playing an increasingly important role in the creation of knowledge in Wikipedia. In many cases, editors and bots form tightly knit teams. Humans develop bots, argue for their approval, and maintain them, performing tasks such as monitoring activity, merging similar bots, splitting complex bots, and turning off malfunctioning bots. Yet this is not the entire picture. Bots are designed to perform certain functions and can acquire new functionality over time. They play particular roles in the editing process. Understanding these roles is an important step towards understanding the ecosystem, and designing better bots and interfaces between bots and humans. This is important for understanding Wikipedia along with other kinds of work in which autonomous machines affect tasks performed by humans. In this study, we use unsupervised learning to build a nine category taxonomy of bots based on their functions in English Wikipedia. We then build a multi-class classifier to classify 1,601 bots based on labeled data. We discuss different bot activities, including their edit frequency, their working spaces, and their software evolution. We use a model to investigate how bots playing certain roles will have differential effects on human editors. In particular, we build on previous research on newcomers by studying the relationship between the roles bots play, the interactions they have with newcomers, and the ensuing survival rate of the newcomers.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**.

Additional Key Words and Phrases: Wikipedia; bots; roles; taxonomy; governance; online communities

ACM Reference Format:
Lei (Nico) Zheng, Christopher M. Albano, Neev M. Vora, Feng Mai, and Jeffrey V. Nickerson. 2019. The Roles Bots Play in Wikipedia. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 215 (November 2019), 20 pages. https://doi.org/10.1145/3359317

1 INTRODUCTION

Bots have become increasingly important in aiding human work. They help us collect data [46], produce online news [29], make automatic responses [23], and design products [42]. They are designed to perform many tasks that humans are capable of completing, but they amplify human effort in both speed and scale [41]. Together, humans and bots form an ecosystem, in which they adapt to and learn from each other. Usually, ecosystems are based on the transformation of energy: here, by analogy, the focus is on the transformation of information. That is, the knowledge chain, as opposed to the food chain, relates to the way collective knowledge is built up from smaller

Authors' addresses: Lei (Nico) Zheng, Stevens Institute of Technology, 1 Castle Point Terrace, New Jersey, USA, lzheng9@stevens.edu; Christopher M. Albano, Stevens Institute of Technology, 1 Castle Point Terrace, New Jersey, USA, calbano@stevens.edu; Neev M. Vora, Stevens Institute of Technology, 1 Castle Point Terrace, New Jersey, USA, nvora1@stevens.edu; Feng Mai, Stevens Institute of Technology, 1 Castle Point Terrace, New Jersey, USA, fmai@stevens.edu; Jeffrey V. Nickerson, Stevens Institute of Technology, 1 Castle Point Terrace, New Jersey, USA, jnickers@stevens.edu.

2573-0142/2019/11-ART215
https://doi.org/10.1145/3359317

units of knowledge contributed by both humans and bots. This knowledge is built in a process that is mediated through the artifacts, the units of knowledge represented as textual fragments that are modified and composed to create sections, articles, and projects. This process is complex, and hence difficult to decompose and analyze. One approach involves subdividing bots into categories; given the dynamic nature of ecosystems, ongoing monitoring and analysis would be facilitated by automatic classification. By analogy to recent work on the roles of editors [52, 54, 56], we explored the roles of bots and their influence on humans and human work on the English version of Wikipedia. Specifically, we developed a method to identify bots' roles and, to illustrate the use of the taxonomy, tested how bots with different roles affect the retention of new editors.

Bots are extensively used in online communities, the Wikipedia community being a salient example. In 2009, bots and assisted editing programs comprised 28.49% of all English Wikipedia edits [13]. Continuously taken statistics focused only on bot edits currently show a proportion of about 10% [1]. In Wikidata, this proportion has reached 88% [43]. Wikidata, a large knowledge graph, is intended to aid the generation of Wikipedia articles in all languages by both bots and people [51]. Wikidata is relevant to this study because increased reliance on it has affected bot functions. In particular, Wikidata facilitated a shift in policy related to inter-language linkages [16]. We show this led to the reduction of bots we classify as connectors. More generally, Wikidata may be contributing to consolidation by obviating the need for certain bots or facilitating the creation of multi-function bots.

Bots are designed to aid a variety of tasks, including fixing redirect links, countering vandalism, and recommending tasks to editors [11, 14, 15, 21]. They are designed for different purposes, including the improvement of productivity and the preservation of order [17]. They are built to provide different kinds of support by alternatively focusing on content, task, and community [34]. Since there are complex interactions involved, the operation of bots also introduces a variety of effects that relate to trade-offs between quality, productivity, creativity and user engagement [15, 16, 18, 21, 49]. Thus, there is a need to better understand these automated tools: what they consist of, how they function, and how they interact with humans. Wikipedia bots are amenable to study because they are governed transparently, and because records of bot actions are publicly available. By analyzing Wikipedia bots we may gain insight into bots in other contexts that are less transparent (for example, chat bots), and, more generally, into the automated tools that are being used in industries ranging from chip design to video game design [42].

In this study, building on previous works that code bot activity types manually [21, 34], we develop a taxonomy of bot roles in English Wikipedia using machine learning algorithms. We start with a search procedure to align bots with their functions. We proceed by identifying the bots' roles with respect to their functions; we use a graph-based clustering algorithm to cluster similar bot functions. We use the identified bot roles as our taxonomy for classification. We then apply a multi-class classifier to classify Wikipedia bots into different roles. We discuss different bot activities, including their edit frequency, their working spaces, and their software evolution. We demonstrate the applicability of the taxonomy by using the derived bots' roles to predict newcomers' survival rates. Finally, we conclude by discussing ways of extending the study.

Our work seeks to study and document bot functions and bot roles systematically on English Wikipedia. The code we developed to analyze bots is available for researchers to use.[2] The automatic classifier we developed can be adapted to identify bot roles on non-English Wikipedias by adjusting the textual features to the target language. Even though the derived taxonomy focuses on bots that were designed to aid collaborative editing, the procedure we propose could be used in other

[1] https://stats.wikimedia.org/EN/EditsRevertsEN.htm
[2] Source code: https://github.com/Nicozheng/Wikipedia_bots_taxonomy

online communities in which bots have different functions. Furthermore, the creation of a bot taxonomy can facilitate three types of work. First, it can support studies of the effects of bots on human work. With bots classified into different categories, we can study not only a single bot, but also certain types of bots, or even the entire multi-bot system. The type of a bot may be a cause or moderator: for example, some types of bots may motivate human editing behavior, and by contrast other types of bots may drive humans away. Second, a taxonomy can provide a foundation for a better governance framework that improves information transparency and collaborations between bot developers, while also detecting novelty. Novel bots might be useful, and therefore good to recognize early. Or they may be malicious, in which case early detection will also be beneficial. Third, with respect to the effects of bots, we show how different bots within the same role category can have different effects on newcomer survival rates. That is, a taxonomy provides a way for analysis to focus on particular parts of the design space. More generally, a taxonomy can also serve as a building block in human-computer system design. That is, a taxonomy can be used to understand how design decisions made by humans alter the functions and roles of bots in a particular design subspace. This in turn might guide future design decisions.

2 ROLE THEORY

The Wikipedia bots of today are human delegates rather than fully autonomous agents that modify their own purposes. That is, bots reflect the purposes of their creators. They play roles not of their own choosing, but of the choosing of their designers. Studying what the bots do is to a great degree studying what their designers want to accomplish. With this in mind, we impute roles to the bots; these roles should be understood to be heavily tied to the roles human Wikipedia editors choose to play; in playing these roles, they create bots to assist them.

Role theory can be used to explain both behavior and motivation [7, 30]. It also provides a structure for studying coordination, the division of labor, and the allocation of tasks in communities and organizations [4, 26]. Existing studies on identifying bot roles in online communities rely on functional role theory [25, 44, 53]. For example, Storey and Zagalsky identify five bot roles that are frequently used in software development – code bots, test bots, devops bots, support bots and documenting bots – by reasoning that bots are used in different stages of development [44]. Similarly, Wessel et al. classify Github bots into thirteen categories according to the tasks they perform [53]. By contrast, Seering et al. use structural role theory to classify chatbots in Twitch into five roles including information, moderation messages and warnings, user-engagement, mini-games, and promotion based on the type of messages they send to users [41].

Bot roles have been identified in Wikipedia. Halfaker and Riedl categorized Wikipedia bots activities into four categories including injecting public domain data, monitoring and curating content, augmenting MediaWiki software, and protecting the encyclopedia from malicious activity [21]. Müller-Birn et al. identified 18 bot activity types and manually coded 353 bots into six clusters based on the information listed on their descriptions. The identified activity clusters include editing articles, organizing articles, supporting other bots, supporting editors, supporting communication, and supporting decision-making [34]. These are qualitative studies based on a small sample of bots. Our study extends previous research on bot roles by developing a taxonomy based on functional role theory. We also create a machine learning classifier to automatically classify bots into associated roles based on their edit activities and textual descriptions, making it possible to analyze all bots.

As previous studies on governance suggest, bots are not only software tools but also managerial protocols that are part of the Wikipedia infrastructure [6, 14, 36]. By converting social rules into source code, these automatic machines change the nature of rule enforcement in the community [21, 34]. Through interactions with humans and other bots, bots trigger complex human-bot and bot-bot dynamics [16, 18, 32]. These dynamics can be intricate; an analysis conducted by Halfaker and Geiger [21] found that most bot-bot reverts are not the result of conflict, but rather a result of different clerical procedures. We extend this line of research by studying how bots with different roles affect newcomers' retention. Such within-category comparisons of the effects of bots on human editors may be useful for Wikipedia bot governance as well as for bot design.

3 BOTS IN WIKIPEDIA

In online communities like Wikipedia, a bot is defined as "an autonomous software program which is developed and operated by volunteers" [15]. The first Wikipedia bot appeared in October 2002; it was deployed to add and maintain U.S. county and city articles [28]. As of February 2019, there are 1,601 registered bot accounts. Of those accounts, 25.36% made more than 10 thousand cumulative edits and 24 bots made over 1 million edits in their lifetime. These bots differ with regard to the sophistication of technology employed: some use basic regular expressions or heuristics, while others incorporate machine learning techniques. While it is theoretically possible that one or more roles of an autonomous bot might be self-generated through imitation learning or mutation, currently the roles of most, if not all, Wikipedia bots are determined by their human developers.

Currently, Wikipedia uses a "decentralized, consensus-based model" to regulate bot-related work [16]. Contributors who want to develop and deploy a bot are expected to submit a bot approval request that provides information about the bots' functions, the bots' programming language, and the estimated number of pages affected. Then a Bot Approvals Group (BAG) run by experienced and trusted developers will go over the request and discuss its potential influence. If the bot is deemed to be helpful and follows the community bot policy, it will be approved for a short trial period during which the bot is closely monitored to ensure that it operates correctly. After successfully passing the trial, the bot may be allowed to be fully deployed in Wikipedia. The same procedure is repeated whenever the bot owner wants to add new functions to the approved bot. If the bot misbehaves, an administrator can temporarily block the bot; other editors can also report problems on the talk page of the bot or the bot operator. Operators are obligated to respond to the concerns, and the bot will be revoked if the operators fail to do so.

A recent study of bots in Open Source Software (OSS) projects highlights the accomplishments and challenges of bot usage in the software development process [53]. While bots are commended for automating processes, they are also criticized for giving misleading feedback or taking inappropriate actions. These challenges call for the design of more sophisticated bots or multi-bot systems in which each bot performs specific functions, but can communicate, integrate, build upon each others' work, and learn from each others' experience. For this to happen, there must be collaboration among developers working on related functions with respect to interfaces, processes, and shared code. Moreover, new bot designs may have different impacts on different stakeholders. A fine-grained bot taxonomy might provide a general framework for the analysis of bot impact, governance, and design, and thus could serve as a first step to improve the bot ecosystem in online communities.

4 TAXONOMY OF BOTS

4.1 Bot Role Taxonomy Creation

A role is a bundle of tasks, norms and the behaviors that are expected of those who occupy a position in a social structure [8]. Based on functional role theory, we define a bot role as a bundle of bot functions (see Appendix A for definition), in which bots share similar actions, target objects or goals. Extending previous work that coded bot activities manually [34], we take a two-stage procedure which integrates both human knowledge and algorithms to create a taxonomy. A two-stage procedure is used due to the large design space of bot functions. In the first stage, the first author enumerated bot functions following a simple search procedure and assigned bot-to-function relationships manually. In the second stage, the authors estimated the distances between the functions of the bots, built a network of bot functions, and applied a community detection algorithm to distinguish bot roles. Specifically, we encoded a bot function as a vector of bots v_i using the bot-to-function relationships. In this way, similar functions are exhibited in a set of bots in the same role category, and thus have a similar probability distribution on v_i. We use cosine similarity as our similarity measurement because it is easy to understand in the absence of specific reasons to compute similarity otherwise [22, 35]. In short, we bundle similar bot functions and use them to define roles.

In Wikipedia, bot operators usually list their bots' tasks on the bot's user page in order to introduce the bot to the public audience. Such information can also be found on the bots' Request for Approval pages (BRFA pages). Thus we first retrieved all 1,601 registered bots in Wikipedia as of February 28, 2019 under "Category: All Wikipedia bots." For each bot, we collected its user page and request for approval history as our corpus. Then we enumerated bot functions following a greedy search procedure.

The procedure goes as follows. First, we randomly separate all bots into small chunks (10 bots in each chunk). For each bot in a chunk, we read its user page and bot approval history to extract bot functions manually, looking in particular at the Function Details section of these pages. Then we aggregate similar functions and add new functions into a list of discovered functions. We annotate the bot-to-function relationship as a 1 if the bot has the proposed function, otherwise as a 0. Then we proceed to the next chunk and repeat the above procedures. We stop when no new functions are discovered in a series of five consecutive chunks (50 bots).

As a result of this procedure, we obtain 25 bot functions and 200 bots with their function labels. Then we estimate the distance between bot functions using the function-to-bot relationship. More specifically, each function F_i can be represented as a binary vector $v_i = \{B_1, B_2, B_3, ...\}$, where B_i is 1 if the bot has this function and otherwise 0. Thus the distance $D_{i,j}$ between two functions can be measured using the cosine similarity of their vectors.

We construct a bot functions network based on the function distance measurement. In this network, each node represents a function. There is an edge between two functions if their cosine similarity exceeds a threshold of 0.3. We then apply a community detection algorithm [9] to identify the bots' hidden roles. Figure 1 shows the bot functions network, in which different node colors represent different roles identified by the algorithm. We also refine the function clusters to make the functions assigned within one role more coherent (see Appendix A). There are nine roles and their associated bot functions that are summarized in Table 1. Next, we discuss these roles in detail.

4.2 The Roles of Bots

Generator. These bots generate article pages based on predefined templates. Their functions include generating redirect pages and creating pages based on other sources. Rambot, the very first bot in Wikipedia, was built as a generator to create articles of U.S. cities based on the census data [28]. There are also other similar bots that generate articles about mountains, rivers, and other geographical entities. A different function generates redirect pages rather than geographical entities. For example, when one searches "Apple Tree" in Wikipedia, the term can represent a variety of things, including a plant, a band, and a location. Bots like RussBot will identify related pages based on the ambiguous search term and create a redirect page to guide users.

Fixer. These bots fix errors in article pages in order to keep the information neat and correct. Related functions include fixing links, fixing content, fixing files and fixing parameters in the template, category, and infobox. The function "fix links" is an example: many bots that bypass links to redirect pages, fix double redirects or fix incorrect link formats have this function. Similarly, bots

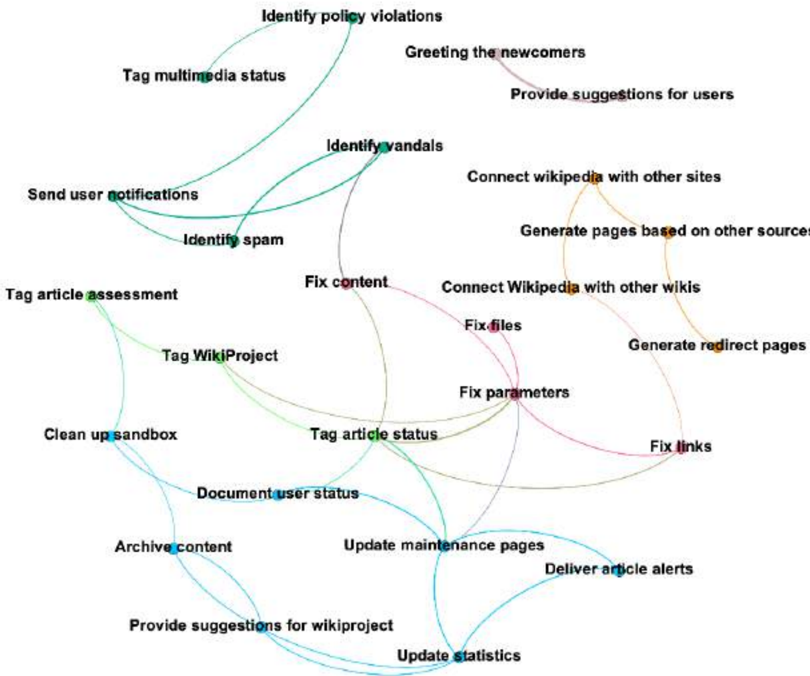


Fig. 1. A bot functions network

like CmdrObot fix typos and spelling errors based on predefined common rules. At the same time, a bot like Yobot would fix biography articles category from "Category: Date of birth missing" to "Category: Date of birth missing (living people)" if the people are indeed alive.

Connector. Some bots are built to connect Wikipedia with other sites and databases. For example, the KasparBot will extract authority control information and move them to Wikidata. The Citation bot will add reference identifiers (DOIs, PMIDs, ISBNs) that are obtained from other sources such as arXiv to references. There are also a large number of interwiki bots which are used to link the same content within different Wikipedia languages. These interwiki bots accounted for many edits until a 2013 policy change, explained below.

Tagger. These bots continuously patrol articles and tag articles with different templates and categories. Templates and categories in Wikipedia are mainly used for administrative purposes [1]. For example, if a statement was added without additional citations for verification, editors will add the citation needed template to the article so that a banner will show up above the content, indicating that the page needs more citations. There are a variety of templates to indicate article status (`{{AFD}}`: article for deletion), article quality and importance (`{{subst:GAR}}`: Good Article), policy violation (`{{COI}}`: conflict of interest), Wikiproject ownerships (`{{WikiProject Biography}}`: page belong to WikiProject Biography), and multimedia status (`{{FFD}}`: file for deletion).

Clerk. These bots do a variety of tasks including updating statistical information, documenting user status, updating maintenance pages, and delivering article alert to Wikiprojects. For example, the "WP 1.0 bot" tracks assessment data and aggregates the statistics into an index page that shows the quality and importance of all rated articles in English Wikipedia. Similarly, there are also bots tracking the status of all the articles in a Wikiproject, calculating statistics, updating maintenance

Table 1. Bot Roles and Associated Bot Functions

Roles	Functions	Example Bot	Cronbach's α
Generator	Generate redirect pages	RussBot	0.877
	Generate pages based on other sources	JJMC89 bot	
Fixer	Fix links	Xqbot	0.956
	Fix content	Yobot	
	Fix files	ImageRemovalBot	
	Fix parameters in template/category/infobox	Yobot	
Connector	Connect wikipedia with other wikis	EmausBot	0.977
	Connect wikipedia with other sites	KasparBot	
Tagger	Tag article status	AnomieBot	0.888
	Tag article assessment	BU RoBOT	
	Tag Wikiprojects	Tom's Tagging Bot	
	Tag multimedia status	Fbot	
Clerk	Update statistics	WP 1.0 bot	0.951
	Document user status	AnomieBot	
	Update maintenance pages	AnomieBot	
	Deliver article alert	alertBot	
Archiver	Archive content	AnomieBot	0.946
	Clean up sandbox	Cyberbot_I	
Protector	Identify policy violations	COIbot	0.888
	Identify spam	XLinkBot	
	Identify vandals	ClueBot NG	
Advisor	Provide suggestions for Wikiprojects	Mathbot	0.777
	Provide suggestions for users	SuggestBot	
	Greeting the newcomers	HostBot	
Notifier	Send user notifications	NoomBot	0.871

pages (like a page lists all articles for deletion) and delivering a summary report (also called an Article Alert) to the Wikiproject.

Archiver. These bots help archive closed discussions and maintain the archived content by assigning an index and sorting them alphabetically. They also help remove content on the user's sandbox every few hours. For example, Lowercase sigmabot III will help users archive talk page threads that are over thirty days old.

Protector. These bots detect and regulate destructive behaviors. Their functions include identifying policy violations, spam, sock puppetry, and vandalism. These bots are the most well-known and studied in the literature [14, 15, 17, 21]. For example, ClueBot NG detects possible vandalism and revert such malicious edits using neural network techniques. The bot is able to identify and revert a possible vandal within seconds, which significantly reduces the website's time-to-revert [15]. COIBot tracks edits that are made by users who may have a conflict of interest. It also tracks links that were reported to spam or COI noticeboard or blacklisted by other protector bots.

Advisor. These bots provide editors with suggestions about articles that they might want to contribute to. For example, Mathbot collects miscellaneous science-related articles from various sources and updates a "missing science topics" list which indicates topic coverage in related Wikiprojects. For the missing topics, it also suggests potential redirects to the related existing articles by matching their names. Similarly, SuggestBot plays an advisor role by maintaining an open task portal and providing personalized suggestions to editors [11]. There are also other bots playing an advisory role, for example, HostBot will greet newcomers, invite them to Wikipedia Tea house and suggest they participate in an online training program [31, 32].

Notifier. These bots deliver messages to editors. The messages can be a system notification or a newsletter about recent activities in Wikipedia. For example, NoomBot will send a notification to the new article reviewers about whether their reviewed article has been deleted; Ralbot will deliver a Wikipedia Signpost (like a newspaper) to its subscribers.

4.3 Comparison of Bot Roles and Human Roles

We relate our findings to previous discussions of the roles of editors [54]. Fixer is similar to the Fact Checker, Wiki Gnomes, and Copy Editor roles, whose work is mainly focused on making article content accurate and fluent. Tagger and Clerk work like the Wikipedian, which is tasked with maintaining the information flow of WikiProjects. Protectors are similar to the Vandal Fighter, while Advisor bots work as the Social Networker to motivate and suggest human editors. However, roles like Generator, Connector, Archiver, and Notifier are particular to bots. Related to a study that classifies human editors by their access privileges [2], we find bots tend to have higher levels of access (level 2 to level 4). For example, the Tagger and Clerk perform Technical Administration, Quality Assurance, and QA Technician roles by patrolling articles, removing invalid files and assessing article quality. The Protector bots perform roles such as Border Patrol and Security Force. This finding is consistent with the view that bots serve to enforce rules in the Wikipedia communities as the community gradually transfers the "right to rule" to bots [13, 34].

A bot can play multiple roles for two reasons. First, the bot may be intentionally programmed to do multiple tasks. For example, AnomieBOT is a bot operated by Anomie, who works at the Wikimedia Foundation and is an active member in the Bot Approval Group. Unsurprisingly, AnomieBOT is extremely productive – there are 69 individual tasks on its user page. Its functions include fixing links, fixing parameters in the template, tagging articles and updating maintenance pages. Hence AnomieBOT serves as a Fixer, a Tagger, and a Clerk. Second, the bot may work on tasks from different role categories as a standard BAG-approved work procedure. For example, the work of ClueBot NG follows the anti-vandal procedure. When it detects a possible vandal, it will fix the errors in articles by reverting the vandal edit, and then it sends the corresponding editor a warning message. This bot serves as a Fixer, a Protector and a Notifier. Similar standard procedures include archiving, auto assessment, newsletter delivery, and so on.

5 IDENTIFICATION OF BOT ROLES

Given a taxonomy, we apply a machine learning model to automatically identify the Wikipedia bots' roles. To get a comprehensive, reliable dataset to train our model, two authors with Wikipedia editing experience labeled bots using the derived bots' roles based on bot annotation guidelines defined by the first author. During the labeling procedure, each author labeled the most prolific 500 bots by matching the bots' roles with their functions listed in their user page and approval history. Since a bot can have multiple roles, the authors were asked to label the bots with one or more roles. We collected two valid copies of annotations for 500 bots. We used Cronbach's α to evaluate the internal consistency of the annotations. The overall Cronbach α score is 0.903, which indicates strong agreement between different annotators [48]. We present Cronbach's α per role in Table 1.

5.1 Feature Space Design

We frame the automatic identification of bots' roles as a multi-label classification problem. Our target is to classify bots based on the task descriptions and the bots' editing behavior. To capture the characteristics of different roles, we designed three sets of features as follows. Details of the feature can be found in Appendices B and C.

Distribution of Editing Namespaces (7 features). Previous studies on human role identification in Wikipedia show that editing behavior is a strong predictor of individuals' roles [52, 54]. Thus, we first extract features that represent bots' editing behavior. Specifically, we look at the distribution of bots' edits among different Wikipedia namespaces. As editing on different namespaces is a different kind of work, we aggregate namespaces that represent the same working areas following Welser's fine-grained taxonomy [52]. Moreover, since bots often update maintenance information on their own user page, we assign such edits to the "Wikipedia" namespace. We extract each bot's namespace distribution by looking at its last 3000 edits.

Frequent Verbs (174 features). We find verbs in the bots' task descriptions are extremely informative. While a Fixer bot will use verbs like *fix*, *rename*, *redirect* many times, a Clerk bot is more likely to use verbs like *update*, *maintain* and *assist*. The text mining procedure proceeds as follows. First, we tag all the verbs that appear in the bots' user page and their Request-for-Approval history using the natural language processing pakcage spaCy [24]. Second, we extract a set of frequent verbs that were mentioned in the training corpi of at least 100 bots. Third, we train a set of Random Forest classifiers [27] using sklearn [38] for each role and extracted verbs whose feature importance are larger than 0.005. Last, we adjust the machine filtered verbs by removing misidentified HTML tags and stopwords. We counted whether the remaining verbs appear in the corpus of each bot and represented it using a Bag of Words (BOW) model.

Predefined Lexicons (2 features). We also defined a set of lexicons that represent special domain knowledge in Wikipedia. Specifically, we built lexicons that indicate two aspects of knowledge: policy violations (pov, aiv, verifiability, etc.) and frequently mentioned websites outside English Wikipedia, including wikidata, wikimedia commons, and arxiv.

5.2 Classification Results

As the task is a multi-label classification problem, we apply three multi-label classifiers following previous literature [54, 55, 58]. We use the multi-label classifiers implemented in python scikit-multilearn package [47]. Specifically, we applied the Binary Relevance kNN (**BR**) [12], Multi-label k Nearest Neighbours (**MLKNN**) [57], and Multi-label Adaptive Resonance Associative Map (**MLARAM**) [5] methods and tuned each classifier's parameters with 10-fold cross-validation. We used Micro-F1 score and Macro-F1 score to evaluate the model performance.

Table 2 shows the classification results. Both the macro and micro F1 scores improve after adding more features. This indicates that the identified three sets of features are indeed informative. Among all three classifiers, the MLARAM classifier outperforms the others. This may be because this neural network based classifier is generally better than the k-nearest neighbor based classifiers when handling high dimensional feature spaces. Figure 2 shows the average precision, recall and F1 score for each role corresponding to the MLARAM classifier. As we can see, the model is good at predicting all the roles except the Advisor. This happens for two possible reasons. First, the Advisor bots are generally more heterogeneous than bots in other categories (this role also has the lowest Cronbach's α score). They provide suggestions using different supporting mechanisms in different ways. Second, the Advisor bots are relatively few in number in our training dataset; there were only ten such bots.

6 THE EVOLUTION OF BOT ROLES

We applied the MLARAM classifier to identify the roles of all registered bots in Wikipedia as of February 28, 2019. Figure 3 shows the number of bot roles and the number of their corresponding edits. Note that, we aggregate their edits multiple times into the different bot roles. For example, if

Table 2. Performance comparison for predicting bot roles

Features	Metrics	BR	MLKNN	MLARAM
Edit History	Macro-F1	0.302	0.281	**0.552**
	Micro-F1	0.554	0.549	**0.699**
Edit History + Frequent Verbs	Macro-F1	0.239	0.242	**0.849**
	Micro-F1	0.541	0.522	**0.912**
Edit History + Frequent Verbs + Lexicons	Macro-F1	0.241	0.254	**0.850**
	Micro-F1	0.547	0.534	**0.913**

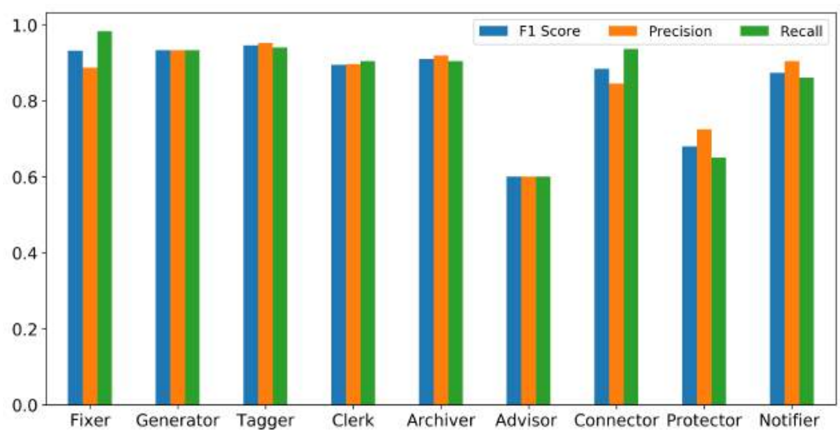


Fig. 2. Mean Precision, Recall and F1 score for each role provided by the **MLARAM** model

a bot is both a Fixer and a Tagger, the bot's edits will be counted toward both the Fixer category and the Tagger category. We find that Fixer is the most common and prolific bot type. The Tagger and Clerk bots are also very productive although they are relatively few in number. By contrast, the Connector bots are many in number but made relatively fewer edits, followed by the Notifiers, the Protectors and the Archivers. The Generators and the Advisors are small in both numbers and edits.

We next looked at the edits made by bots in different Wikipedia namespaces. As shown in Figure 4, bots that serve different roles edit different areas in Wikipedia. The Fixer, Generator, and Connector mainly take care of article content pages. The Tagger, Clerk, and Archiver maintain both content pages and the community pages (namespace: Wikipedia) with different focuses. The Advisor, Protector, and Notifier are more user-oriented when compared to other bots.

Finally, we looked at how the multi-bot system in Wikipedia has evolved. We extracted each bot's first and last edit timestamp and defined the bot as "active" for any time in between. Figure 5 shows the number and edits of active bots from January 2003 to December 2018. The number of active bots and bot edits made a leap during 2005 and 2009, stayed at its peak level for four years, then declined after 2013. During this peak time period, many Fixer, Connector, and Archiver bots with simple functions were developed, which boosted the total number of bots in the Wikipedia community. The decline in 2013 was probably caused by the inactivity of the Connector bots (54.02% decrease in number and 89.78% decrease in edits in the following year). This decrease was triggered by a Wikipedia consensus to move the old style of inter-language links relationships over to Wikidata [16]. Thus, the edits of dozens of Connector bots that were made to maintain inter-language links

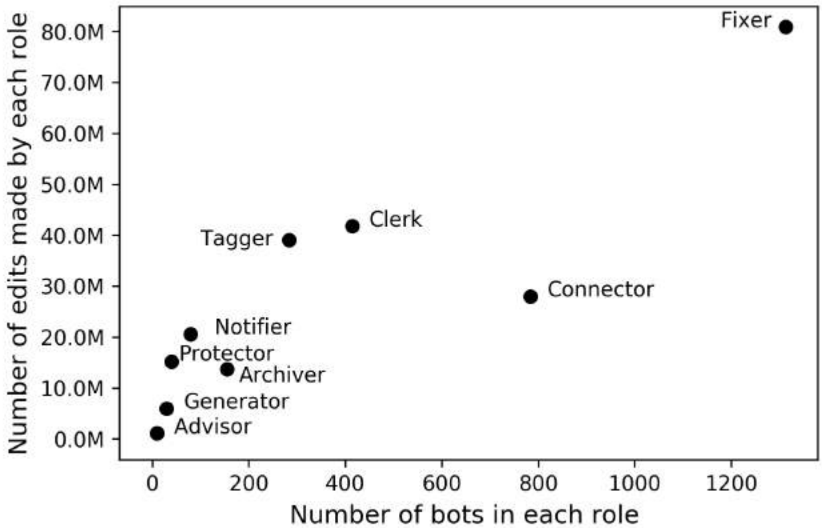


Fig. 3. The number of bots and the number of edits by role from 2002 to 2018

Fig. 4. Proportion of edits in namespaces according to bot role

declined and eventually the bots themselves became inactive. After 2013, the number of remaining active bots decreased slowly; however, the number of bot edits soon climbed back up.

This pattern suggests bot consolidation in English Wikipedia. We observe both within-category consolidations and cross-category consolidations. Within-category consolidation happened when superior or broader-scope bots take over the jobs performed by weaker or narrower-scope ones. For example, in the beginning, there were multiple anti-vandal bots working on the site (18 active anti-vandal bots on 2008), and after the launch of ClueBot NG in 2011, there were only 8 such bots left. ClueBot NG was designed to be responsive, detecting and reverting vandals' edits within 30 seconds [15] and eventually came to occupy over 34% of bot edits in its category, leaving less work for the other anti-vandal bots. Cross-category consolidation happened in two ways. First, it occurred when the bot owners decided to make their bot handle multiple tasks. For example, Cyberbot subsumed some tasks that were performed by Sigmabot. [3] Second, bots became more consolidated when some bots became inactive, because their tasks were taken over by other similar bots. For example, LegoBot took over the tasks originally performed by several bots including HBC Archive Indexerbot, RFC bot, and GA bot who had become inactive. Such consolidation can lead to the increased use of multi-task bots and a decrease in the total number of bots.

There could also be other factors at work: the interest level of bot owners in creating and maintaining their bots, speed and complexity increases in software and hardware, changes in policy, as well as changes in the activities of human editors all could have influenced the number of bots and their productivity. For example, lowering the number of bots, an effect of policy, may also lower coordination-related errors, thus increasing accuracy, which might lessen the need for fixes. Figure 5 suggests the effects of changes in bot roles on the bot population and on the number of bot edits are non-linear. That we can't fully explain the shape of these graphs, nor predict the next steps in the time series, suggests that there is a need to further investigate the dynamics of how and why bots evolve. Perhaps excitable system dynamics might be one useful technique for modeling these effects (for an example of its use in biology, see [37]).

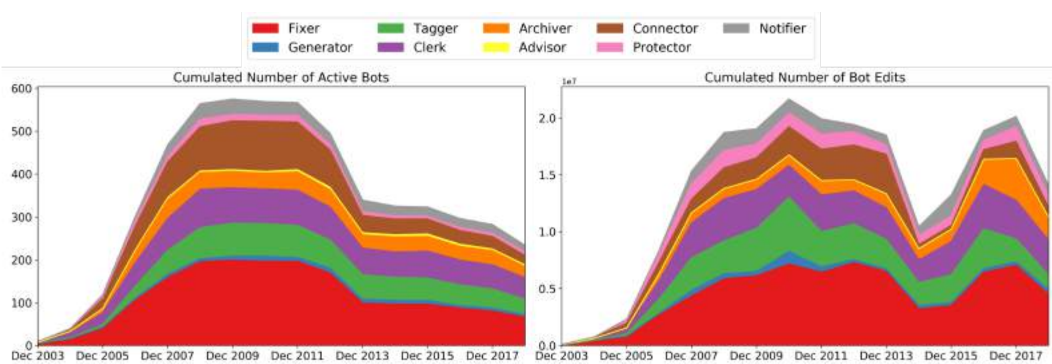


Fig. 5. Number of Active Bots (left) and Bot Edits (right) by Role

7 ROLES AND THE SURVIVAL RATES OF NEWCOMERS

Wikipedia has a long history of struggling to retain its newcomers; maintaining a large number of active participants is crucial for the community's long-term development [18, 20, 31, 45]. New contributors to Wikipedia face both social and technical barriers. For example, newcomers need to learn community policies and norms as well as wiki markup syntax [6, 33]. In addition, the edits of inexperienced editors are more likely to be reverted by experienced editors, who consider them to be threats to article quality [19]. Previous studies found that bots can have both negative and positive effects on newcomers. Halfaker et al. showed that the automatic tools designed for quality control (for example, anti-vandal bots) have inadvertently decreased the newcomers' retention rate [18]. At the same time, Morgan and Halfaker showed that inviting newly registered editors to the community portal (Wikipedia Tea House) will increase survival [32]; these invitation messages are sent by HostBot. A limitation of these studies is that they only looked at the influence of one bot, or one specific type of bot edits: revert. The automatic classification of bot roles allows us to investigate the consequences of bot-human interactions at scale – to investigate multiple bots at the same time. We illustrate this by using bot roles to predict the survival rate of newcomers.

[3] https://en.wikipedia.org/wiki/Wikipedia:Bots/Requests_for_approval/Cyberbot_II_1

Table 3. Logistic Regression Results: Bot Roles on Newcomers' Survival. (N = 10,000)

	Model 1	Model 2	Model 3	Model 4	Model 5
(Intercept)	-3.012 ***	-3.045 ***	-3.028 ***	-2.988 ***	-3.014 ***
Session edits	0.029 ***	0.021 ***	0.027 ***	0.029 ***	0.028 ***
Human	0.528 ***	0.505 ***	0.512 ***	0.563 ***	0.531 ***
Advisor		0.873 ***			
Notifier			0.162		
Protectors				-0.433 ***	
Advisor (except HostBot)					2.901 **
Log Likelihood	-2268.6	-2252.8	-2267.6	-2264.4	-2267.0

Specifically, we look at three kinds of bots that frequently interact with editors (at least 30% of bot edits are left on user talk pages) and see whether different bot roles have different effects on the survival rate of newcomers.

To conduct this empirical study, we randomly sampled 10,000 newcomers who registered between March 2018 and May 2018 and made at least one edit during that period. For all newcomers, we collected their editing history and user talk page history six months after their first edit. We tracked the newcomers' interaction history in their first two months by identifying human editors as well as bots that left a message on the newcomers' user pages. We defined a set of binary independent variables indicating whether the newcomers were reached by humans or certain bots. For example, if an Advisor bot left a message on a newcomer's user talk page during the newcomer's initial two months, the corresponding independent variable will be 1. Following previous studies [18, 32], we defined newcomers as surviving if they performed at least one edit two months after their first edit session. We also controlled for the number of edits the newcomers made during their first edit sessions [18]. The number of edits works as a proxy for the editor's initial investment in Wikipedia.

Among 10,000 randomly sampled newcomers, 4,242 were reached by either human editors or bots, and 610 survived. Table 3 shows the Logistic Regression results. In Model 1, we find the coefficient of interaction with human editors is positive and significant. This is consistent with previous literature that community support and socialization encourage newcomers' retention [10, 31]. In Model 2, we find that interaction with Advisor bots also has a significantly positive effect and that the magnitude of this effect is even larger than the effect of interacting with human editors. Given HostBot, a very active Advisor bot, has been shown to have a significant positive effect on newcomer's retention [32], we examine whether the rest of Advisor bots still have a positive effect. In Model 5, we find the Advisor role still has a significant positive effect after we remove HostBot. In Model 3 and Model 4, we find the Notifier bots do not have a significant effect and Protector bots have a significant negative effect.

The automatic classification of bot roles also allows us to perform within-category comparisons. Do all Advisor bots, as well as the Protector bots, have the same effect on newcomers' survival? Which type has a greater effect on encouraging/discouraging new editors' continual contributions? To answer these questions, we separate individual bots and run the above regressions separately while controlling for session edits and whether newcomers interact with human editors. Table 4 shows the number of editors each bot interacted with and associated logistic regression coefficients. First, the two Advisor bots that interacted with the newcomers in our sample both have a significant positive effect. HostBot interacted with many more editors than SuggestBot, whereas the latter had a larger positive effect than the former. SuggestBot provides editors a list of articles that match their interested topics and need further improvement. At the same time, it lists the quality status of the

Table 4. Coefficient of Bot effect on Newcomers' Survival by Role.

Advisors			Protectors		
botname	# of editors	coef	botname	# of editors	coef
SuggestBot	3	2.159 *	ClueBot NG	911	-0.740 ***
HostBot	462	0.856 ***	SineBot	15	0.513
			XLinkBot	52	0.783 *

articles and suggests a specific task (for example, add external source, add pictures, etc.) to improve the associated article [11]. When looking at the Protector bots, we find that only the ClueBot NG has a negative effect on newcomers' survival. The newcomers seem to not care about the bot signing their comments (SineBot) and are even positively influenced by the bot reverting their added links that violate Wikipedia's copyright policy (XLinkBot). When we compare the messages sent by ClueBot NG and XLinkBot, we find the messages left by XLinkBot are longer, more friendly, and more informative (see Appendix D). XLinkBot points out the specific reason to revert the editor's edits, provides links to the guidelines, lists more detailed information about how the bot works, and eventually supplies a link to the bot creator's FAQ page. Different reactions to these two Protector bots may be caused by different verbal traits. In general, this kind of within-category comparison allows the community to build a better bot governance system to evaluate individual bot's impact (the number of audiences) as well as effect (influence towards a specific question, in this case, newcomers' survival). In this way bot owners could learn from good examples while at the same time the community could identify bots likely to have a negative effect.

8 DISCUSSION AND CONCLUSION

This study is a step toward understanding the functions and functional categories of bots. There are many possible next steps. We identified 9 bot roles and 25 associated bot functions. We identified bots that performed multiple functions, but we did not perform an analysis of the function of each edit: this is a potential path for future research that might provide insights into the evolution of bots. Moreover, we have begun from an assumption that bots play roles related to the roles that their human editors wish to play. There is another interpretation possible: that, while bots may start off assisting an editor's self defined role, once created, a more dispassionate logic based may cause the bot designer to add other functions that are similar in some way, or use similar mechanisms, and so are easy to add. This conjecture might be analyzed by looking at the additions of functions to bots, in particular whether they seem to be based on role similarity or, instead, code similarity. Moreover, we did not distinguish between multi-function bots that grow because the designer finds that the growth is simple to implement, versus bots that are multi-functional because there are some higher level tasks that require more functions.

In considering the regression results, we note there may be self-selection bias at work. Some newcomers may need to register before receiving updates from some Advisor bots: for example, SuggestBot. Future research might try to disentangle self-selection from bot roles; one way would be through randomized assignment of bot edits to newcomers in a controlled environment. We also didn't control for newcomers' other characteristics, for example, edit intentions [55], types of messages [59], and social relationships [10]. We evaluated the complexity of bot functions based on the descriptions on the bots' user page. Alternatively, this might be estimated by analyzing the source code directly. Additionally, researchers can apply our taxonomy to bots working across different languages in Wikipedia. Bots are governed differently in different languages [40] due to different editing cultures, so researchers might produce an adapted taxonomy. Furthermore, researchers could build off our taxonomy to study bots in other Wikis and online communities. More generally, there is a fluid community of experts on bots, exemplified by the Bot Approval Group, and these experts might be consulted for their ideas about ways to both improve and apply this suggested taxonomy.

Labor markets tend toward creating specialized skills. At the same time, the expenses of labor lead to the creation of tools that augment labor. Bots are tools, but they are different in nature because of their autonomy. As their use becomes more prevalent, their effects deserve more study. Their first-order effects are the knowledge artifacts they help protect or help create. Their second-order effects are the reactions that they engender in other humans also contributing to the creation of knowledge. To understand what is happening, we need ways of differentiating bots from each other. This research takes a step toward creating a classification based on functions that bots engage in while also demonstrating how such a classification can be used to study second-order effects. For example, this study looked at the survival rates of newcomers. More generally, taxonomies can be used to create a more nuanced understanding of forces at work in a system.

Taxonomies can be dynamic: in Wikipedia, new types of bots are constantly being created. Studying the dynamics of the bots with respect their changing functions and the effects of such changes on human editors may be important for understanding the dynamic of coordination in knowledge-creating processes. More broadly, this may help us understand the changing ways automation affects knowledge production and human work.

9 ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under grants 1909803, 1717473, 1442840, and 1422066.

REFERENCES

[1] Maik Anderka, Benno Stein, and Nedim Lipka. 2012. Predicting Quality Flaws in User-generated Content: The Case of Wikipedia. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 981–990.

[2] Ofer Arazy, Felipe Ortega, Oded Nov, Lisa Yeo, and Adam Balila. 2015. Functional Roles and Career Paths in Wikipedia. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 1092–1105.

[3] Richard Barker and Cliff Longman. 1990. *Case Method: function and process modelling*. Addison-Wesley.

[4] Beth A Bechky. 2006. Gaffers, Gofers, and Grips: Role-Based Coordination in Temporary Organizations. *Organization Science* 17, 1 (Feb. 2006), 3–21.

[5] Fernando Benites and Elena Sapozhnikova. 2015. Haram: a hierarchical aram neural network for large-scale text classification. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 847–854.

[6] Ivan Beschastnikh, Travis Kriplean, and David W. McDonald. 2008. Wikipedian Self-Governance in Action: Motivating the Policy Lens. *ICWSM* (2008).

[7] Bruce J. Biddle. 1986. Recent Developments in Role Theory. *Annual review of sociology* 12, 1 (Aug. 1986), 67–92.

[8] Bruce J Biddle. 2013. *Role Theory: Expectations, Identities, and Behaviors*. Academic Press.

[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[10] Moira Burke, Cameron Marlow, and Thomas Lento. 2009. Feed me: motivating newcomer contribution in social network sites. *proceedings of the International ACM SIGCHI Conference on Supporting Group Work. ACM SIGCHI International Conference on Supporting Group Work* (2009).

[11] Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. 2007. SuggestBot: Using Intelligent Task Routing to Help People Find Work in Wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI '07)*. ACM, New York, NY, USA, 32–41.

[12] Ioannis Vlahavas Eleftherios Spyromitros, Grigorios Tsoumakas. 2008. An Empirical Study of Lazy Multilabel Classification Algorithms. In *Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008)*. Syros, Greece.

[13] R Stuart Geiger. 2009. The Social Roles of Bots and Assisted Editing Programs. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration (WikiSym '09)*. ACM, New York, NY, USA, 30:1–30:2.

[14] R Stuart Geiger. 2018. The Lives of Bots. (Oct. 2018). arXiv:cs.CY/1810.09590
[15] 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.
[16] 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'. *Proceedings of the ACM on Human-Computer* 1, CSCW (2017), 49.
[17] 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.
[18] 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. *The American behavioral scientist* 57, 5 (2013), 664–688.
[19] Aaron Halfaker, R Stuart Geiger, and Loren G Terveen. 2014. Snuggle: Designing for Efficient Socialization and Ideological Critique. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 311–320.
[20] Aaron Halfaker, Aniket Kittur, and John Riedl. 2011. Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration (WikiSym '11)*. ACM, New York, NY, USA, 163–172.
[21] Aaron Halfaker and John Riedl. 2012. Bots and cyborgs: Wikipedia's immune system. *Computer* 45, 3 (2012), 79–82.
[22] Steve Harenberg, Gonzalo Bello, L Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. 2014. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley interdisciplinary reviews. Computational statistics* 6, 6 (2014), 426–439.
[23] Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in human behavior* 49 (2015), 245–250.
[24] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural Language Understanding with Bloom Embeddings. *Convolutional Neural Networks and Incremental Parsing* (2017).
[25] John Vines Kiel Long, Selina Sutton, Phillip Brooker, Tom Feltwell, Ben Kirman, Julie Barnett, and Shaun Lawson. 2017. Could You Define That in Bot Terms?: Requesting, Creating and Using Bots on Reddit. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM*. 3488–3500.
[26] Aniket Kittur and Robert E Kraut. 2010. Beyond Wikipedia: coordination and conflict in online production groups. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 215–224.
[27] Andy Liaw, Matthew Wiener, and Others. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.
[28] Randall M Livingstone. 2016. Population automation: An interview with Wikipedia bot pioneer Ram-Man. *First Monday* 21, 1 (Jan. 2016).
[29] Tetyana Lokot and Nicholas Diakopoulos. 2016. News Bots: Automating news and information dissemination on Twitter. *Digital Journalism* 4, 6 (2016), 682–699.
[30] John B Miner, Donald P Crane, and Robert J Vandenberg. 1994. Congruence and Fit in Professional Role Motivation Theory. *Organization Science* 5, 1 (Feb. 1994), 86–97.
[31] Jonathan T Morgan, Siko Bouterse, Heather Walls, and Sarah Stierch. 2013. Tea and Sympathy: Crafting Positive New User Experiences on Wikipedia. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 839–848.
[32] Jonathan T Morgan and Aaron Halfaker. 2018. Evaluating the impact of the Wikipedia Teahouse on newcomer socialization and retention. In *Proceedings of the 14th International Symposium on Open Collaboration*. 20.
[33] Jonathan T Morgan and Mark Zachry. 2010. Negotiating with Angry Mastodons: The Wikipedia Policy Environment As Genre Ecology. In *Proceedings of the 16th ACM International Conference on Supporting Group Work (GROUP '10)*. ACM, New York, NY, USA, 165–168.
[34] Claudia Müller-Birn, Leonhard Dobusch, and James D Herbsleb. 2013. Work-to-rule: The Emergence of Algorithmic Governance in Wikipedia. In *Proceedings of the 6th International Conference on Communities and Technologies (C&T '13)*. ACM, New York, NY, USA, 80–89.
[35] Mark EJ Newman. 2011. Communities, modules and large-scale structure in networks. *Nature physics* 8 (Dec. 2011), 25.
[36] Sabine Niederer and José van Dijck. 2010. Wisdom of the crowd or technicity of content? Wikipedia as a sociotechnical system. *New Media & Society* 12, 8 (Dec. 2010), 1368–1387.

[37] Renato Pagliara, Deborah M Gordon, and Naomi Ehrich Leonard. 2018. Regulation of harvester ant foraging as a closed-loop excitable system. *PLoS computational biology* 14, 12 (Dec. 2018), e1006200.

[38] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of machine learning research: JMLR* 12, Oct (2011), 2825–2830.

[39] Marco Perugini and Richard P Bagozzi. 2001. The role of desires and anticipated emotions in goal-directed behaviours: Broadening and deepening the theory of planned behaviour. *British Journal of Social Psychology* 40, 1 (2001), 79–98.

[40] Alessandro Piscopo and Elena Simperl. 2018. Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proc. ACM Hum. -Comput. Interact.* 2, CSCW (Nov. 2018), 141:1–141:18.

[41] Joseph Seering, Juan Pablo Flores, Saiph Savage, and Jessica Hammer. 2018. The Social Roles of Bots: Evaluating Impact of Bots on Discussions in Online Communities. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 157.

[42] Stefan Seidel, Nicholas Berente, Aron Lindberg, Kalle Lyytinen, and Jeffrey V Nickerson. 2019. Autonomous Tools and Design: A Triple-loop Approach to Human-machine Learning. *Commun. ACM* 62, 1 (2019), 50–57.

[43] 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.

[44] 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.

[45] Bongwon Suh, Gregorio Convertino, Ed H Chi, and Peter Pirolli. 2009. The Singularity is Not Near: Slowing Growth of Wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration (WikiSym '09)*. ACM, New York, NY, USA, 8:1–8:10.

[46] Ed Summers and Ricardo Punzalan. 2017. Bots, seeds and people: Web archives as infrastructure. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. 821–834.

[47] Piotr Szymański and Tomasz Kajdanowicz. 2017. A scikit-based Python environment for performing multi-label classification. *arXiv preprint arXiv:1702. 01460* (2017).

[48] Mohsen Tavakol and Reg Dennick. 2011. Making sense of Cronbach's alpha. *The journal of the International Association of Medical Science Educators: JIAMSE* 2 (June 2011), 53–55.

[49] Milena Tsvetkova, Ruth Garcia-Gavilanes, Luciano Floridi, Luciano Yasseri, Taha, and Taha Yasseri. 2017. Even good bots fight: The case of Wikipedia. *PloS one* 12, 2 (2017), e0171774.

[50] Mark E Tubbs and Steven E Ekeberg. 1991. The Role of Intentions in Work Motivation: Implications for Goal-setting Theory And Research. *AMRO* 16, 1 (Jan. 1991), 180–199.

[51] Denny Vrandečić. 2012. Wikidata: A New Platform for Collaborative Data Collection. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion)*. ACM, New York, NY, USA, 1063–1064.

[52] Howard T Welser, Dan Cosley, Gueorgi Kossinets, Austin Lin, Fedor Dokshin, Geri Gay, and Marc Smith. 2011. Finding social roles in Wikipedia. In *Proceedings of the 2011 iConference*. 122–129.

[53] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum. -Comput. Interact.* 2, CSCW (Nov. 2018), 182:1–182:19.

[54] Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2016. Who did what: Editor role identification in Wikipedia. In *Tenth International AAAI Conference on Web and Social Media*.

[55] Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2000–2010.

[56] Diyi Yang, Robert Kraut, Tenbroeck Smith, Elijah Mayfield, and Dan Jurafsky. 2019. Seekers, Providers, Welcomers, and Storytellers: Modeling Social Roles in Online Health Communities. (2019).

[57] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40, 7 (2007), 2038–2048.

[58] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. 649–657.

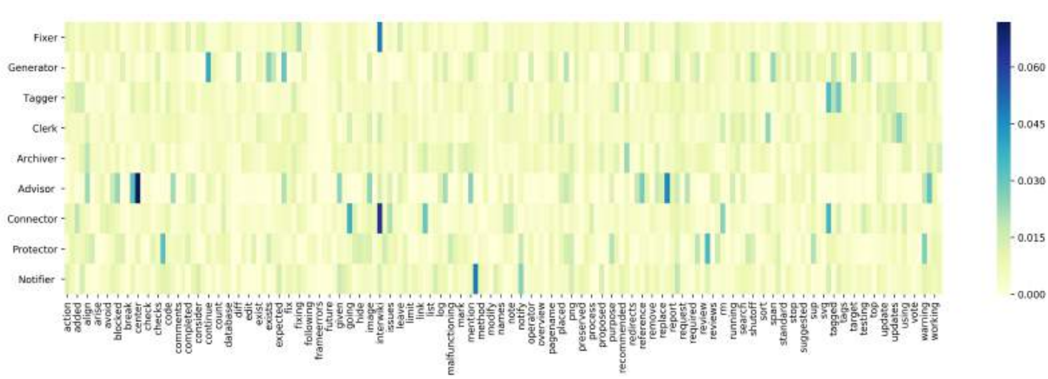
[59] Haiyi Zhu, Amy Zhang, Jiping He, Robert E Kraut, and Aniket Kittur. 2013. Effects of Peer Feedback on Contribution: A Field Experiment in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2253–2262.

A DEFINITION AND ADJUSTMENT IN SECTION 4

Definition of Bot Functions. A function is a specific process, action or task that a system is able to perform [3]. Here we define a bot function as an action on certain objects in Wikipedia. We differentiate two functions if they have different actions or act on different objects. For example, "fix content" and "tag article status" are different functions because the former acts to fix errors on a content page and the later acts to add tags such as cleanup templates to a content page. Similarly, "fix links" and "fix files" are different functions because the former fixes invalid links (for example, a dead link, an incorrect link format, or a double redirect) and the latter fixes invalid files (for example, images that are too large, or copyright infringing images/audios/videos). We also consider the goal of bot functions. A goal implies the need and desire of a social actor; goals have been widely discussed in role theory [39, 50]. Thus, we also differentiate bot functions that have different goals. For example, "send user notifications" and "provide suggestions for user" are different functions because even though they both act by leaving a message on the target audience's user page, the latter has a different goal: an intention to motivate.

Adjustment of Bot Function Clusters. We distinguish the "clean up sandbox" and "archive content" functions from other clerk functions because their goal is to make the page neat and clean, whereas the others are used for updating and maintaining information. For the same reason, we separate "send user notifications" function, and we separate "connect Wikipedia with other wikis" and "connect Wikipedia with other sites" from the generating functions. We found the "tag multimedia status" function is highly associated with the "identify policy violation" function. This is because bots that identify copyright violations will also tag the invalid images for deletion. Thus we align the "tag multimedia status" with other tagging functions because all of them are used for an administrative purpose. For the same reason, we separate the "provide suggestions for Wikiprojects".

B FEATURE IMPORTANCE OF FREQUENT VERBS



C FREQUENT VERB AND LEXICON FEATURES

Features	Words
Frequent Verbs	interwiki, run, fixing, function, span, using, running, diff, reviewed, purpose, completed, rm, center, link, reference, links, fix, exists, redirects, expect, know, redirect, names, replace, testing, hand, pagename, created, regex, updated, exist, mention, search, matching, based, added, confirm, view, caps, removes, change, forward, sort, note, follow, skip, sup, tag, tagging, supervised, tagged, preserved, modify, contribs, lists, stop, add, requested, adding, started, archive, review, svg, code, assisted, hide, report, unsupervised, namespaces, problems, comment, issues, missing, updating, space, programming, toolserver, tags, fixes, updates, action, check, reviews, works, edit, list, contact, log, count, debate, boilerplate, remove, break, arise, right, parameter, operator, separate, moved, ignore, recommended, suggest, suggestion, consider, suggested, frameerrors, following, reports, placed, given, limited, required, avoid, left, going, test, taking, standard, request, find, image, update, padding, database, png, go, align, warning, rate, sandbox, method, overview, input, comments, vote, process, cleanup, error, blocked, noticed, continue, top, shutoff, double, operated, existing, rest, replacing, issue, leave, future, plan, order, malfunctioning, messages, welcome, complete, notify, working, expected, limit, post, target, reduce, checks, manual, waiting, mark, speedy, checking, proposed, generate, provide, mainspace
Lex: Sources	wikidata, wikimedia commons, google books, crossref, adsabs, arxiv, oaadoi, pubmed, bibcode, jstor, consensus data, consensus table, database
Lex: Policies	vandal, vandalism, aiv, uaa, irc, orphan, orphaned, substitution, coi, pov, npov, verifiability, copyrighted, copyright, license, privacy, fair-use, bias, harassment, spam, spams

D MESSAGES SENT BY XLINKBOT AND CLUEBOT NG

Welcome to Wikipedia. Although everyone is welcome to contribute constructively to the encyclopedia, your addition of one or more external links to the page Samsung Galaxy A7 has been reverted. Your edit here to Samsung Galaxy A7 was reverted by an automated bot that attempts to remove links which are discouraged per our external links guideline. The external link(s) you added or changed (https://samsungupdatez.blogspot.com/Samsung-galaxy-a7-review-with-pros-and-cons) is/are on my list of links to remove and probably shouldn't be included in Wikipedia. If the external link you inserted or changed was to a blog, forum, free web hosting service, fansite, or similar site (see 'Links to avoid', #11), then please check the information on the external site thoroughly. Note that such sites should probably not be linked to if they contain information that is in violation of the creator's copyright (see Linking to copyrighted works), or they are not written by a recognised, reliable source. Linking to sites that you are involved with is also strongly discouraged (see conflict of interest).
If you were trying to insert an external link that does comply with our policies and guidelines, then please accept my creator's apologies and feel free to undo the bot's revert. However, if the link does not comply with our policies and guidelines, but your edit included other, constructive, changes to the article, feel free to make those changes again without re-adding the link. Please read Wikipedia's external links guideline for more information, and consult my list of frequently-reverted sites. For more information about me, see my FAQ page. Thanks! —XLinkBot (talk) 04:13, 25 June 2019 (UTC)

Hello, and welcome to Wikipedia. This is a message letting you know that one or more of your recent edits to Reserve Bank of India Act, 1934 has been undone by an automated computer program called ClueBot NG.

- ClueBot NG makes very few mistakes, but it does happen. If you believe the change you made was constructive, please read about it, report it here, remove this message from your talk page, and then make the edit again.
- For help, take a look at the introduction.
- The following is the log entry regarding this message: Reserve Bank of India Act, 1934 was changed by 2405:205:429F:67BC:3C9A:48CD:8B57:B259 (u) (t) ANN scored at 0.941174 on 2019-08-15T02:59:16+00:00

Thank you. ClueBot NG (talk) 02:59, 15 August 2019 (UTC)

E PREDICTED BOT ROLES AND PROPORTION OF EDITS WITHIN ROLES

Received April 2019; revised June 2019; accepted August 2019

				Fixer		Generator		Tagger		Clerk		Archiver		Advisor		Connector		Protector		Notifier	
Index	Botname	Total edits	%	P	%	P	%	P	%	P	%	P	%	P	%	P	%	P	%	P	%
0	WP 1.0 bot	6362966	5.70	0	0.00	0	0.00	0	0.00	1	15.20	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
1	ClueBot NG	5275030	4.80	1	6.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	34.70	1	25.60
2	Yobot	4692988	4.20	1	5.80	0	0.00	1	12.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
3	AnomieBOT	3773621	3.40	1	4.70	0	0.00	0	0.00	1	9.00	1	27.50	0	0.00	0	0.00	0	0.00	0	0.00
4	SmackBot	3734324	3.40	1	4.60	0	0.00	1	9.60	1	8.90	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
5	InternetArchiveBot	3115657	2.80	1	3.90	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	11.10	0	0.00	0	0.00
6	Addbot	2838809	2.60	1	3.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	10.10	0	0.00	0	0.00
7	SineBot	2261484	2.00	0	0.00	0	0.00	1	5.80	1	5.40	0	0.00	0	0.00	0	0.00	1	14.90	1	11.00
8	MediaWiki message delivery	1998561	1.80	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	9.70
9	EmausBot	1939862	1.70	1	2.40	0	0.00	0	0.00	1	4.60	0	0.00	0	0.00	1	6.90	0	0.00	0	0.00
10	Xqbot	1674366	1.50	1	2.10	0	0.00	1	4.30	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
11	RjwilmsiBot	1602950	1.40	1	2.00	1	26.90	1	4.10	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
12	ClueBot	1596818	1.40	1	2.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	10.50	1	7.80
13	KasparBot	1549811	1.40	1	1.90	0	0.00	1	4.00	1	3.70	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
14	AAlertBot	1339946	1.20	0	0.00	0	0.00	0	0.00	1	3.20	1	9.80	0	0.00	0	0.00	0	0.00	1	6.50
15	RussBot	1285862	1.20	0	0.00	1	21.60	1	3.30	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
16	COIBot	1239554	1.10	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	8.10	1	6.00
17	InceptionBot	1184730	1.10	1	1.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
18	KolbertBot	1155819	1.00	1	1.40	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
19	Legobot	1096300	1.00	1	1.40	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	3.90	0	0.00	0	0.00
20	AvicBot	1092078	1.00	1	1.40	0	0.00	1	2.80	1	2.60	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
21	Lowercase sigmabot III	1060112	1.00	0	0.00	0	0.00	0	0.00	1	2.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
22	Cyberbot I	1014464	0.90	1	1.30	1	17.00	1	2.60	1	2.40	1	7.40	0	0.00	0	0.00	0	0.00	0	0.00
23	BG19bot	1005055	0.90	0	0.00	0	0.00	1	2.60	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
24	BattyBot	975725	0.90	1	1.20	0	0.00	1	2.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
25	The Anomebot2	972155	0.90	1	1.20	0	0.00	1	2.50	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
26	SporkBot	964875	0.90	0	0.00	0	0.00	0	0.00	1	2.30	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
27	Fbot	960904	0.90	1	1.20	0	0.00	1	2.50	1	2.30	1	7.00	0	0.00	1	3.40	0	0.00	1	4.70
28	Luckas-bot	929662	0.80	1	1.10	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	1	3.30	0	0.00	0	0.00
29	Bender the Bot	829985	0.70	1	1.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
30	FrescoBot	818150	0.70	1	1.00	0	0.00	1	2.10	0	0.00	0	0.00	0	0.00	1	2.90	0	0.00	0	0.00

Note: *P* represents a binary prediction result (1=yes, 0=no); % represents proportion of edits within that role. Showing top 30 bots as an example, more deatils can be found in https://github.com/Nicozheng/Wikipedia_bots_taxonomy.